Final thesis

Story telling engine based on agent interaction

by

Juan Carlos Porcel

LIU-IDA/LiTH-EX-A--08/028--SE

2008-06-05
Final Thesis

Story telling engine based on agent interaction

by

Juan Carlos Porcel

LIU-IDA/LiTH-EX-A--08/028--SE

2008-06-05

Supervisor and examiner:  Mikael Kindborg
Department of Computer and Information Science
At Linköping University
To my friends; old and new.
Abstract

Comics have been used as a programming tool for agents, giving them instructions on how to act. In this thesis I do this in reverse, I use comics to describe the actions of agents already interacting with each other to create a storytelling engine that dynamically generate stories, based on the interaction of said agents.

The model for the agent behaviours is based on the improvisational puppets model of Barbara Hayes-Roth. This model is chosen due to the nature of comics themselves. Comics like those found on newspapers and children magazines are funny because their characters behaviour depends heavily on emotions, which is why this model is well suited for this application.

This project implements an emotion-based model for agent behaviour in a way that tells a story in the form of comic strips. For this, the model is adapted to a discrete time form since the actions no longer occur in real time (like in traditional simulation games) but rather in a sequence of frames or panels. The model is inspired by the analysis of time and space mechanics in comics by Scott McCloud. The emotional model is also adapted to reflect the rather extreme emotions and responses that characterize cartoon characters.
Acknowledgments

This thesis would not have been possible without the guidance and inspiration provided by my supervisor and good friend Mikael Kindborg. His work and ideas were absolutely invaluable not only as an innovative academic resource but as a strong reminder that there is so much more to computer science than just computers and science.

I would also like to dedicate this work to my parents that supported me throughout my time away from home.

Finally I would like to thanks all the new friends that accompanied me during my time in Sweden, their friendship made this experience complete.
# Table of Figures

| FIGURE 2.1: EARLY WORDS | .............................................................................................. | 4 |
| FIGURE 2.1: EARLY WORDS | .............................................................................................. | 5 |
| FIGURE 2.3: THE SAME PANEL AS IN THE PREVIOUS FIGURE BUT NOW SEPARATED IN SEVERAL PANELS THAT SHOW SUCCESSIVE EVENTS IN TIME | .............................................................................................. | 5 |
| FIGURE 2.4: A REALISTIC COMIC (LEFT) AS DRAWN BY COMIC ARTIST SUSANA VILLEGAS AND A CARTOONY COMIC (RIGHT) AS DRAWN BY COMIC ARTIST JOAQUIN CUEVAS | .............................................................................................. | 6 |
| FIGURE 2.5: “SURPRISE” AS PORTRAYED BY TEX AVERY’S CLASSIC CHARACTER BUTC | .............................................................................................. | 7 |
| FIGURE 2.6: GAME COMPLEXITY AND GENRE | .............................................................................................. | 8 |
| FIGURE 2.7: FUZZY F.L.A.M.E. MODEL | .............................................................................................. | 12 |
| FIGURE 2.8: IMPROVISATIONAL PUPPETS MODEL | .............................................................................................. | 13 |
| FIGURE 3.1: BASIC AGENT ARCHITECTURE | .............................................................................................. | 16 |
| FIGURE 3.2: A SINGLE PANEL SHOWING SEVERAL EVENTS AT THE SAME TIME | .............................................................................................. | 17 |
| FIGURE 3.3: SEVERAL PANELS SHOWING SEVERAL SUCCESSIVE EVENTS | .............................................................................................. | 17 |
| FIGURE 3.4: GRAPHICAL REPRESENTATION OF AGENT EMOTIONS | .............................................................................................. | 19 |
| FIGURE 3.5: AGENT BASIC STATE MACHINE | .............................................................................................. | 20 |
| FIGURE 3.6: AGENT’S MIND CYCLE | .............................................................................................. | 21 |
| FIGURE 3.7: FLOWCHART – SENSOR SWEEP | .............................................................................................. | 22 |
| FIGURE 6: FLOWCHART – AGENDA MANAGER | .............................................................................................. | 22 |
| FIGURE 3.8: FLOWCHART – SCHEDULER | .............................................................................................. | 23 |
| FIGURE 3.9: FLOWCHART – EXECUTOR | .............................................................................................. | 24 |
| FIGURE 3.10: STATE DIAGRAM – AGENTS COMMUNICATION STATES | .............................................................................................. | 25 |
| FIGURE 3.11: FLOWCHART FOR THE MODE CHANGING PROCESS IN THE SENSOR SWEEP PART OF THE MIND’S CYCLE | .............................................................................................. | 26 |
| FIGURE 3.12: CONTROL PLAN WHERE BEHAVIOURS ARE ORGANIZED AND EXECUTED ACCORDING TO A HIERARCHICAL TREE STRUCTURE | .............................................................................................. | 27 |
| FIGURE 3.13: SIMPLE CONTROL PLAN STRUCTURE WHERE BEHAVIOURS ARE EXECUTED LINEARLY ONE AFTER THE OTHER | .............................................................................................. | 28 |
| 3.14: AGENTS PERFORMING SEVERAL BEHAVIOURS DEPENDING ON THEIR MOODS AND LEVEL | .............................................................................................. | 29 |
| FIGURE 3.15: PAGES AS ORGANIZED IN A TYPICAL COMIC BOOK | .............................................................................................. | 29 |
| FIGURE 3.16: A PAGE AS PROPOSED BY SCOTT MCCLOUD IN THE FORM OF AN “INFINITE CANVAS” | .............................................................................................. | 30 |
“When people told themselves their past with stories...
Explained their present with stories...
Foretold their future with stories...
The best place by the fire was kept for the storyteller”

- Jim Henson’s “The Storyteller” -

1.1. Background

Comics have been among us, in one form or another, for a very long time, their influence in our culture is undeniable. In later years, Hollywood has helped to increase interest in this form of art by producing quite a lot of movies based on classic comic book heroes like Superman or Spiderman.

This renewed interest in comics is a great opportunity to give them new life. Comics have, despite all the advances in technology and media distribution, kept their original form intact since their beginnings as static images with static text. Because of this, I consider that exploring the possibilities of this popular media a very important task.

In this project I propose a new form of comics in which the stories are no longer a succession of static images that have no relation with one another. I show an adapted model for dynamic comic creation in which comic character can actually act and interact with the comic creator and each other.
1.2. Objectives

- Elaborate a time-discreet model for agent interaction that can be used in a comic strip generation application.
- Create a comic strip generating application prototype that uses the aforementioned time-discreet model.
- Use the application to replicate pre-existing scenarios of agent interaction

1.3. Research questions

- Which model or models are suitable as the basis for comic strip generation?
- How can the chosen model be applied to an application that generates comics?
- What kind of fun-potential does the resulting application have?

1.4. Contributions

- For comic artists and artists in general, this project shows how computer science and technology can provide a completely new approach to a classic art form such as comics.
- For game developers, this project presents the basic framework and ideas for a kind of game, or toy, where the story and game-play are blended into one through a new approach to dynamic storytelling.
- This project serves as a review and further research in the field of agents as improvisational actors, a field left rather unexplored since the late 90s.
As the basis for this thesis I consider important to take a look at the main subjects treated throughout this work which are storytelling, comics, and software agents.

2.1. Storytelling and comics

2.1.1. Storytelling

Storytelling is the ancient art of conveying events using words, images and/or other media. It has existed since humanity has had languages. Stories have been shared in every culture and in every land for many different purposes that range from entertainment and education to preservation of culture and instilling moral values.

Storytelling comes to us so naturally that we don't even think about it when we recount some experience we have had. We also consume stories constantly, both fictional ones, through novels or movies, and non-fictional ones through books, documentaries and the news media.

So what is a story exactly? A Story is defined as a "narrative"; a construct created in a suitable medium (speech, writing, images) that describes a sequence of events fictional or otherwise. (Sto08)

Stories can be interesting to us all but they are especially appealing to children, they can be easily transported to different places and times. Children love listening to tales read to them, watching stories or creating their own. Rarely do children get tired of hearing stories over and over again.
Asides from being a valuable source of entertainment for children, stories can also have educational benefits. Stories enable children to share their experiences and show their creativity. It can also be a rewarding social experience as they can exchange ideas, and increase their communication skills. (Hayes-Roth, et al., 1995)

### 2.1.2. Comics

Comics have been with us for a really long time, maybe not as we know them today, but the idea was similar nonetheless. The earliest words actually looked much like a kind of comic strip (McCloud, 2003). See Figure 2.1. for an example of Egyptian hieroglyphs, here the “characters” are actually a closer representation of the actual objects than real letters.

![Figure 2.1: Early words](image)

Though any person can tell stories, there are many individuals that have brought this skill to the level of art through a variety of means and in a wide range of forms and media. Comics are such an art form. In this particular case, the narrative is conveyed using images and words.

In Understanding Comics (McCloud, 2003), comics are defined as “juxtaposed pictorial and other images in deliberate sequence, intended to convey information and /or to produce an aesthetic response in the viewer”. There are several different definitions for comics and even if McCloud’s definition
technically excludes single panel illustrations we can understand that a single image can very well tell us a whole story.

The building blocks of comic are then, the previously mentioned illustrations, which are organized in panels, and the texts that accompany them. Panels are not only containers for images but can actually convey much information about what happens inside the comic. They can even show us how time behaves or how to go through the story. Figures 2.2 and 2.3, which are from Scott McCloud’s book (ibid.), show examples of this; panels tells us how time behaves within comic strips.

Figure 2.2: A panel showing a single moment in time with several characters.

Figure 2.3: The same panel as in the previous figure but now separated in several panels that show successive events in time.
2.1.3. Comic characters

Comics as any other narrative tool can have any kind of story with all sorts of different characters. For this project, I will focus on the kind most commonly associated with comic strips, those caricatures and little funny cartoons that most commonly appear on newspapers or children’s books, see Figure 2.4. (Villegas, 2007) (Cuevas, 2008)

Figure 2.4: A realistic comic (left) as drawn by comic artist Susana Villegas and a cartoony comic (right) as drawn by comic artist Joaquin Cuevas

The term sometimes used for this style of comics is “cartoony”, since it is not a realistic representation of people, things or palaces and has a humorous connotation in its representations and themes. (Com08)

These “cartoony” comics are useful to us because of their appeal to children as part of stories. Cartoony characters can make story ideas and characters idiosyncrasies very clear because of its ability to exaggerate actions and expressions. This exaggeration makes a story more dramatic, and in doing so, can effectively illustrate a point. In Figure 2.5, a character expresses fear with exaggerated features (Porcel, 2006).
There has been extensive research on children and how they use puppets to tell stories. While using them, storytelling becomes and interactive and multisensory experience and the anonymity gained by the child by using the puppet as a tool for story crafting allows her to have more freedom to experiment and improvise. I believe this same principle can be applied to drawn representations of puppets in the form of comic characters (Duran Huard, et al., 1996). We will come back to the topic of agents in Section 2.2.

2.1.4. Comics and programming

Comics and programming might not be topics that are usually connected with each other, but they share more characteristics than most realize. Comics are basically the representation in static form of dynamic content, just like the source code of a program is the representation of its runtime result. This is why comics have been successfully used in the field of programming as a tool that can help to bridge the gap between the program and its result. Comics posses several signs that can represent event based programs that involve characters. Also, children testing this programming environment commonly conceived their
programs as linear stories which we can use to our advantage. The fact that this visual event based programming resulted in a possibly useful way for creating interactive stories is actually one of the findings of that research (Kindborg, 2003).

The central part in the comic-based programming model is the use of “agents” as the components that would be programmed by the player. These agents represent the character and objects that appeared in the children’s programs. In our case, agents once again become the most important part since stories are the narration of the actions and events that happen to the agents.

As we will see further on, by reversing this logic, instead of using comics to program agents, a comic generator can be created by using already programmed agents to create comics.

2.1.5. Interactive storytelling

Interactive storytelling is said to be the single most hotly debated issue not only among people who study narrative but among people who study games and play.

On the field of video games, fictional stories are usually used in a way that goes beyond the events of the games themselves. Game designers add stories to enhance a gamers’ entertainment value.

The inclusion of stories in games is a widely disputed topic, since some gamers want just the game-play experience and others enjoy a good story to come with it. The degree to which a game needs a story can be appreciated in Figure 2.6. (Adams, et al., 2007).

![Figure 2.6: Game complexity and genre](image-url)
As we can see, games can actually be categorized according to the way they use the element of storytelling.

But what about games where the storytelling is not just an addition to the gameplay but the gameplay itself? In this case I would consider such “game” as a toy rather than an actual game. At this point I think it’s important to differentiate the characteristics of a game from those of a toy.

2.1.6. Games and Toys

As defined in Fundamentals of Game Design, “A game is a type of play activity, conducted in the context of a pretended reality, in which the participant(s) try to achieve at least one arbitrary, nontrivial goal by acting in accordance with rules” (Adams, et al., 2007).

From the previous definition we can see that there are two important characteristics that define a game, as opposed to a toy, these are rules and goals. A toy does not come with any specific rules about the right way to play with it nor does it has any particular goal for the player to achieve. For example, soccer is a game whereas a football is a toy.

From now on, I will refer to the person or user that interacts with the system as the “player”.

2.2. Social agents

2.2.1. Agents and social agents

In computer science an agent is a piece of software that acts on behalf of a user or other program, it is in this sense a representative; something that represents something else or someone else, like a sign in semiotics. In the context of this project, agents are embodied in pictorial form to represent the characters in stories.

Another important characteristic of agents is that they are not strictly invoked for a task but are active themselves. They have the ability to act autonomously and to communicate with other agents.
A social agent is an agent that acts in a social domain. Social domain is understood as being defined by phenomena found in a society, concepts that form and are formed by the individuals that form a society.

“So in summary, a social agent is and embodied entity that can express emotions and whose acts and relations are related to a social domain” (Kindborg, 2003).

2.2.2. Emotions and Emotional Agents

Emotions have proven to lead an important role in human intelligence. It is said that the complexity in human behaviour is largely a reflection on the complexity of the environment in which a person is in. Emotional agents are actually a result of the research on the field of intelligent agents. In the search for a human intelligence in a machine there have been attempts at simulating emotions since emotions play a crucial role in human cognition. The interaction between emotional process and the cognitive process may explain why humans excel at making decisions based on incomplete information. (El-Nasr, et al., 1998)

2.2.3. Emotional agent models

There are several models for emotional agent behaviour that can be mentioned but most of them have levels of complexity that might not be necessary in this case. I will first mention a couple of these as an example of models that deal with emotional agents, and then explain briefly the improvisational model which will be actually used as basis for this project and explained in more detail in the next chapter.

The Oz Project

This model, developed for high quality interactive fiction and virtual realities, has as objective the creation of dramatically interesting micro-worlds for the players that include moderately competent, emotional agents (El-Nasr, et al., 1998).

The architecture of agents on the Oz project is called Tok. The behavioural engine for Tok chooses an action to perform based on the agent’s physical
environment, its active goals, their importance and any action it was working on previously.

Its emotion model works on emotion types instead of trying to describe every possible emotion. This way, emotions are grouped depending on their common causes.

The emotions in this architecture can be grouped as follows (Bates, et al., 1992):

<table>
<thead>
<tr>
<th>Joy, Distress, Hope, Fear</th>
<th>Love, Hate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pride, Sham, Admiration, Reproach</td>
<td>Gratification, Gratitude, Remorse, Anger,</td>
</tr>
</tbody>
</table>

Emotions are then generated based on the success or failure of achieving goals and its desirability. This model uses Ortony et al.’s model to formulate the rules for the triggering process. Emotions are triggered with different intensities and this is then used to map them to Behaviours. (El-Nasr, et al., 1998)

This model serves as a base but might not be used directly due to the fact that the multiple emotions that it uses can actually be derived dynamically from one another or even represented by behaviours.

**The F.L.A.M.E. model**

This model is based previous models but fixes some problems not considered in them, for example, the binary (true or false) nature of the desirability of an event. F.L.A.M.E considers that an event can be desirable to some degree instead of being 100% desirable or 100% undesirable. (El-Nasr, et al., 1998)

The F.L.A.M.E model is based on three important concepts:

- **Fuzzy goals**: This introduces the degree of success of a goal
- **Fuzzy Membership**: This is the degree of membership of an event to a goal
• Fuzzy Mapping: Fuzzy mapping technique to map emotions to Behaviours

The emotional process is described in Figure 2.7.

![Figure 2.7: Fuzzy F.L.A.M.E. model](image)

In the end, this model has realistic emotional states that present a high granularity so it’s not suitable considering the characteristics of comic characters.

**Improvisational Puppets model**

This model uses a rather simple principle for its behaviour model, improvisation. This model follows the most important rule on improvisational acting which is to agree to anything any other actor says, one should never accuse anything anybody said to be false, instead, add on to it (Hayes-Roth, et al., 1999).

Improvisational actors create performances in real time without planning. This improvisation might not reach the artistic height of conventional theatre but
that’s not the point, what is important is to make the performance entertaining, and the audience usually enjoys it.

This model tries to create synthetic actors that can be embodied as animated characters and perform in a manner that resembles that of human improvisers. Characters never have and specific plan and they act in ways that reflect their moods. While interacting with other agents, improvisational agents just follow the rules of improvisation, as mentioned before, accepting automatically the other agent’s plan.

The architecture of the agents is shown in Figure 2.8. (Hayes-Roth, et al., 1995)

The agent’s architecture is basically separated in two parts, the mind and the body. The mind receives sensory data and controls all the behaviour generation and Control Plan management which are then sent as orders to the body.

The mind is also separated into three parts:
• Agenda Manager, which uses perceptual and cognitive events, states information and other information to instantiate Behaviours that are possible and relevant.

• Scheduler, which selects relevant Behaviours depending on the present situation.

• Executor, that sends the associated behaviour command to the mind body interface.

The Behaviours are organized in class hierarchies with a set of relevance conditions that can be satisfied by the occurrence of certain events. Behaviours are also selected depending on the mood of the agent and the “typical” mood of each behaviour.
“If a technological feat is possible, man will do it; almost as if it's wired into the core of our being.”

– Major Motoko Kusanagi, Ghost in the Shell –

The main objective in the creation of the application is to produce an application capable of reproducing the examples show in the Hayes Roth improvisational puppets model but in the form of comic strips. (Duran Huard, et al., 1996)

In this chapter I will first show the design and functionality of agents while explaining all the changes made to the original Hayes Roth model, and the concepts used as reference from Scott McCloud. In the last part I will explain the general organization of comics into panels and pages, also explaining their characteristics in the model and functionality inside the application.

### 3.1. Agents

#### 3.1.1. Basic agent architecture

As mentioned before, the basis for the agent architecture is Hayes Roth’s improvisational puppet model. In Hayes Roth’s model we can note the two basic components inside the agent’s mind which are the Agenda Manager, that generates appropriate Behaviours based on sensors and internal states, and the Scheduler which actually builds the list of actions the agent will execute at the end of its cycle (Hayes-Roth, et al., 1999).

The basic parts are shown in Figure 3.1. The generation of Behaviours by the Agenda Manager will depend upon Events coming from the environment and the Scheduler will in turn organize these and then execute them.
3.1.2. Adjustments over the improvisational puppet model

Although the model is very useful when used for storytelling, there are several significant differences between animations and comics that must be taken into consideration. The drastic change from animated images to action-by-action frames used in comics changes not only the overall structure and appearance of the application but the model itself.

Discreet Time

As described by McCloud (McCloud, 2003), time can be interpreted is several different ways in comics. Time does not depend on “time” itself, but on the shape and arrangement of the panels and also on the content of the panels. This is why time actually becomes a matter of space. Figure 3.2 and 3.3 show this.
The way agents perceive events will also change considerably due to the discreet nature or time. The agents in the improvisational puppet model are real time sprites that move in a simulated world and continuously take information input from its surroundings; by contrast, agents in the comic strip model will only receive input from its environment once in every panel, thus, making time a discreet measurement in such a way that the reasoning cycle of the agent happens before any action is taken.

This change in time perception will affect the agent’s mind model by changing the order in which processes are performed.
3.1.3. Agent architecture in detail

**Emotional model**

It is important to begin the detailed description of the agent’s architecture by describing the emotional model used. The model used as a basis is a simplified version of the one shown in the Oz project on its example “Lyotard the cat” (Bates, et al., 1992). The emotions considered represent those that seemed most interesting for the agents to base their actions upon. Comic characters usually don’t have a wide range of emotions since the usually drastic change from one mood to another is what makes them funny.

In this case I have considered the following emotions based on the most commonly found moods across several web comics

- Happiness
- Sadness
- Anger
- Fear

The emotions have also been put into a scale of 5 levels:

- “Not at all”
- “A little”
- “Normal”
- “Very”
- “Super”

This way, agents can have a reasonably wide range of emotions while remaining clearly separated with each other, as this table shows, all emotions are available in their 5 levels:
Agents will always act upon an Emotion-and-Level pair. A more realistic emotional model with finer granularity in levels and emotions was considered in the beginning but it was not suitable considering the nature of comics and cartoons. Cartoon characters are usually characterized by its extreme emotions, responses, actions and even mood swings. A realistic emotional change flowing smoothly from happy to sad or any other might not be as funny.

The graphical representation of emotions is obviously of great importance. Comics are mainly a graphical medium so images must transmit the emotion as clearly as possible. Figure 3.4 shows the cartoony representations of the emotions and levels for the agents.

<table>
<thead>
<tr>
<th></th>
<th>Not at all</th>
<th>A little</th>
<th>Normal</th>
<th>Very</th>
<th>Super</th>
</tr>
</thead>
<tbody>
<tr>
<td>Happy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sad</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Angry</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Afraid</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Figure 3.4: Graphical representation of agent emotions
Agent States

Before continuing with the agent’s mind cycle, it’s important to explain the states of the agent. These, similarly to those used by the panels, are a result of the time model used. Once the agent is created on a panel, it must calculate its new mood and actions depending on its surroundings and on the information it has from the previous panel. See Figure 3.5.

![State Machine Diagram]

**Figure 3.5: Agent basic state machine**

Mind cycle

Now that the emotional model is clear we can continue with the details of the actual mind cycle. Figure 3.6 shows a summary of the processes taking place in the agent’s mind cycle.
Figure 3.6: Agent's mind cycle

Sensor Sweep

The first step in the agent’s cycle is to run a sensor sweep, this step gathers all necessary information from the environment pervious to any actual “thinking” or decision making occurs. All emotional states are adjusted depending on the current condition of the environment and then the agent resolves the strongest emotion and it’s level. See Figure 3.7.
Agenda Manager

Once the strongest emotion and its level have been determined by the sensor sweep the Agenda Manager selects all possible behaviours from the behaviour library depending on the agent’s mood.

Behaviours in the Agenda Manager are selected depending on their compatibility with the current mood and its level. All behaviours in the behaviour library have a “common mood” and “common level” variables which indicate which mood and level does an agent needs to be in order to perform said behaviour. If the agent’s mood and level matches the behaviours criteria then it is added to the list of possible behaviours. See figure 3.8.
Scheduler

This is the most important step on the agent’s cycle. And its most important component is the Control Plan; the list of all Behaviours to be executed by the agent. It starts by removing from the existing Control Plan (if any), the Behaviours that are no longer compatible with the current agent’s mood and mood level.

The agent already has a list of possible Behaviours that came from the Agenda Manager so the next step is to remove from this list all Behaviours already in the current Control Plan.

Once all redundant Behaviours are removed from the “possible” list, this is mixed with the current Control Plan (if any). The result is the final Control Plan. See Figure 3.9.

Executor

The final step on the agent’s cycle starts by adjusting the graphical representation of the agent depending on its mood and level.

Once this is done the Executor must check if the agents is or not alone. If the agent is alone it will proceed to adjust its graphical representation depending on the first action listed on the Control Plan. If the agent is not alone the Executor checks its mode to see what kind of communication must be executed. See Figure 3.10. (For more details regarding agent communication see next section.)
Agent communication

This part is the first of two that deal with agent communication. The second part can be found in the section describing the Panel. Here I will explain the different states and modes of communication for the agents and then the different messages exchanged and how the work, see also Figure 3.11.

Communication modes

Listener mode: This is the default mode for any agent. In this mode agent can receive proposed Behaviours to follow or respond to greeting by other agents.
• Hello mode: This is the mode adopted by an agent after it is selected as a speaker and has to start a conversation with other agents. This mode also is adopted by agents when a greeting is received and a response is to be sent.

• Propose mode: Once the greeting have occurred, the chosen speaker has to propose an objective to the other agents. In this mode the agent selects at random and behaviour and proposes it to the rest. In this step the agent clears its Control Plan and adds the proposed behaviour to it.

• Ready to mode: This mode is the equivalent to the Propose mode but for agents that are not chosen as speakers; the agent clears its Control Plan and adds the proposed behaviour to it.

• Action mode: Both speaker and non speaker agents go into Action mode after they agree on a behaviour to perform. At this point all conversation stops and both perform the same behaviour as listed in their Control Plans.

![State diagram – Agents communication states](image-url)
Agents change from one mode to the next on the sensor sweep step of the minds cycle as shown in Figure 3.12

Figure 3.12: Flowchart for the mode changing process in the Sensor sweep part of the mind’s cycle

Messages

The messages exchanged by agents can be of three types, two of them used in the “Hello” state and one used to propose a behaviour among agents.

- “Hello” type: this is the message used by agents when starting a conversation. These messages contain text with some greeting like “Hello!” or “Hi!”
• “Proposition” type: this message is used to send let the other agents know what kind of behaviour is being proposed. These messages do not contain any text but the one on the proposed message.

• “Ok” type: just like the “Hello” type, these messages just carry the text used as a reply to a behaviour proposition. Messages will contain text such as “Ok!” and “Great!”.

Agent Behaviours

Behaviours, as some other parts of Hayes Roth’s improvisational puppet model are not explained in much detail so they are based on other models and adjusted to fit specific needs within the project.

Behaviours were originally thought in several other models as hierarchical tree structures where behaviour “B1” can be a group of Behaviours “B1”, “B2”, ..., “Bn” in such a way that agents go recursively through them in order to perform their control plan. In this way, the Control Plan would be assembled as shown Figure 3.13.

![Figure 3.13: Control plan where Behaviours are organized and executed according to a hierarchical tree structure](image)

This structure turns out to be far too complex for this model and presents us with no significant advantages. Our objective is to replicate the basic examples shown in the improvisational puppet model where the most important point is to implement the main rule of improvisational acting. A simple Control Plan where
Behaviours are executed linearly provides all the functionality necessary. This plan is shown in Figure 3.14.

![Control Plan](image)

**Figure 3.14: Simple Control Plan structure where Behaviours are executed linearly one after the other**

**Relation to the emotional model**

The relation behaviours and emotional model is of big importance since all behaviours are chosen according to the mood of the agent. All behaviours are related to one or several moods and its respective levels. This way, behaviours will be chosen for the Control Plan if it matches the agent’s mood and mood level range. See Figure 3.15.

![Relation between behaviours and agents](image)

**Figure 3.15: Relation between behaviours and agents**

**Graphical representation of Agents**

Agents are graphically represented by three main parts, see Figure 3.16. The first is the representation of the agent itself. The player can see which behaviour it is performing on each panel based on the cartoon and dialog of the agent. The second part is the agent’s emotional bubble, which shows the current mood and level of the agent. Finally there is the dialog bubble, which the agent uses to speak and to show the player what they are thinking.
Comic structure
Comics within the model are arranged in pages, which in turn contain panels, where the agents are placed to perform their behaviours.

3.1.4. Pages
Comics are an arrangement of sequential images (with or without text); the larger unit containing an arrangement of these images will be called Pages. Pages (as shown in gray in Figure 3.17) will contain all the panels that will in turn contain the agents acting on any given story.

Figure 3.16: Agents performing several behaviours depending on their moods and level

Figure 3.17: Pages as organized in a typical comic book
Comics are most commonly arranged in such a way that panels are read from left to right and strips from top to bottom. As there is no limitation as page size or panel layout on a display screen, panels can be arranged in any way by the player. Panels are connected by a line, as proposed by Scot McCloud (McCloud, 2003), to get a sense of logic to the overall story and sequence in which events occur. See Figure 3.18.

**Page 1**

![Graphical representation of Pages](image)

**Figure 3.18: A Page as proposed by Scott McCloud in the form of an “infinite canvas”**

**Graphical representation of Pages**

A page has one panel creation button and one button to move the created comic sideways, see Figure 3.19. In this way the user can pan around the comic.
3.1.5. Panels

The panels contain the actors and all the actual actions and events that happen to them. As described previously, panels can be arranged in any way the player find interesting and the lines connecting them will guide the reader through the story. See Figure 3.20.
Panels, just like agents, have to consider two states given the discreet nature of time within in the model; the initial state, with information coming from the previous panel and the current state, with all the changes made by the player or the agents, see Figure 3.21.

![Figure 3.21: State diagram of the Panel object](image)

Its “initial state” contains all the information coming from its previous panel, like the characteristics of its environment and list of agents. The “currents state” is the one that the player actually sees onscreen. This state shows the panel after any changes have been made to it either as a result of agent interaction or input from the player. Because of this, every panel has to keep all information from its initial state to calculate new “currents states” at any time. See figure 3.22.
Panels handles two main tasks: panel updates, and agent communication.

**Panel updates**

The updates happen when the player changes any characteristic of the panel or any of its agents; this also affects all subsequent panels, which will in turn update their agents. See Figure 3.23.

![Panel updates diagram](image-url)

**Figure 3.22: A series of panels illustrating their two states**

**Figure 3.23: Flow chart for the Panel Update process**
Design

Agent communication

Since agents are reloaded every time the player makes a change in the panel, the panel is the only object that remains constant and thus, needs to be in charge of managing the communication between agents. The panel, when updated, will check if communication between agents was already going on in the previous panel, if not, a new speaker is chosen and its message transmitted to the rest of the agents. See Figure 3.24.

![Figure 3.24: Flowchart for the Agent Communication process](image)

Graphical representation of Panels

The panels contain the agents and connections to the previous and next panels if they exist. They also contain the controls necessary for the player to alter the state of the panel and to create new agents. See Figure 3.25.
3.1.6. The Gutter

Another important aspect in comics, which is not treated in detail in this project, is the gutter, which is the physical space or distance between each panel where the imagination of the reader takes place and performs much of what is not seen on the actual images. The reader makes time pass, places change, and even events happen from one panel to the next. See Figure 3.26.

Figure 3.26: The gutter, space between panels
Graphical representation of the gutter

The gutter is editable by the player, see Figure 3.27. For this project the only change taken into consideration is the pass of time between panels.

Figure 3.27: Graphical representation of the gutter between panels
This chapter deals with the details of the computational model actually used in building the application. As in the previous chapter, I will first explain the classes involved with the Agents and their communication and behaviour and later the classes that add to the overall structure of the comics. The system was implemented in Flash (Adobe, 2005), and programmed in ActionScript, the programming language in Flash. Flash was chosen because of its ability to create dynamic user interfaces that run on the web.

**4.1. Partial class models**

**4.1.1. Agent**

The diagram in Figure 4.1 shows the Agent class with along with the classes necessary for agent communication and behaviours. Agents must consider an initial state and a current state to differentiate the information from previous panels and the new information gathered on the current one. Agents will mainly perform the routines necessary for the minds cycle: sensor sweep, agenda manager, scheduler and executor.
4.1.2. Message

Every agent has a message library containing all available messages, see Figure 4.2. The message type is used to distinguish between the different modes used to communicate greetings and behaviour proposals.

4.1.3. Behaviours

Behaviours contain the specific action they are representing and, in this particular case, the character image for the behaviour. Like messages,
behaviours are kept in a library object that is present in every agent. See Figure 4.3.

![Class diagram for the LibBehaviours and Behaviour classes.](image)

**Figure 4.3: Class diagram for the LibBehaviours and Behaviour classes.**

### 4.1.4. Partial computational model of the comic structure

The pages and panels form the overall structure of comic strips, and it is worth mentioning that not all classes will have a graphical representation nor do all graphical representations (sprites) have a class implementation, even if they behave like objects. The most important classes that do have an on-screen representation will be shown as inheriting this characteristic from the MovieClip class in ActionScript. See Figure 4.4.

![Partial class diagram for the classes that form the overall comic structure.](image)

**Figure 4.4: Partial class diagram for the classes that form the overall comic structure.**

### 4.1.5. Page

Pages are a holder for other objects; the visible objects inside the application will be the panels, so the role of the page is to keep track of them. When a new
panel is created, the page will tell each new panel the identity of its predecessor. See Figure 4.5.

![Figure 4.5: Class diagram for the Page Class](image)

4.1.6. Panel

Panels, like agents, must consider an initial state and a current state. Panels will also take care of initiating agent communication. An id for the next and previous panels is also necessary to cascade updates to later panels and to retrieve past information respectively. See Figure 4.6.

![Figure 4.6: Panel class diagram in context with the Page and Agent classes](image)

4.1.7. The gutter

The gutter will simply take care of updating the time difference between two panels. If panel A is the first panel and B is the second, the gutter on Panel B will contain the time difference between them, see Figure 4.7.
4.2. Complete class model

Figure 4.8 shows the complete class model, as implemented in ActionScript with the component parts previously explained. I can now show the overall model class diagram containing both the classes that form the overall structure of the comics and those that belong to the agent.

Figure 4.8: The complete class diagram for this model. It shows all the classes that form the overall structure of the comics as well as those that belong to the agents.
Chapter 5

Results

“You are an imperfect being, created by an imperfect being. Finding your weakness is only a matter of time.”

— The Borg Queen, Star Trek —

I will now discuss the results from the implementation of the modified Hayes Roth model and then I will speak briefly about the feedback from the test players.

5.1. Comic generation

As mentioned in beginning, the main objectives are to create a comic generating engine and to demonstrate its usefulness by replicating examples previously existing on in Hayes-Roth’s improvisational puppets model (Hayes-Roth, et al., 1995).

To begin with, the system can generate strips where agents take actions depending on their moods and surroundings as shown in the examples in Figures 5.1 to 5.3.

Figure 5.1: Comic strip example with a “spooky” environment
In the following part, I will show the behaviours replicated from Hayes-Roth’s examples (Hayes-Roth, et al., 1999). I used the first interaction example between a large character (Character 1) and a small character (Character 2). Their behaviours are shown in Figure 5.4, taken from Hayes Roth paper Directed Improvisation by Computer Agents. (Hayes-Roth, et al., 1999)
In the first part (Step 1) we can see both agents’ moods as the starting point. In the next step, both agents decide to play for a while (Step 2) until communication is established (Step 3). Once Character 2 notices Character 1, it proposes an activity (Step 4), in this case “playing”. After Character 2 agrees to it (Step 5), both perform the agreed behaviour as a final step (Step 5). All this activity is rendered onscreen through continual animation as show in Figure 5.5.

Let’s now take a look at these same steps (1 to 5) as performed by the characters in the comic strip model, see Figure 5.6.
Step 1: Characters evaluate their behaviours according to their moods. Based on its mood the agent in this example that it will “play”, “sing” and “whistle”.

Step 2: As there is no one else present in the panel yet, the agent performs its control plan normally, starting with the “play” behaviour.

Step 3: As a new agent appears in the panel, they both start to communicate by sending greetings.
Step 4: One of the characters proposes an activity to which the other agent agrees, in this case to “eat”.

Step 5: Both agents performed the previously agreed behaviour.

Figure 5.6: Steps in a comics strip example

These steps are of course presented as a continuous comic strip, which is shown in Figure 5.7.

Figure 5.7: Comic strip example
5.2. Feedback from players

As a way to get feedback on the prototype, it was uploaded to DeviantArt.com (DeviantArt, 2008) and the players (ages around 15 years old) were asked to and answer two questions:

- What do you think about the comic generator? Is it fun? (considering of course that is just a prototype)
- What would you add to it?

The first question was thought as a way to verify if the prototype was actually appealing to players. The second question was useful to get the common factors that could be improved on the prototype. I considered the answers given by 8 test player.

The fun factor

As for the first question, regarding the “fun factor” the players thought the toy was indeed entertaining and the fact that the characters create their own stories (and that they didn’t know what was going to happen) kept them playing for a good period of time. This fact also reassures the brief research done on randomness (explained on the next chapter) as the players develop a lot of interest when unexpected event are presented to them.

Possible improvements

Regarding the possible additions to the program; players agreed that the agent creation should be more personalized. Almost all the players mentioned that they wanted to create their agents with more unique characteristics than just their moods. The creation of personality archetypes for the agents was suggested as a good way to differentiate each agent.
“Imagination is more important than knowledge”

— Albert Einstein —

This project presents the first step into a possibly wide range of applications, specifically games and toys that include comics as its main component. Since the development of applications similar to this is practically non-existent, there is a vast field of further research and improvements.

Based on my own experience with interactive media and the feedback received from players who used the prototype, I propose four main fields where further research could be carried on; artificial intelligence and in-game options, graphic design, comic structure and humour.

6.1. Artificial intelligence and in-game options

AI is of course the basis for this kind of intelligent-agent-based application. For this project the intelligence model has been adopted from previous related work for its simplicity and effectiveness but more complex models that can consider a wider range of variables and factors would without any doubt help produce stories more appealing to players.

As for the in-game options, the most common suggestion among the test players was the personalisation of the agents. The creation of personality archetypes or character roles like “The bad guy” or ”The hero” can be applied in the form of emotional filters in the agent cycles as a way to guide their actions throughout the story.
6.2. Visual Style

In this thesis, the graphic design of the characters was not considered in any detail yet it is a central part since this application is targeted to people with some pre-existing knowledge or ideas as what is a comic, how it looks and works. This project presents a simplistic representation of characters and sceneries approaching those presented in the models used as reference. A more developed visual style can help to widen the audience from kids to more adult audiences. Different visual styles can also serve to change the system’s purpose from a toy to, for example, a learning tool.

6.3. Comic structure and storytelling

The comic structure in this project follows the most common standards for comic strips, sequential panels that tell about the actions of character. This linear characteristic (even in the infinite canvas (McCloud, 2003)) could very well be changed to nonlinear stories, branching stories, parallel or recursive storylines as seen regularly in videogames (Adams, et al., 2007).

Like computer games, a comic taking placed on an infinite canvas can produce storytelling model that are impossible to achieve in traditional media.

6.4. Humour

Humour is an integral part of comics (more usually than not) and always adds to any kind of story, yet there is no useful research on now it works at a practical level.

As a starting point, early on this project, I did a study on web comic humour. The main focus of the study was to investigate randomness as a source of humorous situations. Players were presented with a comic strip generator that would generate a couple of “alien” characters performing random actions and talking in a non-existing alien language, also randomly generated. See Figure 6.1.
This randomness was actually a very appealing factor to test players. Among the multiple actions that characters could do was the possibility to appear holding a small pink pig, which of course had nothing to do either with the comic or with the characters; this small detail was, according to almost all the players’ one the funniest parts of the strips.

This specific kind of jokes is an example of a clearly defined humorous tool that can be easily used and applied to dynamic comic strip generation.

The specification of more of these tools would be another important step towards further funnier automatic comics.
Conclusions

Chapter 7

Conclusions

“And where does the new born goes from here?... the net is vast and endless.”

– Major Motoko Kusanagi, Ghost in the Shell –

7.1. Thesis summary

Comics are an established and quite common approach to storytelling, and the visual language of comics has proven useful to describe events as “frozen animations”. Comic strips have been used as a graphical language to program the behaviour and actions of software agents by children.

Previous work with agent based systems and children have also shown how appealing it can be as a tool for interactive storytelling.

This work shows how to mix these two ideas into one; use the appealing language of comics and previously developed emotional agent models to create a comic generation engine that can produce comic strips similar to the stories generated by the agent models.

The comics strip examples generated by the system presented in this thesis are an example of how emotional agents can be applied to comic strip generating engine that has the potential to be an appealing toy.

7.2. Findings

This section provides a summary of the findings that relate to the research question presented in Chapter 1.
7.2.1. The agent model

There are many potentially useful agent models for this type of application, and a good selection criterion is their complexity; given the nature of comics and the stories told, Hayes Roth’s improvisational puppet model (Hayes-Roth, et al., 1995) is a very good choice. It provides all the basic functionality needed for storytelling and agent interaction while remaining simple in its implementation.

7.2.2. The application prototype

The prototype was able to generate comic strips that follow the structure and characteristics proposed by Scot McCloud (McCloud, 2003).

The prototype was also able to apply the adjusted version of the emotional model proposed by Hayes Roth’s, as well as reproducing the examples given for this model (Hayes-Roth, et al., 1999).

7.2.3. The fun factor

The application was also tested with several players that agreed that it was an entertaining activity. Even if the prototype is a rather simple implementation with limited options, players also agreed that with further development it could be a great toy.

7.3. Contributions

The contributions of this thesis can be appreciated differently from different perspectives.

From the point of view of the comic artist, or artist in general, this project is an example of how computer science and technology can provide a completely new approach to a classic art form such as comics. This is important since artists, more often than not, are prone to overlooking the use of new technologies as a way to further explore their artistic potential.
From a game development point of view, this project presents the basic framework and ideas for a kind of game, or toy, where the story and game-play are blended into one through a new approach to dynamic storytelling.

This project also serves as an inspiration for further research in the field of emotional agents as improvisational actors. This field of research was left rather unexplored in the late 90’s, even though it presents several possible uses as the basis for games and toys and should be explored as such further.
Chapter 8

References


Bates Joseph, Loyall Bryan and Reilly Scott An Architecture for Action, Emotion, and


Duran Huard Ruth and Hayes-Roth Barbara Character Mastery with Improvisational

Duran Huard Ruth and Hayes-Roth Barbara Children's Play with Improvisational Puppets

Ellington Henry, Addinall Eric and Percival Fred A Handbook on Game Design [Book]. -
1982.

Elliott Clark Hunting for the Holy Grail with "Emotionally Intelligent" Virtual Actors


El-Nasr Magy Seif and Skubic Marjorie A Fuzzy Emotional Agent for Decision-Making in

El-Nasr Magy Seif and Yen John Agents, Emotional Intelligence and Fuzzy Logic [Book]. -
1998.

El-Nasr Magy Seif, Iorger Thomas R. and Yen John Learning and Emotional Intelligence


På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Uphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© [Författarens för- och efternamn]