Believable Information Delivery for Prototype 3

Jarmo Laaksolahti, Thomas Rist, Mark Steedman,
Colin Matheson, Johan Bos, Fiorella de Rosis,
Phil Hand, Stephen Crampton, Catherine Pelachaud

Distribution: Public

MAGICSTER
Embodied Believable Agents
IST-1999-29078 Deliverable 3.4

Jan 2004

The deliverable identification sheet is to be found on the reverse of this page.

Project rf.no.	IST-1999-29078
Project acronym	MAGICSTER
Project full title	Embodied Believable Agents

Security	Public
Contractual delivery date	M33 = Aug 2003
Actual date of delivery	M38 = Jan 2004
Deliverable number	D3.4
Deliverable name	Believable Information Delivery for Prototype 3
Type	Report
Status & version	Final 1.0
Number of pages	28 (excluding front matter)
Contributing WP	WP3
WP/Task responsible	BARI
Other contributors	SICS, All Partners
Authors	Jarmo Laaksolahti, Thomas Rist, Mark Steedman, Colin Matheson, Johan Bos, Fiorella de Rosis, Phil Hand, Stephen Crampton, Catherine Pelachaud
EC Project Officer	Anna Katrami-Bezirtzoglou
Keywords	Cinematography, Anticipatory planning, socio-emotional dialogue
Abstract	The document describes the final prototype where the user interacts with several agents in a social context. Emotions and social relations are enhanced using affective cinematography.

The partners in MAGICSTER are:		
	University of Edinburgh ICCS	EDIN
	Università degli Studi di Roma "La Sapienza"	ROMA
	Deutshes Forschungscentrum fur Kunstliche Intelligenz	DFKI
	Swedish Institute of Computer Science	SICS
	Università degli Studi di Bari	BARI
	AvatarMe	AME

For copies of reports, updates on project activities and other MAGICSTER related information contact:

The MAGICSTER Project Administrator

University of Edinburgh

2Buccleuch Place

Edinburgh, Scotland EH8 9LW

Copies of reports and other material can also be accessed via the project's administration homepage, http://www.ltg.ed.ac.uk/magicster/

Table of contents

1 Introduction

The MagiCster project is concerned with the development of believable conversational interface agents which make use of gaze, facial expression, gesture and body posture as well as speech in a synchronised fashion. In order to cover a broad variety of potential applications the project consortium decided to investigate three different application domains each representing a different conversational setting (see Figure 1) and each having its own specific requiremts for believability. Earlier prototypes developed by the consortium have explored three of these settings as summarized below. The current report focuses on the requirements for believability for the third prototype which is a user-character dialogue system, featuring muliple characters.

The first prototype, Avatar Arena, explores the *multi-character dialogue setting* where a user observes multiple characters carrying out a meeting appointment negotiation. The basic idea of prototype 1 is that, somewhat similar to an arena, users send their delegates (avatars) to a virtual space where the avatars negotiate on behalf of their owners. Both the results and the process of a negotiation are then displayed to the users in the form of a dialogue simulation in which the avatars appear as embodied, affective conversational characters. User can influence avatars behaviour by specifying its likes, dislikes, social relations and agenda (see D3.2 for more detail).

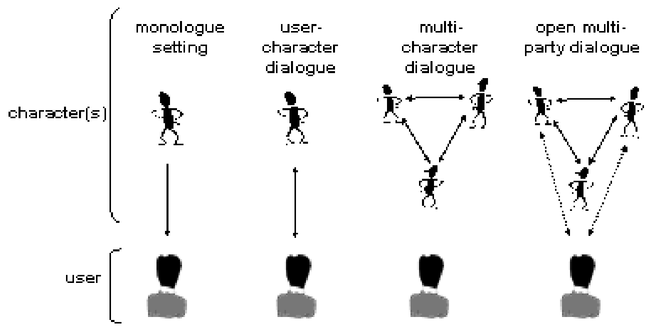


Figure 1: Dialogue settings for MagiCster prototypes

The second prototype addressed the *monologue* and *user-character dialogue* settings where a user observes or interacts with a single character. The general goal of prototype 2 is to explore how to introduce believability in emotional dialogues in general and is medical advice dialogues in particular. In such dialogues an ‘expert’ introduces users to ‘appropriate’ behaviors in some specific domain (e.g. healthy eating), by providing them with information and suggestions and, if necessary, by persuading them to follow these suggestions. The dialogues are emotional in the sense that the affective state of the user may be influenced by the information imparted and because the expert may react to the users dialogue moves by showing an ‘empathic’ attitude (see D3.3 for more detail).

The third prototype consists of an interactive multi-character game application. The scenario focuses on three teenage girls arranging a party for their friends and classmates. The basic idea is that the user takes on the role of one character, and interacts with the other – computer controlled – characters in the scenario. The user communicates with the characters by typing sentences on the keyboard. Depending on how the user acts or reacts events will unfold in different ways. The events taking place in the scenario are displayed to the user in the form of a simulated dialogue featuring embodied, affective and social characters. Expressiveness and believability of the characters is further enhanced by the use of cinematography to emphasize important aspects of events. The motivating factor for playing the game is not necessarily to win it, but instead to explore the social and emotional relations between characters. Instead of monster bashing the goal of the game is to navigate the socio-emotional web and find a path through it. This is something which requires both sensitivity and attentiveness to social and emotional cues from players.

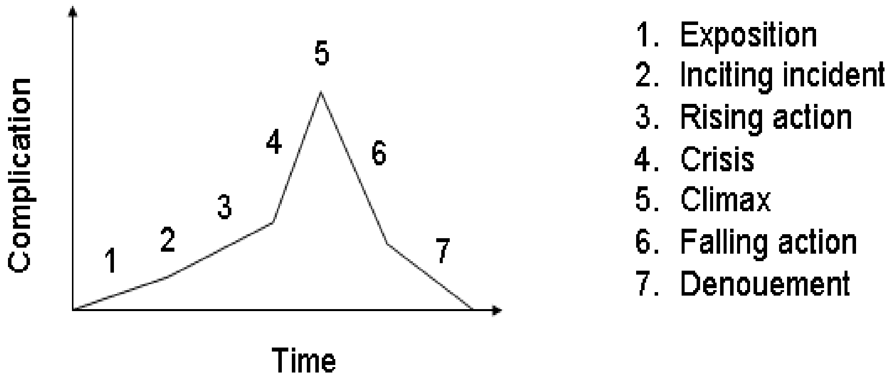


Figure 2: Dramatic arch

Establishing a narrative context for characters to exist in can be very important for believability. For example, Rizzo (1999) has shown that emotions sometimes require a narrative context in which they can be given their meaning and understood. As emotions are an important aspect of believability (see e.g. Bates, 1994) these results transitively apply to believability as well. Phoebe Sengers (1999, 2003) takes the notion of a narrative context a step further and argues that some situation or interaction history is necessary for characters to be understandable in general. Believability is less of an issue if a character exists in a social and emotional vacuum, a world devoid of other characters, situations, and humanly relevant events. A player's interaction with a character becomes meaningful and rich once we know something about the character's background (back-story), and it is placed in a narrative situation.

The game attempts to create a narrative context explicitly by providing characters with a backstory and a situation in which they exist, but also implicitly by attempting to make the user experience a dramatic arc while playing the game. The game is loosely based on aristotelian dramatic principles which state that a drama consists of several parts or phases as depicted in Figure 2 (see. e.g. Laurel 93, Mateas 2002). The vertical axis represents time and the horizontal axis represents complication. In the exposition phase complication rises slowly as the viewer gets to know the characters and the situation. An inciting incident then adds momentum to the story leading to a quick rise of complication eventually building to a climax where the drama peaks. During the climax complications are resolved after which the level of complication rapidly drops.

Dramatic principles manifest themselves in the game by the fact that the game keeps track of the level of complication or tension and attempts to nudge events in a direction that would roughly fulfil a dramatic arc. However fulfilment of a dramatic arc is not guaranteed. The player can very well ruin the story. The main vehicle for doing this is an execution manager controlling the characters based on anticipatory planning, a planning strategy that takes predictions about future events into account when deciding what to do in the present.

The final prototype builds on the experiences gained from constructing a prototype in early stages of the project to investigate interactive narrative as a possible domain for MagiCster technology. While the early prototype was entirely text based, used scripted input from the user and had a predefined story structure, the current version has a graphical interface and allows a more free form of input from the user – albeit still in textual form – and finally has a more dynamic structure. The focus lies less on sequencing predefined pieces of a story than simulating a situation in which a story may emerge.

The game consists of a server component hosting the game logic and a player component hosting the graphical representation of the world and characters. The player component is implemented around the AvatarMe rendering engine which constitutes the graphical interface to the game. Figure 3 shows a screen dump of the current player version. The player also integrates the Greta FAP generation engine and the Festival Speech synthesis engine. Figure 2

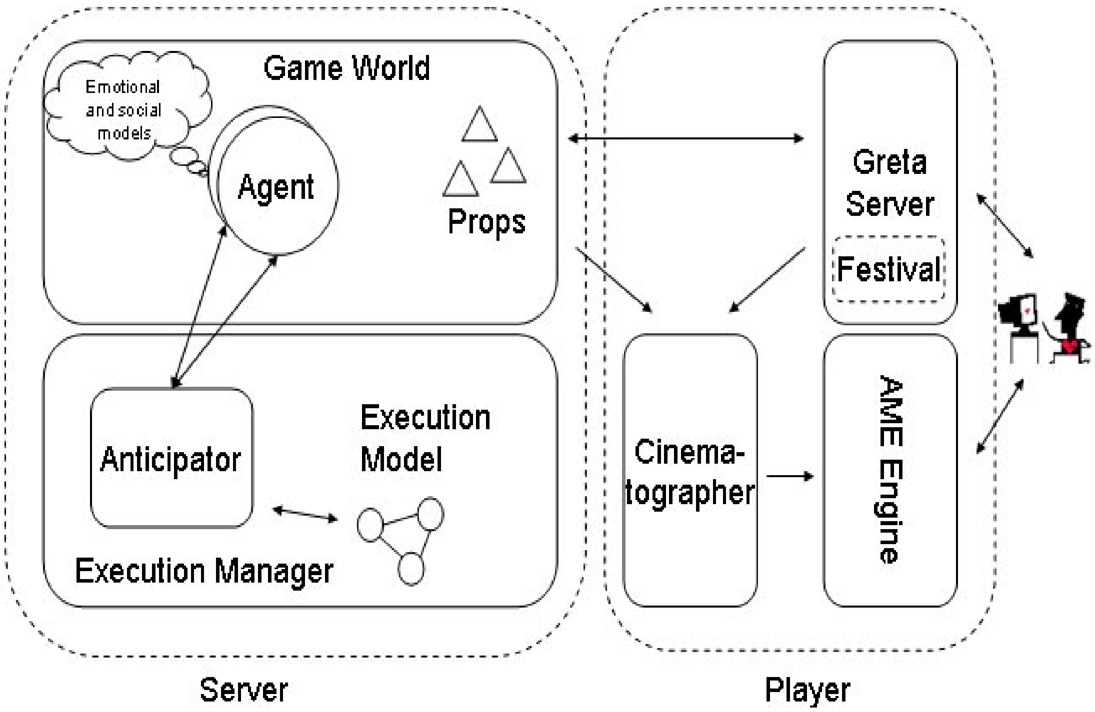


Figure 3: The party game application

The cinematographer and 3Dworld together forms the player part of the system (see Figure 4) while the other parts constitute the server part. The party game also builds on and integrates many features from the previous prototypes, in particular Avatar Arena. They both share the following characteristics:

- Both simulate negotiation dialogues, albeit with different characteristics.
- The social model developed for Avatar Arena is used to provide interesting group dynamics in the scenario.
- A template-based dialogue generation approach is used.
- The JAM agent architecture is used as a basis for authoring agents/characters.

Figure 4 Screen shot of the player

2 Believability for interactive game scenarios

Believability is often – somewhat circularly – defined as a character's ability to suspend disbelief. Believability arises between a specific user and some symbolic representation of a character (image or text). A character is considered believable "if it allows the audience to suspend their disbelief and if it provides a convincing portrayal of the personality they expect or come to expect" (Loyall, 1997). That is, believability makes the user 'disregard' the physical circumstances of the cinema, theatre, book, reading situation or the technical features of the computer game, and instead 'enter' the story world, focusing on the events and becoming cognitively, emotionally and morally engaged in the lives of the characters. This is partly dependent on the features of character, partly dependent on the attitude and ability 'to let go' in the user.

Believability exists on different levels. In films, computer games or plays, a character may move, act and behave believably on a graphical level. However, most of what we normally define as believable character arises from how the character behaves in relation to a socio-emotionally rich 'situation'. Such a situation or 'scene' involves other characters' goals, intentions, personalities, and social identities as well as objects and artefacts handled and manipulated by those characters. This level of believability has little to do with the medium of the presentation, and can emerge through written (literature) and spoken text (radio theatre) as well as through moving images and live acting.

Whether a character is experienced as believable or not is thus a function of how well its behaviour fits with the reader's or user's expectations about characters and the situation in which the behaviour takes place. Such expectations, which are of a folk-theoretical 'common-sense' nature, are vast and complex (see D3.1 for a more thorough discussion). Hence believability is not something universal that cuts across domains or applications but rather something very specific. What is believable for one character or context may not be for other characters or contexts. For instance, a piano falling on a character's head without causing serious injuries is believable in many cartoons but would hardly be believable in a medical advice scenario.

2.1 Believability in the party game application

Work on achieving believable agents within the project has so far resulted in agents capable of expressing their personality through gaze, facial expressions, gestures, body posture and synthesized speech in a synchronized fashion (Pelachaud et. al, 2002; Rist and Schmitt, 2003). However, believability exists on multiple levels and is not only affected by the internal state or appearance of characters. For instance, movies show us that there are several believability factors that are *non-diegetic*, external to any agent or even the world in which agents exist. Examples of such factors include music, sound effects and cinematography. By altering these factors different dramatic and emotional effects can be constructed, clarified or enhanced without changing the content (Boardwell and Thompson, 2001; Mascelli, 1965). We believe that taking these factors into account when constructing believable agents can make them more believable. Figure 3 suggests how believability can be divided into different spheres. These spheres are complementary to the levels of believability discussed in deliverable D3.1.

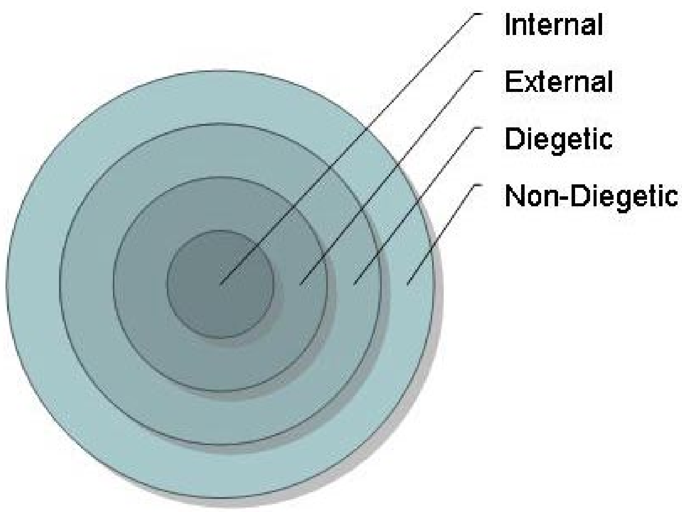


Figure 5: Layers of believability

The internal sphere refers to a character's internal processes, for instance emotion models, social models and reasoning capabilities. Internal processes or the results of those are visible to observers in the world through the external sphere, for instance through facial expressions, gestures or movement schemes. The diegetic sphere refers to a character's believability in relation to the world or setting it exists in. The outmost layer is the non-diegetic sphere that lies outside the character or even the story world and includes cinematography, music and sound effects. Not all spheres need to be modelled but when present they should be consistent or the chain of believability will be broken.

Prototypes one and two of MagiCster focused on the internal and external spheres of believability. Prototype three builds on these prototypes but also considers the extra-diegetic sphere by introducing cinematography as a believability factor. Cinematography refers to how something is filmed in contrast to what is filmed and will be discussed below. Prototype three also has a slightly stronger focus on the diegetic level in that more work has gone into designing the environment where characters live, and the situation they exist in.

In contrast to the other MagiCster applications the party game application is one in which agents are not intended to be helpful or useful. Instead their task is to create an entertaining experience for the user within the given context. The party game application is also an interactive application. In general believability is much harder to achieve in interactive media than in linear ones. In film and literature, where the producer can rest assured that the consumer actually accesses the narrative in a particular sequence, the character's behaviour and reactions can be shaped and specifically timed to be meaningful and believable. Since an author or director fully determines the context in which behaviour takes place, s/he can also control the believability to a much larger extent than a computer-based or real-life interactive drama. In the latter cases, the producer must instead create a 'whole' personality or social role, which is able to behave meaningfully in many, perhaps unknown, contexts (cf. Loyall, 1997).

2.2 Believable characters in interactive entertainment

Within the gaming industry believable characters have always had an important role in computer games from early Infocom text adventures like Planetfall to modern classics like the Sims. Different games have different requirements for believability. First person shooter games such as Quake or Unreal require computer controlled opponents to be 'intelligent' enough to provide the user with some resistance while in other games such as role-playing games it is more important to express a personality. Typically these games contain characters equipped with personalities and sometimes rudimentary models of emotion (the SIMS) aiming to spice up their behavior. While dialogue-driven interaction has always been an important part of role-playing games, it is lacking in Sims-like simulation games. However, most aspects of a game, including character-character dialogue, user-character dialogue and any accompanying gestures or emotional expressions, are pre-scripted by an author making most levels of believability a function of the game designer's abilities. At the same time the rich settings and roles constructed by the game designers make many games successful in meeting the fictional expectations that users may have vis-à-vis characters in the game, thus making games believable at the fictional level (cf. MagiCster Deliverable D3.1).

Within the research community there has also been a great deal of interest in believable characters in relation to games and entertainment applications, especially interactive drama.

The OZ project (Loyall, 97) created an agent system HAP that provides tools to create reactive agents with distinct personalities – or believable agents. The personality of HAP agents is coded by a programmer as a set of goals and a set of behaviours that are executed in order to pursue the goals, react to the environment and express reactions related to goal fulfilment or failure. HAP agents were used in several prototypes e.g., The Edge of Intention prototype which contained multiple quite simple HAP agents called “woggles”. The user could interact with the agents via their own avatar’s body language and later also limited text input. The level of personality that is shown by characters is determined by the number of behaviours defined by the author.

Façade (Mateas and Stern, 2000) is a system developed within the OZ project that integrates plot and character in interactive drama. The system builds on the HAP architecture but extends it in several ways. The system contains virtual characters able to express themselves with gestures, facial expressions and speech. Characters also have limited natural language capabilities. The system consists of a central story manager that guides an ensemble of agents in a story world. The main architectural entity that the system uses to structure the story is the beat. Beats in Mateas' and Sterns' system contains information about which story values are affected by the beat as well as joint plans and goals needed by the characters to enact the beat. Stories are interactively created by selecting beats to perform that fulfil global narrative goals.

Barbara Hayes-Roth uses principles from improvisational drama, for instance Commedia del arte, to create agents – or actors – capable of improvisation (Hayes-Roth et al., 97). Agents are cast in different roles that define their personality and constrain their possible actions and are then told to improvise with their roles as starting points. Hayes-Roth's system also uses simple plots that aid in structuring improvisation by providing agents with cues – or preconditions – that must be met for scene transitions to take place. A similar approach is adopted by (Kleesen, 00). Their system produces emergent narratives, created when a player interacts with a group of synthetic actors in a virtual story world.

One of the earliest efforts in computer science to create an interactive story system is described by Laurel (Laurel 93). She describes a design for an expert system, which was never implemented, that acts as a computerized playwright to create interactive drama. Laurel's system collects suggestions from each character about what to do next, then evaluates the effect of the actions and chooses the first acceptable suggestion. To help it decide which actions are appropriate the system uses a formal specification of the intended drama.

From these few examples we can see that there has been a fair amount of work done on believable agents in relation to interactive drama. The character-based approach has so far been the most popular one. In purely character-based systems the idea is that a story will emerge from the interaction between a player and a set of personality-rich characters. However, often a certain amount of planning is necessary to create a rewarding narrative experience. If successful the story can suggest to players what their next step should be, thus keeping them within the limits of the system. Galyean (1995) refers to this process as 'narrative guidance of interactivity'. With a few exceptions (see e.g. Mateas, 2002) there appears to be very little research on how to design systems that accomplishes this.

2.3 Believability factors for the party game application

The party game application inherits believability factors from the Avatar Arena since they have a common foundation in both technology and the type of dialogue modelled.

- *Domain competence*; the characters need to show some understanding of the subject matter domain (here inviting people to a party)

- *Dialogue skills*; the characters need to adhere to some basic rules for participation in a group discussion/negotiation dialogue.
- *Social competence*; the characters should display affect in compliance with both assigned personality traits and changes of affective states and social relationships that may occur in the process of a negotiation dialogue.

In addition the party game application must also address believability requirements stemming from the fact that it is an interactive application which uses multiple characters.

- *Reactivity/Proactivity*. Agents must react to the user but also sometimes act proactively. When agents in an interactive application, e.g. a computer game, fail to proactively take action, although sometimes a very long period of time has passed, they do not act believably. Many current dialogue systems are designed to be question answering machines instead of believable characters.
- *Working towards global goals*. When dealing with a single agent, or a monolithic system acting as several agents, the author – analogous to a movie director – can be reasonably certain about what message is being delivered. In a system with multiple autonomous agents this is not as easy since information delivery is distributed over the whole agent community. Sometimes agents may act in opposition to goals established for the community as a whole. Information delivery in such an environment needs to be controlled somehow.

Finally the party game application explores the use of other techniques for creating believability.

- *Extra-diegetic believability*. Factors outside the agent system may also have a crucial impact on believability. Consider for example what happens when a horror movie is shown with the "wrong" music. Instead of being frightening the whole movie becomes ridiculous and unbelievable. Choosing the right channels and expressions we believe that extra-diegetic believability factors can be of great value.

3 Domain, dialogue and social competence

The concept of dialogue as entertainment is not new in computer games. Many computer games, in particular games belonging to the role-playing (RPG) genre, feature some sort of dialogue with characters inhabiting the game world (see e.g. Rouse, 2001). In most cases the dialogue exists to enhance the feeling of being immersed in a game world and has little or no effect on the outcome of the game. The players' role in the dialogue is usually limited to choosing from a set of predefined statements to which the characters give scripted answers.

The party game takes steps towards giving dialogue a more prominent role in gaming. In contrast to current games the dialogue between the players and characters stands in focus. Dialogue is not merely an enhancer but the tool that players use to interact with characters and make things happen in the world. In fact the whole purpose of the game is to simulate a social situation where the dialogue drives the action and where the result of actions is in itself dialogue. In addition players are not restricted to choosing from predefined statements but are free to work out the proper things to 'say' for themselves.

3.1 Modeling Social Relationships

Social relationships are of special importance in the scenario. All through the game relations change and transform according to the actions taken by the player, and there has to be continuous appraisal of the social consequences of one's actions.

In the same way as Avatar Arena social networks are based on characters' attitudes towards a set of value dimensions and assessments of other characters attitudes towards the dimensions. For our purposes we have limited ourselves to three quite broad dimensions pertaining to the theme of the scenario and the characters' traits.

Dimension	Importance
School	1-3
Romance	1-3
Popularity	1-3

Figure 6: Value dimensions

As in Avatar Arena it is possible to assign different importance to these value dimensions for each character.

3.2 Domain modeling

In gaming, a narrative context is often planted in the game via some form of riddle, puzzle or task in which the player/protagonist should engage, in our case organising the party. The party scenario in fact introduces several puzzles that must be resolved in order to successfully organize the party:

1. Invite people
2. Arrange a place to hold the party
3. Get food and drink
4. Arrange music

For the final prototype we have chosen to focus on the first of these puzzles; inviting people. The players' task during the game is to agree on a list of people to invite to the party. As the game advances the girls will air their opinions about the persons being considered. Like the meeting arrangement scenario, the invitation scenario can be seen as a type of negotiation dialogue albeit with different characteristics. The scenario is of an *open multi-party dialogue*, including both agent-agent interaction as well as agent-user interaction.

3.2.1 Modelling persons

To be able to discuss whom to invite to a party characters need to have some understanding of what makes a person desirable to invite. For our purposes we wanted a set of dimension along which a character could be assessed that would make sense from a teenager's perspective as well as provide material for interesting disputes among the characters. What we ended up with is a characterisation of people along a few attributes that capture some qualities that are interesting for our scenario.

- Age
- Looks – how pleasing is the person to the eye (subjectively)?
- Status – what is the status of this person?
- Intelligence
- Sportiness
- Wealth

Based on the dimensions modelled above characters can perform stereotypical reasoning about the invitees (Rich, 1979). For example a character with high intelligence and low status could be classified as a typical *geek*, someone with good looks and high status would be the *cool guy,* and high sportiness and low intelligence would indicate a *jock*. Stereotypes and traits form the basis for how characters think about and describe invitees'; "I think X is really cute!" or "X is such a geek!". Together with the value dimensions of the social model characters can reason along the dimensions to decide what to think of a potential invitee. For instance, status can be important when considering popularity, assuming that people with a high status are more important for popularity than people of low status. Intelligence (or sportiness) might be important for determining if the invitee could be important for school work.

In addition these traits can help us model another dimension that seems very important, namely attraction. That is, invitees that a character finds attractive are more desirable to invite. Research in social psychology has examined what qualities are desired in potential romantic partners (Chapdelaine et al, 1999). Physical appearance seems by far to be the quality most accurately predicting attraction. Several studies have also verified that men's preference for this quality is stronger than women's. On the other hand, status seems to be more important for women than for men, while age is equally important for both sexes. However there is also a social dimension to attraction. Other characters' *attitude* towards a person – we will call it *reputation* here – is also an important factor, perhaps especially for teenagers. Hence attraction can be seen as a function of age, looks, status and reputation.

Trait/Dimension	School	Romance	Popularity
Age		+	
Looks		+	
Status		+	+
Intelligence	+		
Sportiness	+		
Wealth			+

Table 1: Relation between traits and value dimensions

The factors affecting the desirability of an invite are summarised in Table 1 together with links between traits and a value dimensions. As in Avatar Arena, expressing a high interest in e.g. school would assign high importance to the traits intelligence and sportiness in an invitee.

3.3 Modeling dialogue skills

As in the Avatar Arena prototype the third prototype is modeled as a form of negotiation dialogue. In fact the dialogue moves related to negotiation used in the system are borrowed from Avatar Arena. However, the structure is simplified in this respect in that some dialogue phases have been left out.

- Greeting Phase
  - Greet
  - Reply-to-greeting
- Negotiation Phase
  - Suggest-Person,
  - Accept -Person
  - Reject-Person
  - Justify-Rejection, Accept
  - Ask-Attitude
  - Inform-Attitude
  - Justify-Attitude
- Leave-taking
  - Goodbye
  - Reply-to-goodbye

In addition to the negotiation-related dialogue moves we have added some moves intended to "stir the pot". The groups are linked by the fact that moves belonging to the latter group influences which moves are selected in the first group (and vice versa). For instance a character that you have insulted will be less likely to accept invitees that you propose.

- Social Regulation
  - Insult
  - Reply-to-insult
  - Apologize
  - Reply-to-apology
  - Flatter
  - Reply-to-flattery

Sometimes it is reasonable to think about e.g. insults being a property of other moves. A proposal can, after all, be rejected in an insulting way. However, often enough an insult does not carry any additional information, in which case its existence as a separate move is motivated.

3.3.1 Turn taking

The party game application contains several agents, each potentially running in its own separate thread. Agents request the turn to speak whenever they feel the desire to do so along the lines outlined in D3.2. In the party game application we also have to overcome the problem of synchronizing turn taking in some fashion. One solution is to model turn taking behaviour as an auction that is only open for a specified time slot. Agents can request the turn whenever it is available. The first agent that requests the turn starts the auction by posting a bid for the turn in terms of motivation to speak. Following the first bid the auction stays open for a specified amount of time. After the auction closes the turn goes to the highest bidder. All agents that placed bids on the turn during the auction are informed about who received the turn and who else took part in the auction.

The mechanism allows agents to mimic situations where several people want to speak simultaneously with a time aspect added to the equation. If some agent repeatedly keeps winning the turn, blocking other agents from speaking, the ‘frustration’ felt by those agents may affect their desire to speak the next time the turn becomes available. In situations where there is no competition for the turn the time aspect of course is not an issue.

As the goal of the game is to entertain players by letting them experience a dramatic arc we must also work towards achieving that goal. If agents were completely free to regulate the turn taking behaviour themselves this would be hard to achieve. Hence the system has the opportunity to intervene in the process by giving the turn to another agent instead of the winning one. Which agent is given the turn is determined based on an evaluation of the dramatic consequences of a turn. If any dramatically undesirable effects of a turn are detected – e.g. the dramatic arc becomes too distorted – the turn instead passes to another agent. The procedure is repeated until a dramatically pleasing turn is found, or if none exists, all agents have been tried. In the latter case the least undesirable turn is chosen for execution.

The player of the game holds a privileged position. If the player requests the turn it is usually given to her, since unresponsiveness to player input might quickly break the illusion we are attempting to uphold. As a result the player has a great influence on how the game proceeds.

4 Reactive/proactive behaviour and achieving global goals

The player’s mode of interaction with the game and characters is through dialogue. The player uses the keyboard to type sentences that are parsed by the agents (see D2.5 for more details). Understanding and reacting to the players input is a very important aspect of believability. Loyall (1997) claims that “whether the agent is intelligent or competent at problem solving is much less important than whether the agent is responsive, emotional, social and in some sense complete”. If the system fails to understand the user’s input the illusion of believability quickly breaks. However, natural language is very complex, and understanding all possible strings that the player may input is beyond current technology. Hence, the characters’ language understanding is narrowly focused around the topic of the game. However we have tried to make the system somewhat resilient to off-the-wall remarks from the user by providing dramatically interesting fallback mechanisms. For instance, characters may cover for parsing failures by having temper tantrums or pretending the user said something she did not (see e.g Loyall, 1997). The aim of the fallback mechanisms is to pull the user back onto the right track if possible. As one of the joys of playing a game is to test its limits, remarks that are unrelated to the topic at hand are to be expected.

In interactive applications pro-activity is also an important aspect of believability. Situations where nothing happens simply because the player has not given any 'orders' should be avoided. Thus characters will go about living their lives if the player for some reason stops acting. For the party game this means that if the player has not done anything for a specified amount of time the characters will try to involve players e.g., by addressing them: “Karin, has the cat got your tongue?” or “Karin, what do you think about Niklas?”. Alternatively the characters will continue to assemble the list of people to invite without the player's involvement. In this case the system turns into a pure simulation.

4.1 Working towards global goals

As multi-agent systems grow more and more complex it becomes difficult to foresee what the effects of individual actions within the system will be. One solution to this is to run a simulation of the agent community so that alternative paths of action can be assessed before one is committed to. If the need arises the simulation can serve as a basis for tuning the agent community so that a desired path of execution is achieved. This is the idea underlying anticipatory planning. Anticipation is a mental activity that is routinely practiced by many organisms in various situations. In short it means that knowledge of future states guides current behaviour. For instance, tennis players anticipate the trajectory of the ball and stockbrokers anticipate prices on the stock market (Davidsson, 1996).

Anticipatory systems contain a predictive model of themselves and/or their environment, which allows them to change state based on the model's predictions about future states (Rosen, 1974). The basic architecture of an anticipatory system is shown in figure 6.

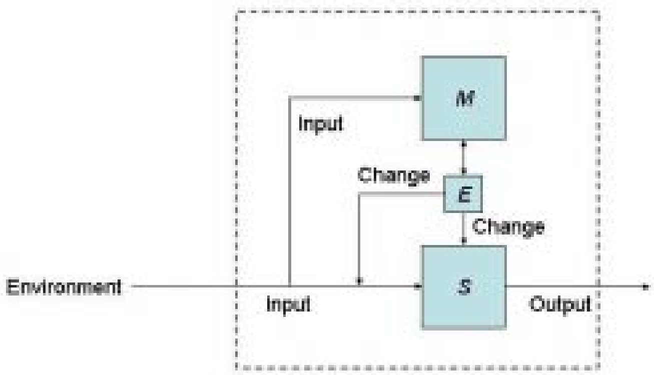


Figure 6: Anticipatory system architecture

It consists of an object system S, an associated model M of the object system, and a set of effectors E that changes parameters in S itself or in the environment. The model is executed faster than real-time which means that it is always ahead of the object system. That is, any time we look at the model its sequence of states will have preceded that of the object system. In this way the model predicts the behaviour of the object system by looking into the future.

Anticipatory systems can be classified based on the role of the predictive model M. The classification scheme introduced by Butz et al. (2002) distinguishes between four types of anticipatory systems:

- *Implicitly Anticipatory Systems* are systems without any predictive model M but where implicit anticipatory information is encoded by nature or a designer in the interplay between sensors, algorithms and actuators.
- *Payoff Anticipatory Systems* consider predictions regarding the possible payoff of different actions to decide on which actions to execute. Typically preconditions estimate the payoff of an action.

- *Sensory Anticipatory Systems* use predictions about future states to influence sensory processing. Although state predictions are made they do not directly influence decision making, but instead function as a way to prepare the system for a certain kind of input.
- *State Anticipatory Systems* use predictions of future states to directly influence current decision making.

Davidsson (2003) adds a further dimension to the categorization by considering how predictions are affected by the quality of the world model used and the properties of the environment. Agents that have complete world-models and operate in a deterministic environment are said to be linearly anticipatory since they are always able to predict exactly what will happen. Agents that have incomplete world-models or operate in a non-deterministic environment on the other hand face a harder task, since they need to evaluate a tree of possibilities.

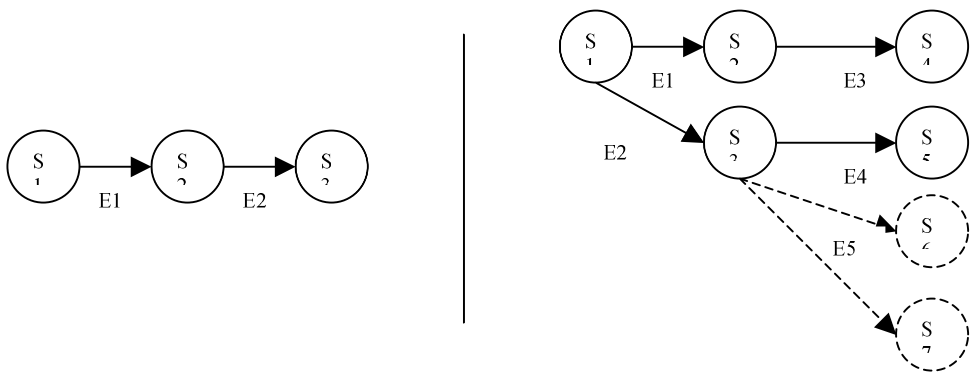


Figure 7: Linear and non-linear anticipation

Predictions about future states can influence S in many ways. A simple approach, suggested by Rosen (1974), is to divide the state space of S into desirable and undesirable regions, or in Rosen's own words "Let us imagine the state space of S (and hence of M) to be partitioned into regions corresponding to "desirable" and "undesirable" states. As long as the trajectory in M remains in a "desirable region", no action is taken by M through the effectors E. As soon as the M-trajectory moves into an "undesirable" region [...] the effector system is activated to change the dynamics of S in such a way as to keep the S-trajectory out of the "undesirable" region." (Rosen, 1974:p.247). Hence, if a system remains in a desirable state most of the time an anticipatory system will rarely interfere with the system's normal execution.

Anticipatory planning has been applied in various domains including classifier systems (Butz and Goldberg, 2002) and stock market prediction (Edmonds, 2002). Davidsson presents a linearly anticipatory agent framework that combines reactive and deliberative behaviour (Davidsson, 1996; Davidsson, 2003). The framework is illustrated by simple path-finding agents, but more complex scenarios are also considered. A similar framework is used by Boman et al. (2000) to implement a multi agent system handling energy management in a public building. In computer games anticipation has been used to make 'bots' in the computer game Quake smarter (Laird, 2001). The SOAR Quakebot uses a state anticipatory mechanism to predict what enemies are likely to do based on what it would do itself in a similar situation. The predictions are then used to ambush enemies and deny them 'power-ups'.

4.1.1 S, M and E in the party game application

The party game belongs to the state anticipatory category of system in that it explicitly tries to predict future outcomes and allows them to influence present behaviour. The main loop of the game uses a turn-based algorithm. For each cycle the turn is given to an agent along the lines described above. After the speaking agent has executed its turn the effect of the turn for the other agents is calculated, and so on. The point where anticipation comes into play is when assigning the turn to an agent. When the competition for a turn closes an execution manager runs a simulation of the agents to decide whether any of them have undesirable consequences. That is the execution manager attempts to predict what the next state would be and let that influence present behaviour. In our case we only simulate a single turn. However, if desired the execution manager could run longer simulations.

Linking the terminology of anticipatory systems to components in the party game is relatively straightforward. The object system S clearly consists of the agents themselves, as it is their behaviour that we wish to anticipate. The model M consists of *copies* of the agents that are made when simulating the agent community, but also a model of the desired execution path of the system. This model takes the form of a finite state machine, which will be described below. Finally, the system only has a single effector which is choosing which agent should receive the turn. More complex – and active – effectors are certainly possible. For instance effectors performing some limited "brain surgery" on agents by adding or removing goals, facts or plans. However we decided to adopt a simple solution before turning to complex solutions that would require more work.

Anticipation is used to control the agent community as a whole. However, it can also be beneficial for the believability of individual agents. For instance, an agent that predicts that a fellow agent will become very sad by what it has to say may instead chose to keep quiet or not tell the whole truth, which would mimic empathic behaviour. The framework that we have implemented also allows these kinds of anticipatory agents to be constructed.

Compared to traditional planning techniques, for instance search used by Weyhrauch (1997) to combine scenes into desirable plot paths, anticipation as described here is a more passive way of planning. Anticipation as used in the party game application gives the system a look-ahead similar to that provided by search. By contrast, Rosen's description suggests that the main concern for an anticipatory system is to detect and avoid undesirable states instead of actively searching for the best. Instead of searching for an optimum the goal is to avoid minima.

The environment facing agents in our scenario is deterministic in the sense that agents will behave in the same way given the same circumstances. This is in line with the requirement of consistency of behaviour, which is an important component of believability (Isbister & Nass, 2000). Having a human user involved adds an element of uncertainty to the system in that the system cannot predict what the user's next action will be or how the user will act if a certain event takes place. Extending the system with a user model might be a way of reducing uncertainty by integrating the user into the prediction process. However, constructing accurate user models of this type is notoriously difficult and lies outside the scope of this project.

4.1.2 Model of desired execution path

The purpose of the execution manager is to steer the behaviour of the system as a whole in a direction that lets the player experience a dramatic arc. In order to do that we need to determine what constitutes complication in the game. As previously stated, level of complication in drama refers to unresolved questions and unsettled feelings, among other things. For simplicity we decided to turn to the social model used in the game to model complication. The 'story' of the game is one of changing friendship relations between the characters. Social relationships are defined numerically in the social model as integer values between -3 and +3. Negative numbers indicate a negative relationship between characters while positive numbers indicate a positive relationship, and 0 is neutral. At the beginning of the game relationships are somewhere on the positive side of the scale. As the game progresses the execution manager will attempt to push events in a direction that makes the social relationships first become negative and then either return to neutral/positive or remain in the negative range as the game ends. Thus the goal of the execution manager becomes to manage the development of social relationships over the course of the game.

Tracking social relationships does not require a very complicated model. However, to make the system more general, and usable in other situations as well, we decided to model the execution of the agent community as a finite state machine, where each state corresponds to a certain configuration of the community. The FSM approach should be sufficiently powerful to handle other situations as well.

The FSM consists of states and transitions. Each transition between states corresponds to a set of conditions that can be either *true* or *false*. Conditions can refer to any belief, desire or intention that an agent may have but also to other factors such as global parameters. Currently we have the following general purpose relational condition types:

- Range
- Boolean
- Greater
- Less
- Equal
- Less/Equal
- Greater/Equal

In addition we have designed a set of conditions specific to agents that relate directly to the facts, goals, plans, emotions and relations that an agent may have:

- Knows
- Feels
- Goal
- Plan
- Attitude

For the simple scenario in the current implementation of the game we have only used two of the condition types. Figure 8 shows an example of a state machine encoding desired execution paths for the game. For clarity the states have been arranged to resemble the dramatic arc depicted in Figure 2. The transitions are labeled with the values of the two preconditions that must be met for the transition to be possible. The first number indicates an 'Attitude' precondition and in which direction it must change. The second number indicates the minimum number of turns – modeled as a 'Greater' condition – that must have passed since the current state was entered before the transition can take place. Thus we can see that as the game starts a slight decrease in the relationship is required, then a period of relative calm between q1 and q2, after which the relationship hits the bottom notation in q3. After that the relationship either starts to mend or stays negative, reflecting a more permanently changed relationship. The reflexive transition going from q1 back to q1 indicates that between q1 and q2 the relationship may in fact improve slightly. The preconditions are used by the execution manager to evaluate the desirability of different agents' actions. For instance if the system is in the start state and an agent wants to say something which would improve instead of worsen the relationship another agent would be given the turn instead.

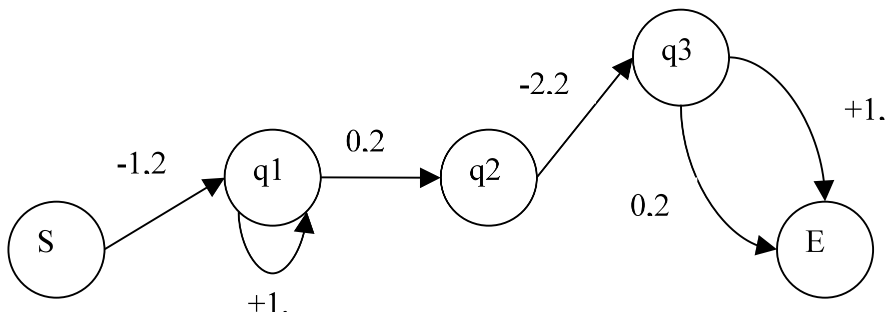


Figure 8: Execution model in form of an FSM

The state machine in Figure 8 only contains desirable states. In some situations it might be necessary or beneficial to model undesirable states as well. By including undesirable states the system can be made more resilient because transitions leading back from the undesired state can be thought out while designing the state machine. However, the tuning of resilience procedures is a delicate matter which should be tested empirically.

5 Expressive Cinematography

The cinematographer is situated between the game world and the renderer component. It is designed to help convey characters' facial expressions and body language, as well as attitudes within a group. In the general case limited screen size makes it difficult to simultaneously show (several) characters' facial expressions since characters need to be of a certain size for expressions to be readable. The cinematographer overcomes this by acting as an intelligent camera that knows when to switch between characters in an appropriate way, or more broadly, uses cinematographic techniques to enhance and clarify the state of a single agent, a group of agents or a whole scene. One could say that the cinematographer acts as an amplifier of social and emotional signals.

There is more to making a movie than simply filming what is put in front of the camera. A filmmaker also has to consider the cinematographic qualities of the shots (Boardwell &

Thompson, 2001). Cinematography refers to *how* something is filmed – in contrast to *what* is being filmed – and typically involves three factors:

- *Photographic aspects of a shot,* e.g. how a shot is illuminated. For instance a shot can be very dark and gloomy, light from the sides casting sinister shadows, or it can be bright and happy.
- *Framing of a shot,* i.e. what is included in the camera rectangle and its location within the rectangle. For instance, a shot can be centred on a person talking to someone outside the frame, or both persons can be visible on opposite sides of the frame.
- *The duration of a shot*. A shot can be very long, e.g. showing a person giving a speech, or short, showing the person giving the speech and then rapidly cutting to the audience's reactions to the speech. As a general rule shots last between two and eight seconds.

By altering these factors different dramatic and emotional effects can be constructed although the content remains the same. We have focused on the framing and duration aspects of cinematography leaving photographic aspects for later work.

Figure 9: Overview of the scenario

Most work in automated cinematography has so far focused on techniques for describing and solving the geometric constraints of a scene, such as where to place the camera given two characters talking to each other so that none of the characters are occluded (Hornung et. al, 2003). In contrast very little work seems to be available on how cinematography can actually be used to make a scene more expressive, understandable or believable. Our work belongs to this second category in that less has been spent on the geometry of cinematography than on exploring the expressiveness afforded by cinematographic techniques.

In addition we have been inspired by the language of comics, which often uses shape and colour of panels to convey emotions and attitudes. Below we explain how cinematography and comics relate to our system.

5.1 Cinematographic concepts

In order to give film a defined structure, it helps to think of it as a hierarchy. At the top level there is a series of *scenes* showing us a situation or action. Each *scene* is composed of several *shots* – the interval between two camera cuts or movements. During a *shot* the camera has a specific *camera placement*. The camera placement determines what is shown during a shot, for instance a close-up of a speaking character or a view over-the-shoulder of a listening character. *Idioms* function as recipes for combining shots into a pleasing whole as described above.

An important concept in cinematography is the *line of action* (or the axis of action). According to Mascelli (1965) "The axis is a means of remaining on one side of the players, so that the players' positions and looks appear consistent from shot to shot as the sequence progresses, regardless of the player or camera movement involved". Consequently once the line of action has been established for a certain setup, the camera must remain within a 180 degree angle of the line. The line of action is established by drawing an imaginary line that stretches in the direction of movement, or in case the scene is static, through the characters nearest the camera on opposite sides of the picture.

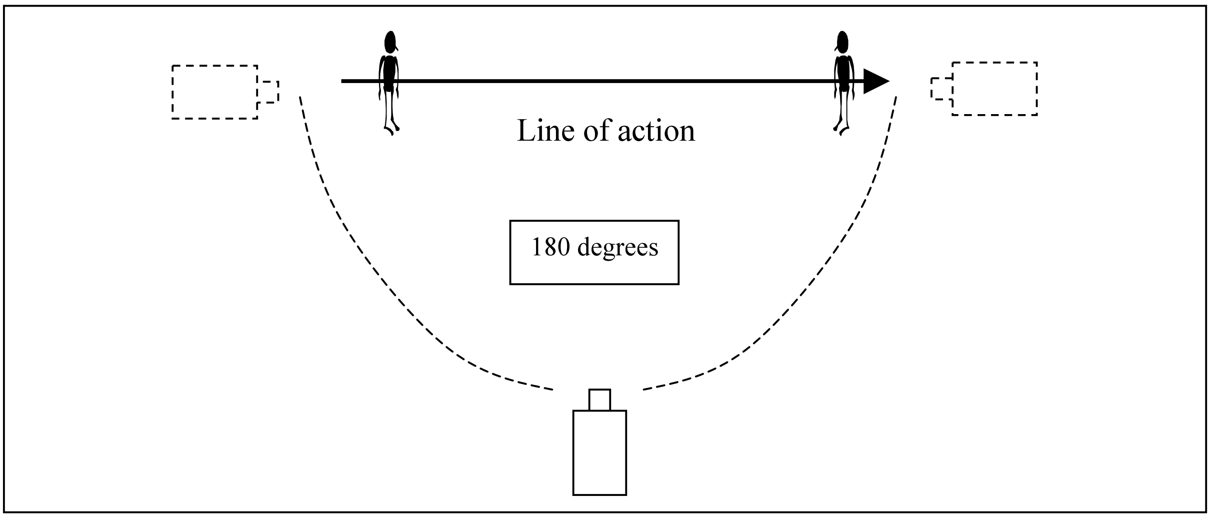


Figure 10: The line of action

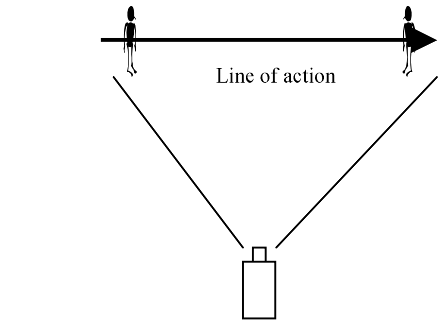
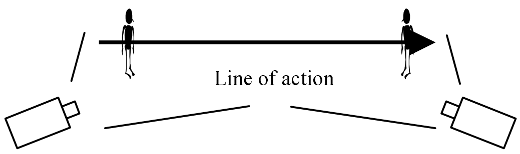
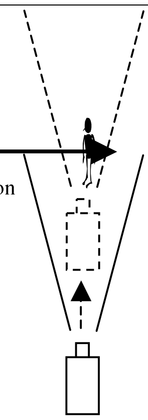
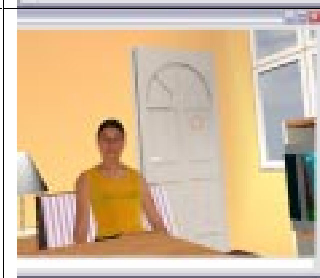
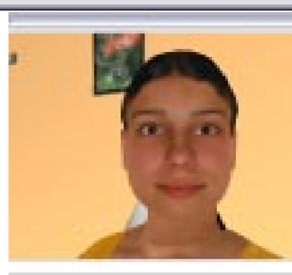
5.1.1 Shot types

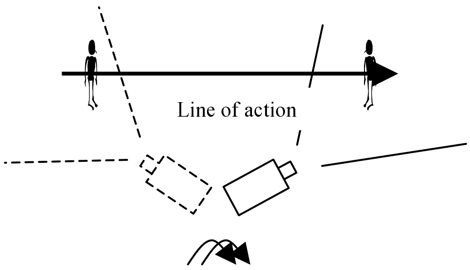
There are several common shot types that can be defined in relation to the line of action, for instance apex and over-the-shoulder shots. Shots differ in terms of the size of the camera window, the shape of the window, the angle to the target (high, low) and distance from the target (Boardwell & Thompson, 2001). The shots that we have implemented include:

- *Apex2, Apex3*. An overview shot from a medium distance showing the participants of a situation.

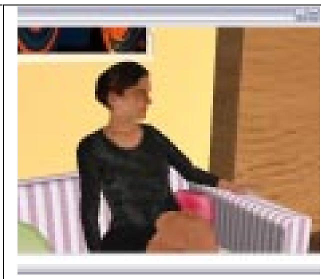
- *Over-the-shoulder*. As the name suggests a shot directed at one character slightly from behind and to the side of another character.
- *Close-up*. A shot showing e.g. a characters face or portions of the face.
- *Full.* A shot showing the whole character.
- *Zoom.* A mobile shot that starts with a full shot and zooms in to a close up of a character or vice versa.
- *Pan*. Another mobile shot that rotates the camera from one position to another.
- *Stay.* A shot that simply copies the information from the last shot.

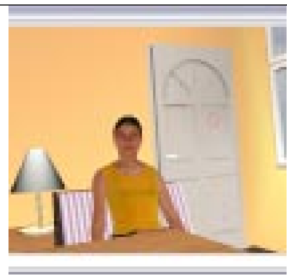
Below we give examples of some shots and their appearance on screen.

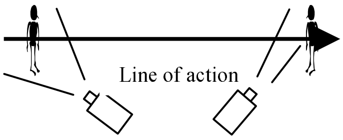
 Apex shot	
 Over-the-shoulder shot	
 Zoom	 



Pan







Full shot/ Close-Up

In addition to the shot types described here filmmakers usually employ various shot types related to moving characters, such as tracking, and crane shots (Boardwell & Thompson, 2001). However since the characters in our application do not move around we have not implemented those shot types, although if would be quite easy to do so.

5.1.2 Idioms

Over the years filmmakers have accumulated knowledge about how to apply cinematography in various situations. This knowledge has been condensed into *cinematographic idioms* that encode working knowledge about how to set up and sequence shots to capture a specific situation or action. For instance, an idiom can describe how to film a conversation between two characters. Using idioms consisting of commonly used shots will make a player feel comfortable watching a scene since s/he recognises the film language.

The core of our system is a set of cinematographic idioms that decide how to sequence shots and various shot types. Each idiom controls how the camera(s) should act or react within a certain type of scene. Apart from camera placement the cinematographer also uses other techniques to enhance a scene. To make certain that the player does not miss any of the action in the scenario the cinematographer has the ability to display several cameras at once if needed, either as overlays in the main camera window or as separate cameras on the screen.

Idioms can have varying degrees of specificity ranging from very general, such as a general "two-talk" idiom as depicted in Figure 11, to very specific, such as an "short angry two-talk with dramatically changing relations". The cinematographer chooses the most suitable idiom from its collection of idioms based on the information it receives from the party server application. The information available for the cinematographer to base its decision on is:

- The number of characters involved in the scene
- Speaking character
- Listening character(s)
- Relationships that have changed between the speaker and listeners, including the amount, direction and unexpectedness of the change. This information is provided by the social model.
- The previous history of shots and scenes.
- Information provided by the AME rendering engine such as position or heading of avatars.
- FAP length
- Timing info from the Greta server about the onset and duration of emotional expressions

Furthermore the system is also prepared for handling information about the emotional state of characters although an emotional model has not been integrated into the current version of the system (see below).

The information above is also used by the idioms to create the shot sequences. For instance, the timing information from Greta tells us when the speaker is saying something emotional. Hence it is used to direct the camera at the speaker so the player does not miss emotional expressions while at the same time showing the other participants in the conversation. Idioms can be thought of as state machines where each node represents a certain shot type. For instance, the two talk idiom in Figure 11 starts with an establishing apex shot that shows the characters involved in the situation. Then it switches to a close up of either A or B or to an over the shoulder shot from A to B. As the scene progresses the camera switches between the different shots as dictated by the transitions and their conditions.

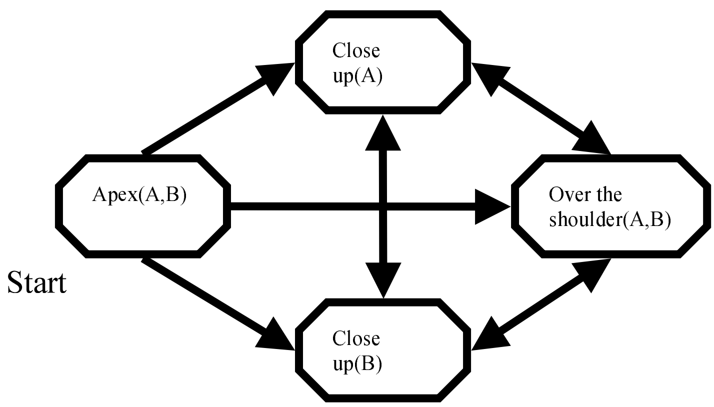


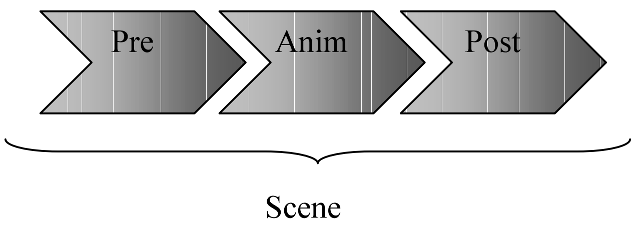
Figure 11: A general two-talk idiom.

In addition to the line of action rule there are also other types of cinematographic rules of thumb that are important to consider (He et. al, 1996). For instance, shots showing the actors of a scene should be used to establish the situation for the viewer. However, none of these rules are automatically enforced by the cinematographer. Rather it is up to the authors of idioms to make sure that they are followed or not. In some cases breaking the rules may be what you want to do.

5.1.3 Scenes

A scene in the cinematographer's view corresponds to a turn with the corresponding FAP, WAV and TIM files. The duration of a scene is in general determined by the length of the FAP/WAV file. However, if the file is very short the scene may be 'padded' to reach an acceptable duration. In general a scene should not be shorter than the duration of a single shot (2 seconds) but can be arbitrarily long.

Scenes have a sequence of pre-shots, a sequence of shots belonging to the actual FAP animation and a sequence of post-shots. Post and pre sequences can be used for establishing and closing shots or for padding the scene with shots in case it is too short to illustrate what is needed. For instance, if the scene is less than 2 seconds long, the cinematographer may add an establishing shot to the list of pre-shots, instead of adding it to the sequence of shots belonging to the FAP animation. Post shots can be used to show changing relations or to simply keep the camera on the target for a short period of time after the animation finishes, avoiding jerky cuts.



5.2 Color and shape of panels

That color affects our emotions and how we perceive things is a well-known fact (Derefeldt and Berggrund, 94). The exact way in which they affect us depends on many factors such as inherent properties of the color, expectations and previous experiences as well as cultural affiliation. For instance Goethe created a color wheel showing the psychological effect of each color. He divided colors into two groups - the plus side (from red through orange to yellow) and the minus side (from green through violet to blue). Colors on the plus side produce excitement and cheerfulness. Colors on the minus side are associated with weakness and unsettled feelings.

Figure 12: A variation of Goethe's colour wheel

The shape of objects can have similar effects on how we perceive things. Within fields such as industrial design, the shape of objects and how they are interpreted is an important topic. Round shapes with soft curves are generally interpreted as friendly and positive while angular, harder shapes are interpreted as more negative (Monö, 1997; Ståhl et. al, 04). Comic artists have embraced the power of shape and often manipulate the shape of panels and "speech bubbles" to clarify what is going on in a strip or to produce different emotional effects (McCloud, 95).

Inspired by these ideas we wanted to use similar techniques in the cinematographer. By framing shots we add a non-diegetic emotional channel to the cinematographic display that functions as an amplifier. Emotions and attitudes expressed through facial expressions, gestures or words that normally may be hard to detect have a chance of coming to the player's attention.

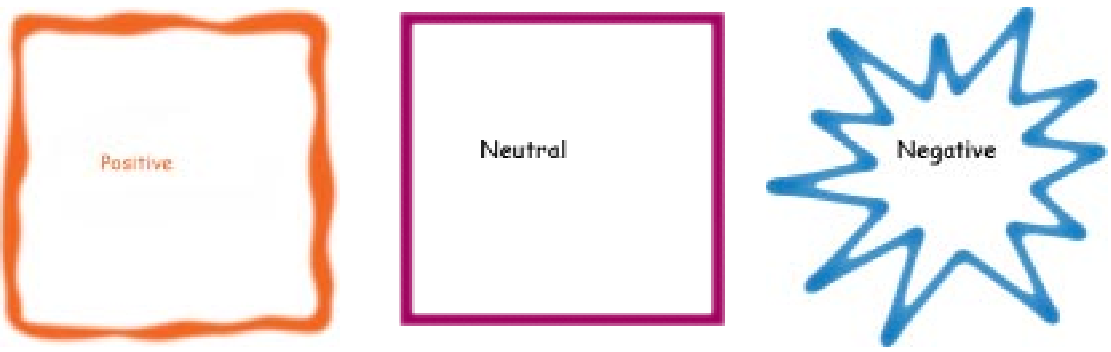


Figure 13: Colour and shape of frames

The cinematographer uses borders with different colours and shapes around the camera windows to enhance emotions and social relations. Following the theories of Goethe a bluish frame would be used in a situation when characters are showing negative emotions or when a relation deteriorates, and a reddish frame would be used when showing positive emotions or when a relation improves. Similarly a jagged frame can for instance mean that a character is angry or that the relation between two characters is tense, while a softly rounded frame can mean that the character is feeling good or the relation is good. Figure 13 shows an example of some borders combining the shape and colour dimensions. When applied during runtime the borders are stretched to match the current size and aspect ratio of the camera windows.

5.3 Animating cameras

The cinematographer has the ability to animate the camera windows on the screen to illustrate e.g. changing relations by altering the distance between two camera views on screen. A growing distance would signify a deteriorating social relationship whereas a shrinking distance would signify the opposite. Since group dynamics is a vital part of the game scenario, it is important that this aspect is made clear to the player. In combination with the use of borders this can result in dynamic displays inspired by comics as illustrated in Figure 14. Again, when and how to use animated cameras is up to the author of idioms.

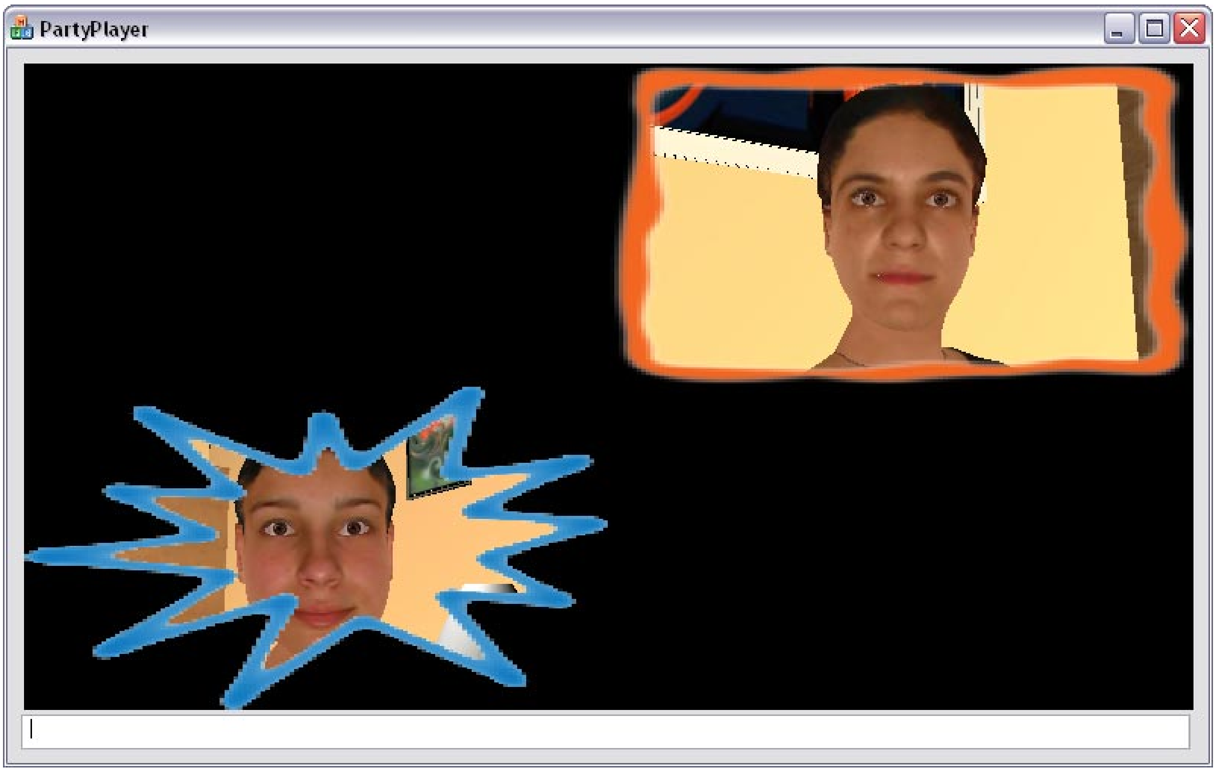


Figure 14: Animated cameras

5.4 Scripting

To make the system more flexible most parts of the cinematographer are scriptable using the LUA extension language (Ierusalimschy et. al, 1996). This makes it fairly easy to create new idioms or shot types without having to rebuild the system. For instance a camera moving along a curve can be constructed simply by writing an appropriate script.

6 Emotion Modeling

In the early stages of the project we implemented a text-based version of the game that used an emotional model based on Roseman's (1996) theory of emotions. According to this theory emotions are appraised according to a set of parameters. For instance, is the event that caused the emotion *motive consistent* or *motive inconsistent*? Is the event that caused the emotion *self-caused*, *other-caused* or *caused by circumstance*? Is it *certain* or *uncertain* that the event causing the emotion will actually occur? Is the person in high or low *control* of the situation? (Could she have done something to prevent the event from happening?) For instance, if the event is caused by someone else and has consequences that are oppositional to a person's goals/motives, then anger will prototypically arise. Roseman proposes five parameters resulting in a total of 15 emotions. For the prototype we only used two parameters (motive consistency, motive inconsistency and self-caused / other-caused / circumstance-caused). Based on their emotional state characters would generate non-obtrusive emotional expressions (in text) that informed the player about their state, e.g. "Ebba frowns". The emotions also partly determined how the user could navigate the game since certain choices were made available or unavailable depending on the emotional state of the characters.

The model was quite rudimentary. It could not handle mixed emotions and did not have any automatic decay function, among other things. Hence we planned to move to the MIND emotion module in later prototypes as it overcomes the shortcomings of our initial model, and we integrated MIND with prototype 3 by writing appropriate Java wrappers. However, as the project proceeded several problematic issues became apparent.

One issue is that the MIND module is designed for scenarios containing at most two agents where the user is one of those agents. Currently MIND is capable of modeling the event-based emotions from the OCC model:

- *joy, distress*
- *hope, fear*
- *envy, gloating*
- *happy-for* and *sorry-for*

In contrast the party game is a three-agent scenario, which means that MIND in its current incarnation cannot be directly used. One solution would be to use several MINDs to model the relationships between the agents. However, there are several theoretical and practical problems with this approach. For instance, it is unclear how the outputs from several different MIND modules interact with each other. It is also unclear what the result would be if one model states that the character feels joy and the other distress. Another more complex example is if an agent feels happy-for one agent but sorry-for another agent. There would probably be some interaction between these emotions that would need to be resolved.

In addition a closer scrutiny of the type of dialogues generated by the third prototype revealed that the prevailing emotions in the dialogues fall somewhat outside the categories above. Thus, using MIND to model emotions in the agents would not alter their behaviour very much, if at all.

As mentioned, from a software engineering point of view the MIND module was successfully integrated into the final prototype. However, given the latter considerations a decision was made not to create MIND models for the agents as the effort involved would have been considerable, and not justifiable given the aims of the prototype.

References

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

Boman, M., Davidsson, P., Kummeneje, J., and Verhagen, H. 2000. An Anticipatory Multi-Agent Architecture for Socially Acceptable Action. In The 6th International Conference on Intelligent Autonomous Systems, pp. 744 - 750. IOS Press.

Bordwell, D. and Thompson, K. 2001. FILM ART - An Introduction. McGraw-Hill, 6th edition.

Butz, M. V. and Goldberg, D. E. 2002. Generalized State Values in an Anticipatory Learning Classifier System. In Proceedings of the Workshop on Adaptive Behaviour in Anticipatory Learning Systems.

Butz, M. V., Sigaud, O., and Gerard, P. 2002. Internal Models and Anticipations in Adaptive Learning Systems. In Proceedings of the Workshop on Adaptive Behaviour in Anticipatory Learning Systems.

Davidsson, P. 2003. A Framework for Preventive State Anticipation. In Adaptive Behavior in Anticipatory Learning Systems. Springer Verlag.

Davidsson, P. 1996. Autonomous Agents and the Concept of Concepts. Ph.D. thesis, Department of Computer Science, Lund University, Sweden.

Derefeldt, G., Berggrund, U., (1994) *Färg som informationsbärare*, FOA Report FOA-R--94-00048-5.2—SE, ISSN 1104-9154.

Edmonds, B. 2002. Exploring the Value of Prediction in an Artificial Stockmarket. In Proceedings of the Workshop on Adaptive Behavior in Anticipatory Learning Systems.

Figueira Filho, C. S., Ramalho, G. L., (2000) *JEOPS – The Java Embedded Object Production System*, Lecture Notes in Artificial Intelligence, vol. 1952, pp. 52 -61.

Galyean, T. 1995. Narrative Guidance of Interactivity. Ph.D. thesis, School of Architecture and Planning, MIT.

Hayes-Roth, B., van Gent, R., and Huber, D. 1997. Acting in Character. In Petta, P. and Trappl, R., editors, Creating Personalities for Synthetic Actors, Towards Autonomous Personality Agents, volume 1195 of LNAI, pp. 92-112. Springer-Verlag.

He, L, M. F. Cohen, D. H. Salesin. (1996) *The Virtual Cinematographer: a paradigm for automatic real-time camera control and directing* in Proceedings of SIGGRAPH 96, Computer Graphics Proceedings, Annual Conference Series, 217-224.

Hornung, A., Lakemeyer, G., Trogemann, G., (2003) *An Autonomous Real-Time Camera Agent for Interactive Narratives and Games*, Lecture Notes in Artificial Intelligence, vol. 2792, pp. 236 – 243.

Ierusalimschy, R., de Figueiredo, L. H., Celes, W., (1996) *Lua-an extensible extension language*, Software: Practice & Experience 26 #6 (1996) 635-652.

Isbister, K. & Nass, C. (2000). *Consistency of Personality in Interactive Characters: Verbal Cues, Non-Verbal Cues, and User Characteristics*. International Journal of Human-Computer Studies, 53(2): 251-267.

Kleesen, M., Szatkowski, J., and Lehmann, N. 2000. The Black Sheep - Interactive Improvisation in a 3D Virtual World. In i3 Annual Conference, pp. 77-80. i3, September.

Laird, J. E. 2001. It knows what you're going to do: adding anticipation to a Quakebot. In Proceedings of the 5fth international conference on Autonomous agents, pp. 385-392. ACM Press. 1-58113-326-X.

Laurel, B. 1993. Computers as Theatre. Addison-Wesley. ISBN 0-201-55060-1.

Loyall, A. B. 1997. Believable Agents: Building Interactive Personalities. Ph.D. thesis, School of Computer Science, Carnegie-Mellon University. Tech report CMU-CS-97-123.

Loyall, A. B. (1997). Some Requirements and Approaches for Natural Language in a Believable Agent. In Petta, P. and Trappl, R., editors, Creating Personalities for Synthetic Actors, Towards Autonomous Personality Agents, volume 1195 of LNAI, pp. 113-119. Springer-Verlag.

Mascelli, Joseph V., (1965) *The Five C's of Cinematography*. Silman-James Press, Los Angeles, 1965, ISBN: 1-879505-41-X

Mateas, M. and Stern, A. 2000. Towards Integrating Plot and Character for Interactive Drama. In AAAI Fall Symposium Technical Report FS-00- 04, pp. 113 – 118. AAAI.

Mateas, M., (2002) *Interactive Drama, Art and Artificial Intelligence*, PhD Thesis, School of Computer Science, CMU, CMU-CS-02-206.

McCloud, S. 1995. Understanding Comics: The Invisible Art. Epix Förlags AB, swedish edition. ISBN 91-7089-008-0.

Monö, R. (1997) *Design for Product Understanding*, Liber AB, Stockholm, Sweden.

Pelachaud, C., Carofiglio, V., De Carolis, B., de Rosis, F., Poggi, I., *Embodied Contextual agent in Information Delivering Application*, First International Joint Conference on Autonomous Agents & Multi-Agent Systems, Bologna, July 2002.

Rich, E., (1979) User Modeling via Stereotypes, Cognitive Science 3, pp. 329 – 354

Rist, T., and Schmitt, M. (2003). *Applying Socio-Psychological Concepts of Cognitive Consistency to Negotiation Dialog Scenarios with Embodied Conversational Characters*. Proc. of AISB'02 Symposium on Animated Expressive Characters for Social Interactions. 79-84.

Rizzo, P., (1999) *Emotional Agents for User Entertainment:Discussing the Underlying Assumptions*. International Workshop on Affect in Interactions: Towards a New Generation of Interfaces, Sienna, Italy.

Roseman, I., Antoniou, A., Jose, P., (1996) Appraisal Determinants of Emotions: Constructing a More Accurate and Comprehensive Theory, Cognition & Emotion, 10(3), pp. 241-277

Rosen, R. 1974. Planning, Management, Policies and Strategies - Four Fuzzy Concepts. International Journal of General Systems, 1:245 - 252.

Rouse, R., (2001) Game Design: Theory & Practice, Wordware Publishing Inc., 2001, Plano, ISBN: 1-55622-753-3

Sengers, P. 1999. Narrative Intelligence. In Dautenhahn, K., editor, HumanCognition and Social Agent Technology, volume 19, pp. 1 - 26. John Benjamins Publishing Company.

Sengers, P., Schizophrenia and narrative in artificial agents. In Mateas, M., Sengers, P., editors, Narrative Intelligence, pp. 259-278, John Benjamins Publishing Company.

Ståhl, A., Fagerberg, P., Höök, K., Designing for Affective Expressions, Submitted to DIS 2004.

Voerderer, P. 2000. Interactive Entertainment and Beyond. In Zillman, D. and Voerderer, P., editors, Media Entertainment: The Psychology of its Appeal, pp. 21-36. Lawrence Erlbaum Associates.

Weyhrauch, P. 1997. Guiding Interactive Drama. Ph.D. thesis, School of Computer Science, Carnegie Mellon University. Report CMU-CS-97- 109.