Re-using Digital Narrative Content in Interactive Games

Journal Item

Re-using Digital Narrative Content in Interactive Games

Annika Wolff[1], Paul Mulholland[1], Zdenek Zdrahal[1], and Richard Joiner[2]

[1] Knowledge Media Institute & Centre for Research in Computing, The Open University, Milton Keynes, MK7 6AA, UK
[2] Department of Psychology, University of Bath, Bath, BA2 7AY, UK

Abstract

This paper presents a model, called Scene-Driver, for the reuse of film and television material. We begin by exploring general issues surrounding the ways in which content can be sub-divided into meaningful units for re-use and how criteria might then be applied to the selection and ordering of these units. We also identify and discuss the different means by which a user might interact with the content to create novel and engaging experiences. The Scene-Driver model has been instantiated using content from an animated children's television series called Tiny Planets, which is aimed at children of 5-7 years old. This type of material, being story-based itself, lends itself particularly well to the application of narrative constraints to scene reordering, to provide coherence to the experience of interacting with the content.

We propose an interactive narrative-driven game architecture, in which a user generates novel narratives from existing content by placing "domino" like tiles. These tiles act as "glue" between scenes and each tile-choice dictates certain properties of the next scene to be shown within a game. There are three different game-types, based on three different ways in which tiles can be matched to scenes. We introduce algorithms for generating legal tile-sets for each of these three game-types, which can be extended to include narrative constraints. This ensures that all novel orderings adhere to a minimum narrative plan, which has been identified based on analysis of the Tiny Planets series and on narrative theories. We also suggest ways in which basic narratives can be enhanced by the inclusion of directorial techniques and by the use of more complex plot structures. In our evaluation studies with children in the target age-range, our game compared favourably with other games that the children enjoyed playing.

Keywords: interactive narrative, content reordering, interface, AI planning algorithms

1. Introduction

This paper presents a model, called Scene-Driver, for the re-use of film and television content to create novel and engaging interactive narrative experiences. Broadcast content often produces spin-off merchandising, particularly when the content is aimed at children. Examples of spin-offs include figurines, soft toys and items such as clothing or pencil-cases with logos and characters from the film or television broadcast. Digital products such as games and interactive DVDs are another type of spin-off that are becoming ever more popular. A feature shared by all these different types of merchandise is that they re-use the original content in some form. The simplest form of content re-use involves simply re-supplying the content in its original state. For example films often go from being shown in the cinema, to being made available on video and DVD and then to being shown on the television. Television series that have proven to be popular are often re-run one or more times. More complex types of re-use include compilation episodes, particularly common with comedy television series, or "outtake" shows, where mistakes that were not included in the original broadcast are shown. These often include clips from several sources. Games and DVDs regularly include "viewing galleries" where the user can select favourite scenes to view. These types of re-use require the sub-division and re-ordering of the original content.

We have been working with Pepper's Ghost Production Company who produced a popular animated children's television series called Tiny Planets that has been shown repeatedly on a major broadcasting channel in the United Kingdom and also syndicated worldwide. There are 65 episodes in the Tiny Planets series, each of which focuses on the adventures of two alien creatures, Bing and Bong, as they travel amongst a group of six "Tiny Planets" interacting with the local inhabitants of each planet and solving problems. Each of the Tiny Planets has an individual theme, such as *sound* or *technology*, which is reflected in the subject matter of an episode set on that planet. Planets may also have characters called *locals,* who are geometrically shaped characters with eyes and *flockers*, which are bird-like creatures. Some planets also have their own *robots*. Figure 1 shows some of the characters from the series. As well as the entertainment value, each episode also has some educational content which is relevant to the planet on which it is set. For example, an episode set on the Tiny Planet of Technology may explore using levers and fulcrums to move heavy objects (see Figure 2), whilst an episode taking place on the Tiny Planet of Light and Colour might investigate the differences between two-dimensional and three-dimensional drawing techniques. Given that the target age-range of the series is children of 5-7 years old, these educational aspects are intentionally presented in a fun and light-hearted, yet accurate, manner.

[Figure 1]

[Figure 2]

In developing the Scene-Driver model for content re-use, we have drawn on aspects of compilation episodes (which have thematic similarities across clips from a single programme) and viewing galleries (where the user has agency in the selection and ordering of scenes to view). Since the model is implemented and tested using content aimed at children, there was the requirement to ensure that the model was both usable and enjoyed by children of the target age-range. However, the model was also developed with the view that it should be flexible enough to allow alternative content aimed at different target end-users to be utilised in place of the Tiny Planets material.

To re-use content, we must first sub-divide it into individual units, describe it according to thematic principles and then choose a criteria that can be applied to the ordering of these units. For the Tiny Planets content, we apply narrative criteria to the sub-division of content into "story units" and to the subsequent re-ordering of these units to create a novel narrative from the existing content.

Narrative principles are particularly suitable for content that is itself originally story-based although our underlying model is generic to allow for the fact that other types of content may lend themselves to the use of alternative scene ordering criteria. A further benefit of narrative is that it provides coherence which aids understanding (Schank, 1990). Additionally, Thorndyke (1977) found that if a story does not conform to an expected story grammar, for example if the theme is introduced later in the story or not at all, then subsequent recall of the story is adversely affected. With the advent of interactive digital television it was initially assumed that it would, at least eventually, be possible to allow a viewer to have interactive freedom in the creation of novel narratives via this medium. In this scenario the user's choices and actions within a story would influence the ultimate outcome. It soon became apparent, however, that in order to implement even the most simple ‘branching narrative’ model (where a user’s choices either enable navigation through alternative paths of a single narrative or else enable the generation of different narratives) would necessitate the creation of prohibitively large numbers of new resources as the number of choices a user was allowed to make was increased (Brooks, 1996; Murtaugh, 1996). Such resources can be expensive to create. One obvious solution is to sacrifice the quality, to some extent, in order to be able to provide the required quantity of new resources at a lower cost. The alternative would be to limit the number of choices a user is permitted to make throughout the narrative, but this effectively diminishes the notion of interactivity. An even more idealistic model of interactive narrative, as explored in *Hamlet on the Holodeck* (Murray, 1997), would have the user totally immersed within the story to the extent that they would ‘become’ a character within the narrative. Their story persona could fully interact with and engage in dialogue with other characters and this interaction within the story world would have a direct effect on the course of the narrative. This scenario would necessitate the complete modelling of believable characters who could react in a realistic manner to any event that might be instigated by the user. It would also have to be possible to dynamically alter plot-lines to accommodate user-actions whilst maintaining narrative coherence, as well as other important aspects of a good narrative such as “pacing” and “suspense” (Ryan, 2001). Even if this were achievable with current technology it is not clear whether such an environment would work for many stories, e.g. those in which it is essential that the user's character acts in a certain manner in order for the integrity of the story to be maintained. Given complete agency, the user is free to act out their role however they choose (Bernstein, 2001; Adams, 1999). For these reasons, we look at approaches to the creation of interactive narratives that can be more easily implemented. Our model aims to find a balance between re-using content using narrative principles to provide coherence and allowing user agency during narrative construction to enable interactivity without compromising the narrative flow. Conceptually we look to re-use elements of plot, for the purpose of applying narrative principles and to retain directorial principles in order to enhance visual aspects of the model and provide continuity between scenes.

We wanted to include the user as an active participant during the process of scene reordering, rather than as a passive viewer of a pre-generated story. Given the age-range of the target viewer of Tiny Planets we considered that the experience would benefit from being playful, to make it more engaging and entertaining. We also wanted to retain the educational aspects from the original series.

We have used the Scene-Driver model to implement a Tiny Planets demonstrator in the form of an interactive narrative game. This game re-uses the content from the Tiny Planets television series and is aimed at 5-7 year old children, in the target age-range for the series. We have developed an icon-based “domino-like” interaction method for playing the game. This interaction mechanism was derived from looking at ways in which transitions could be made between units of content, based on the properties that changed between the two units. It has the benefit of being straightforward for children of the target age to use. This demonstrator has been evaluated with children in four studies, throughout the course of its development. These studies have yielded very positive results, showing that the children can effectively use the domino-like interface and that they rated the game highly in terms of enjoyment.

On a more general level, our approach (though not necessarily the specific interaction mechanism used in the demonstrators described here) has the scope to be applied in a number of different ways, for example with different media types, with different age-ranges and on different platforms such as PC, web-based applications or DVDs. Given the possible convergence of these technologies, such as in the form of Web TV, it is of benefit to look at the potential to re-use the content across platforms.

In this paper, we start by introducing related work which has influenced the design of Scene-Driver. We follow this by outlining our general approach to content re-use and our development of an interface for interacting with content, based on which we then describe algorithms for generating interactive games with existing content. We subsequently describe the application of narrative principles to extend the model and explain how all the above concepts have been incorporated into a demonstrator, using content from the Tiny Planets television series. Finally, we detail the results of four evaluation studies, which were conducted in order to assess the Scene-Driver model.

2. Related Work

The design of our model is influenced by research into several different areas, such as models of interactive narrative, systems for re-using multimedia resources and interfaces for narrative interaction.

Here, we discuss previous research relating to these areas and within the paper we refer back to this work to denote how it is related to the design of our own model.

2.1 Narrative and Story

Before exploring prior research into interactive narrative, it is useful to make a distinction between *narrative* and *story*. Whilst researchers differ in their exact use of terminology, structuralists agree that narrative and story are independent of each other. The story is a collection of facts (such as events, actions, characters, etc.), whereas the narrative relates to the particular way in which these facts are arranged and conveyed to a reader or audience (Genette and Lewin, 1983; Brooks, 1996; Chatman, 1978; Szilas, 1999). In this way, the same story can be narrated from alternative viewpoints and via different media, with different facts included and omitted each time to produce different effects. However, the set of facts which constitute the story itself remain unchanged.

Narratives provide coherence by arranging story facts in certain expected patterns. A common pattern is the classic three-act plot-structure of a film or novel, of which many variations exist, the most basic example involving scene-setting and introducing characters, the introduction of a conflict involving the main protagonist of the story and the subsequent resolving of this conflict. However, the chronological order in which story events occur and the order in which they are narrated need not be the same (Genette and Lewin, 1983). Altering the temporal ordering of events in the narrative, so that it deviates from conventional plot ordering, is often used to produce specific narrative effects. These temporal distortions are usually clearly marked in the narrative in order to enable the reader/viewer to reconstruct the story in their mind in the correct temporal order. One example of changing the temporal order is of a murder mystery that starts with the discovery of a dead body, but then goes backwards in time to start telling the story leading up to this event and then goes forward from this point in time to tell of the events surrounding the solving of the murder. This technique is designed to engage a reader/viewer's interest from the start of the narrative. To further illustrate how narrative and story differ, if the temporal ordering were to be changed so that the discovery of the body was presented in the middle of the narrative (in accordance with the classic plot-structure which first introduces theme and then conflict) this would not in any way alter the story that was being told. Other common temporal distortions include altering the pace of the narrative by speeding up and slowing down events, or by omitting events altogether (e.g. moving directly from evening to the next morning without reference to the night time).

2.2 Narrative and Interactivity

According to the distinction made in the previous section between narrative and story, we take the term 'interactive narrative' to apply to interaction that occurs at the narrative level and not at the story level, i.e. how the user might influence the selection, ordering and presentation of a set of pre-existing story facts, whilst the story facts themselves remain unchanged by user actions. Interactive narratives can therefore be interpreted in terms of how the reader’s actions effect the ordering of events in the narrative. For example, Walker (1999) uses Genette's theory of narratology to interpret the interactive narrative "afternoon". She focuses on how the order in which story events are recollected for the reader differs from the actual ordering in which events occurred at the story level and the effects of this piecing together on the reader.

Some models of interactive narrative also incorporate *interactive story*, whereby the story content itself is dynamically generated in response to user action. This leads to a distinction between models of interactive narrative in terms of whether the model utilises existing story content (e.g. Murtaugh, 1996; Srinivasan 2002; Rocchi and Zancanaro, 2003; Bernstein, 2001; Hooper and Weal, 2005) or if the story content is dynamically generated (e.g. Hayes-Roth et al., 1995; Mateas and Stern, 2002; Cavazza, 2002). In this section we describe some of these previous models of interactive narrative that are most relevant to our work. We focus mainly on those models which use existing 'canned' content, since our own model is based on content re-use.

Murtaugh’s (1996) Automatist Storytelling System explores the possibility for dynamically generating personalised presentations. This model was developed as a response to branching narrative systems in which users can explore how different choices might cause the story to unfold in different ways. Essentially, such systems do not store any history of the user’s interaction and so the user cannot have any real effect on the story, since these choices are all pre-authored and hard-coded. By contrast, the

Automatist Storytelling System uses keyword annotation of content, along with narrative and editing principles encoded by a *narrative engine*, to dynamically alter a narrative in response to user action. Thereby, a user can have a genuine real-time influence in constructing their own personalised presentation.

Murtaugh has used the storytelling system to create Contour and Dexter. These systems both present a user with a selection of multimedia resources, such as videos, documents, pictures, etc. that they might be interested in, based on their previous choices made within that particular narrative presentation. The user is thereby guided through the narrative whilst having the ultimate choice as to the materials they wish to view. To implement these models, it was necessary to devise schemes for subdividing the multimedia video content, to enable its description and also for presenting it to the user as discrete units. Brooks (1996) provides the following perspective on the desirable granularity of story-elements. The larger the story granules, the fewer ways they can fit together and the less reasoning required. Smaller chunks provide more variety in the ways they can fit together but also require more effort to describe the ways in which they can be re-ordered. Therefore, a balance needs to be struck bearing in mind the complexity required by the goal.

Murtaugh suggests that the following principles should be applied when deciding upon the granularity of units. Firstly, each video-clip should be self-contained and coherent within itself, to 'form a kind of story phrase', or *lexia* (Landow, 1992). Furthermore, in order to be dynamically editable with other pieces, the duration of the clip should be relatively short and should not cover too large a range of ideas. Within Contour and Dexter, the length of the video clips tended to be between thirty seconds and two minutes. In cinematographic terms, such a unit is equivalent to a scene. A scene is defined as "a complete unit of film narration. A series of shots or a single shot that takes place in a single location and that deal with a single action" (Monaco, 1977). From this point, we will refer to such units as scenes, but will adopt the term *event* instead of action, since an action can imply a short duration. According to Murtaugh's view, we take a scene to be any part of a story within which it is possible to easily identify and describe one key event.

Based on some of the principles of Murtaugh's automatist storytelling system, Srinivasan (2002) developed "Village Voice". The content for this system was a collection of stories in video format from the Somali community in Boston. The difference between Murtaugh's system and Village Voice is that rather than annotating clips with keywords, the video clips were described in terms of an ontology constructed by the Somalian community.

The work of Rocchi and Zancanaro (2003) includes directorial techniques in the presentation of narratively structured graphical material. Annotated images are chosen from a library to reflect the content of a verbal commentary. The system then provides a plan structure for synchronising the images with the audio soundtrack, including suggestions as to which sort of transition should be used between one image and the next. Transition properties include camera angles, camera movements, such as tilt and panning, and also conventional editing techniques, such as whether to fade between scenes or cut from one to the other with a moment of black screen in between. This work suggests the benefits of introducing directorial techniques for supporting transitions between scenes in order to maintain visual continuity and narrative flow.

In "Card Shark and Thespis", Bernstein (2001) describes sculptural hypertext narrative based on playing cards. A sculptural hypertext is one in which lexia are densely interconnected. As the space is navigated, connections that are not relevant to the path that the reader is taking through the lexia are cut away. In "Card Shark", a player is provided with a set of cards, each of which contains some story text and possibly some constraints regarding when it can be played. The player can only choose amongst cards for which the relevant preconditions are met. To this end, the current story state is represented on a blackboard. Each card asserts to and retracts from the blackboard to reflect events occurring in the story segment. Therefore as a narrative is played out, the playable status of the remaining cards changes depending on their compatibility with the current-story state. If the player holds no legitimate cards, then they are at a dead end and the story is over. In order to ensure that this situation does not arise before the story has had a chance to play out, Bernstein introduces *transitions* that modify the environment so that certain cards can be played that will advance the story. "Thespis" extends Card Shark by allowing computational agents to participate in the story, each agent being in possession of its own set of cards from which it can choose possible actions to take.

Hooper and Weal (2005) have developed StorySpinner. Similar to Card Shark and Thespis, this is a card-based interface which can be used to navigate hypertext narratives. In this case, the cards take the form of tarot cards. Each card has a number of potential meanings, depending on the context in which it is played. StorySpinner deals with the problems of dead-ends being reached before the story is complete by enabling the user to backtrack and make alternate card-choices. Other sculptural hypertext systems include the scenario authoring tools developed in the Ambient Wood project (Weal et al., 2003).

"Directed Improvisation" (Hayes-Roth et al., 1995) is a model of interactive narrative in which characters create a story within directorial constraints. Their test bed application CAIT (Computer-Animated Improvisational Theatre) is aimed at children. It is based around a set of 2 dimensional characters, each of which has a range of behaviours which they can exhibit either in an improvisational role (i.e. the behaviour is decided by the character) or else by direct instruction from the child. This provides flexibility in the level of "direction" by the child versus unanticipated, improvised behaviour exhibited by a character. The child can choose the level of instruction they want to give as well as whether they wish to interact with the system real-time or else build a script and "run" it. The system is intentionally playful and the characters are created to be engaging and entertaining, both of which, it is assumed, will increase the appeal for children.

Various interfaces have been developed for the purpose of selecting and organising resources, as well as for interacting with narrative (e.g. Mazalek et al., 2002; Jacob et al., 2002; Ullmer et al., 1998; Ryokai and Cassell, 1999; Ryokai et al., 2003). The playing cards in Card Shark and Thespis and in StorySpinner, described previously, provide the interface for navigating the hypertext story content in these models.

Gorbet et al. (1998) developed a tangible interface for interacting with narrative, which takes the form of triangular tiles that can be interlocked together in various configurations. Depending on the particular configuration of tiles, a story-segment is relayed either on screen or via an audio track. The authors point out that, due to the high number of possible configurations, authoring all the potential story segments is a very labour intensive task for even a small number of triangles. This suggests the need to develop the model and interface in such a way that it does not necessitate the use of more content than is available.

3. General Approach

The following sections of the paper outline key issues regarding the re-use of multimedia content and their technical implementation within our own model for creating an interactive game from existing material. These key elements are:

1) The subdivision of content into a collection of individual scenes.
2) The automatic generation of novel scene orderings according to some criteria and techniques for making transitions between scenes.
3) An interaction mechanism for a user to dynamically navigate through the content.

We then examine the benefits of applying narrative principles to the reorganisation of content and the extension of our model to incorporate this.

3.1 Sub-division of Content

Existing content must be sub-divided in order that it can be re-organised and re-used. Taking into consideration the work of Murtaugh, we adopt the approach of dividing content into self-contained scenes of approximately 30 seconds to two minutes duration. Each scene is meaningful in its own right such that it is possible to easily identify and describe one key event. An example of an event, taken from the Tiny Planets content, might be "Bing and Bong play on a seesaw". This event can be further broken down to identify and describe the individual elements that make up the event, such as characters (Bing and Bong) and props (the seesaw). We utilise such descriptions within the interaction mechanism and also for determining the contexts in which the scenes can be re-used.

Sub-division of the Tiny Planets content occurred in two stages. In the first stage, the content was viewed in its entirety in order to identify any common properties across the episodes. At this stage, for some episodes, the original storyboards, synopses and scripts were also studied. As well as assisting in the identification of common properties, a further purpose of this was to ascertain the extent to which it might be possible to incorporate some form of content description at this stage of production in order to facilitate later re-use. We found that much of the content description that would be required for our model was expressed, in some form, in these earlier production stages. However, information at these stages was not always carried through into the final product. Therefore, whilst it would in theory be possible to create much of the annotation at this stage, it would always be necessary to refer to the finished episode due to the frequency of changes made between the storyboard stage and the final visual edit. Further investigation would be required to determine whether or not this is commonly true of the production process.

This first stage analysis led to the identification of a plot-structure that was common to all episodes and also several themes that occurred across multiple episodes. The identified plot structure (which is examined in detail in *section 7.1*) is in line with conventional narrative theories and as such should be extensible to the majority of alternative story-based material that is consistent with this common plot model. Any alternative collection of content that does not adhere to this model but which is consistent, within itself, according to some other narrative model could also be used in place of the Tiny Planets content providing that this alternative narrative model is described and used instead.

In our case, where the annotations had not been made at the time that the storyboards were being drawn up, it was necessary to base the content sub-division and description on the visual content only. Therefore, in the second stage, the content was once again viewed in its entirety. This time, for each key event that was identified, the start and end points of that scene were noted. This scene was then further described according to its theme and plot-element type and also according to the characters and props present in the scene. The start and end points were annotated by referring to the episode number from which they were taken and the start frame number and end frame number of the scene. This was so as to provide an easy reference for referring back to a unit should the need arise to make additional annotations and also for playback of that unit during content re-use. Event types and scene description is described more specifically in sections 4.4 and 7.1.

3.2 Transitions between Scenes

We now have a library of scenes from which content can be selected and re-ordered. To consider how we may achieve this, it is useful to look in general at potential approaches to incorporating user-activity into content re-use.

[Figure 3]

Figure 3 shows two dimensions along which we can classify models for re-organising content. The first dimension refers to the content-type, which can be classified along a spectrum from completely canned to completely dynamically generated (as discussed in *section 2.2*). Our model falls within the canned category since we are specifically looking to re-use existing content. The second dimension refers to the whether the user actively interacts with the content during the viewing process or whether the content is reordered according to some criteria specified initially by the user, who then passively views the re-ordered content.

Momentarily putting aside any criteria used for choosing and re-ordering scenes in any given presentation, let's assume that we have a subset of scenes which we wish to present to the user in a novel ordering, i.e. in a context other than the one in which these scenes were originally authored to be shown. This means that certain visual and auditory cues that were originally intended to provide continuity between scenes by bridging across them (and which it is not always possible to eliminate during the editing of the scenes for re-use) will in many instances now cause an obvious discrepancy between scene transitions in the new ordering. Examples include:

- Background inconsistencies, especially when using scenes from multiple episodes.
- Soundtrack, voiceovers, ambient sounds etc. which are played "across" an edit between scenes for continuity reasons.
- Characters being shown to move in a certain direction at the start of a shot to imply a 'continued' movement from the previous scene.

Whilst film editors (whether professional or amateur) bring a knowledge of film grammar and editing principles to the piecing together of content in order to address continuity issues, such knowledge is not easy to formalise. This causes difficulties when automating the editing process. Nack (1996) and Davis (2003) both suggest editing strategies for the selection and ordering of content, based on certain visual and semantic properties. However, this approach will not always resolve such issues in cases where some other criteria for content selecting and ordering must take precedence over that of applying directorial principles to create a smooth transition between scenes (e.g. Bocconi, 2004). Another approach is to look at reuse within the *active* rather than the *passive* category. The incorporation of user-activity has the benefit that it builds different expectations in the mind of a user so that they compare the experience against that of playing a game rather than that of watching a film or television series. Therefore, the experience of interacting with the content is less like watching a second-rate equivalent of an 'episode' in which the selected content is joined seamlessly together. In this scheme, directorial principles may still be used where possible in the selection and ordering of scenes. However, continuity issues can be further addressed by the inclusion of "transition" scenes to reintroduce some of the editing principles that provide continuity between scenes. This additional material will be more effective in the context of game-play than it would be if they were simply inserted between scenes in a passive viewing experience.

3.3 Interacting with Content

In order to determine the way in which user actions can meaningfully affect the order in which scenes are shown during task performance, we need to identify certain properties of a scene that can be manipulated by the user. In order for the user-activity to be effectual, these must be properties that commonly differ across a scene-set. Examples of such properties, from visual content, include whether or not characters, or props, are present in a scene. Visual properties such as these are most appropriate for media of this type, although conceptual properties such as "types of event" depicted in the scene could also be used.

To give an example, imagine that we have three scenes, Scene 1, Scene 2 and Scene 3. There is a property, X, which can have one or more of the values {*c1, c2, c3, c4, c5*}. In the following example, the description "*has-property-x*" refers to which of these values are present in the scene. The scenes are described as follows:

Scene 1
has-property-x: c1, c2, c3

Scene 2
has-property-x: c1, c3, c4

Scene 3
has-property-x: c1, c5

If "characters" is the property that differs in terms of whether or not it is present in the scene and if *c1* is a character called "Bing" and *c2* is a character called "Bong", then the above example states that the character Bing appears in scenes 1, 2 and 3, whereas Bong appears only in scene 1.

These descriptions provide a basis for developing a consistent and intuitive interaction mechanism for making meaningful transitions between units of content. This interaction mechanism should ensure that in *response* to each user-action, or *command*, a scene is presented to the user that is consistent with the command given. In the simplest case a command might state that, in the next scene shown, "*has-properties-x*: c1, c2, c3" must be true. This is true of scene 1 and so the presentation of scene 1 to the user is an appropriate response to this command. A sequence of "user commands" and "system responses" results in a chain of scenes being created (see figure 4). A command may also encompass the current or previous state, as well as the next state. This will be discussed later.

[Figure 4]

We propose the following requirements in order to ensure that the experience of interacting with the content is directed and progressive.

Basic Rules for Interacting with Content:

R1: Every command leads to a response by the system. Therefore, commands can only be made available to a user if there is an appropriate scene to play.
R2: Each response occurs only once in any given presentation by the system, excepting repetition either at the users request or subsequent to user error (in which case repetition occurs immediately after the original presentation). This maximises progression through the scenes during interaction and maintains a novel user experience.
R3: At any given time, each available command must have unique screen representation and action. This increases user-agency by ensuring that the user can make informed choices at each interaction point, such that their interaction produces a meaningful response.

The purpose of these requirements is to ensure that an expected level of consistency with other standard interfaces is achieved.

There are further issues as to whether or not a user is presented with a complete set of possible commands (pertaining to a particular scene ordering) at the beginning of the interaction, or if context dependent commands are made available after the presentation of each scene. This choice will affect the complexity of the algorithms for generating legal sets of commands based on the above interaction rules. For example, if commands are presented to a user on a context dependent basis, then it is possible that commands can be available which are superficially identical, but which are in fact contextually different according to current state at the moment when the command is made available.

To illustrate the above, the following new scene is introduced:

Scene 4
has-property-x: c1, c3, c4

We wish to provide a user with a set of commands that will enable them to traverse through scenes 1 to 4, assuming that scene 1 is the first scene shown to the user. According to the interaction rules, the user may view each scene once only in any given presentation. Also, since there should be no duplicate commands, then once a command has been made it is no longer available to the user.

In the simple case, where a user-command simply dictates the values of the relevant property for the next scene shown, the set of commands that will enable the user to traverse the scenes are show in Table 1.

[Table 1]

We can envisage these commands being presented to a user as options in a drop-down menu. Looking at the above example, it becomes apparent why there is an issue as to whether a user is presented with all possible commands at the beginning of the interaction or whether the commands are context-dependent. Commands 1 and 3 are identical and therefore cannot be made available to a user at the same time. In the case where a user is to be presented with all commands at the start of an interaction, then one of these commands must be "lost" and the user will not be able to give the necessary commands to view all the scenes. Additionally, there is no possibility to apply constraints to the order in which a user is permitted to view the scenes.

To address these issues, one possibility would be to make commands available on a context-dependent basis. In this case, commands 1 and 3 are not presented to the user at the same moment and so are not considered to be duplicates. Additionally, it becomes possible to restrict the options that are made available to the user at any given moment, on the basis of the past history of interaction and on the values of desirable properties that the next scene should portray.

An alternative interface might make some of the interaction history explicit to the user by representing the context for using the command, thereby also representing the transition between scenes. The purpose of a transition is to "bridge the gap" between one scene and the next. Examples of some transitions between scenes 1 and 3 might be represented as in Figure 5.

[Figure 5]

Figure 5 implies that a transition has a left-hand-side and a right-hand-side. If these transitions are placed one after another to form a chain, then it becomes apparent that a transition essentially acts similarly to a domino-like tile. The left-hand-side is placed in reaction to an event that has just happened whilst the right-hand-side affects future events. By acting as a "bridge" between two scenes, the domino is a useful aid to providing continuity between them.

4. Playing a Domino Game with Tiny Planets Content

Standard dominos have dots representing the numbers one to six. Legal domino tile placement involves placing the left-hand side of a tile against the right-hand side of a previously played tile with a matching number, i.e. matching one to one, two to two etc. (see figure 6). The right-hand side of this tile then comes into play and is available to be matched by future tile placements. The game ends when one player has placed all their dominoes.

[Figure 6]

We have chosen to use the domino-like interface with the Tiny Planets content for several reasons. One requirement for the interface is that it should be usable by children in the target age-range of the Tiny Planets television series. Children of this age should have the necessary ability to match tiles. A further requirement is that the experience of interacting with content should be fun and engaging. Making the experience playful by structuring the task like a game of dominoes is one way of achieving this. Furthermore, as mentioned previously, a domino-tile acts to bridge the gap between scenes. This goes some way towards addressing some of the continuity issues that arise from re-using existing content.

The basic premise of our game is that instead of numbers, our domino-like tiles have pictures representing the properties which were identified as commonly differing across the whole scene-set. For example, a tile may depict characters or props. Tiles match primarily to scenes from the Tiny Planets series, rather than to other dominoes.

For each tile set that a user is presented with in a given game, there is a specific sub-set of scenes that have been selected according to some criteria, from which this tile-set has been generated with the express purpose of enabling the possibility that every scene might be viewed during game-play if tiles are placed in a certain order. The primary goal of this domino-game is to view the maximum amount of content available for a given game by placing every available domino-tile at some stage of the game. A secondary goal (in certain games where this is a possibility) is to create a complete domino-chain containing every available tile. However, these goals might be modified according to the criteria chosen for selecting and reordering content.

The domino represents the change, in the chosen property, that occurs between one scene and the next. The dividing line between the left and right side of the tile is effectively the point at which one scene ends and the next begins. Therefore, looking at a domino gives a partial view of the content of the two scenes and helps to bind the scenes together. However, the full "story" can only be seen by viewing tiles in the context of the scenes to which they match, since these contain additional information.

At the start of the game a player is presented with a number of tiles, which constitutes the entire tile-set for that particular game. A scene is then played and the player must select a tile that matches that scene and which specifies certain properties of the next scene to be played. The game progresses in this manner until all tiles have been played or until the player reaches a dead-end, i.e. there are no tiles left that can legally be played. In the latter scenario the user is permitted to "go back" to previous choice-points in order to pursue an alternative route through the game. By doing this the user is able to fully explore the content available for that game. To explain how such matching occurs, we explore three different game-types, based on different ways we have identified for matching tiles to scenes to create engaging game-play. For simplicity, from this point on we will use the example of "characters" taken from the Tiny Planets content for illustration purposes.

4.1 Complete Implicit

The first game-type involves a method of matching tiles to scenes which we term "complete implicit". The term 'complete' refers to the fact that each and every character that is depicted on a given tile must match with a character that appears in the scene to which the tile is matched. The term 'implicit' refers to the fact that the characters might not be explicitly present on screen at the time that matching occurs, but must have appeared at some point throughout the scene so are implicitly assumed to be present and available for matching.

In a complete implicit game, a tile matches a scene if the characters shown on the left-hand-side of a tile are the same as the characters that were present during the scene. The next scene played will at some point during the scene contain each of the characters that are shown on the right-hand-side of that tile. A legal tile-placement is therefore any tile in which the characters on the left-hand-side of the tile are exactly the same as the characters that appeared throughout the course of the scene playing (see figure 7).

[Figure 7]

4.2 Complete Explicit

We call the second type of matching "complete explicit". The term 'explicit' refers to the fact that characters must be explicitly present on screen in order that a tile might be matched to them. Thus, in this game, a tile matches only to the start and end states of scenes. So, a tile matches a scene if the characters on the left-hand-side of the tile were explicitly present at the *end* of the previously viewed scene. The next scene will *start* with the characters that are on the right-hand-side of the tile, but other characters may appear as the scene continues to play and it is possible for the scene to end with characters which are different to those that were present at the start of the scene. A legal tile-placement is therefore any tile in which the characters on the left-hand-side of the tile are exactly the same as the characters that appeared at the end of the previously played scene (see figure 8).

[Figure 8]

4.3 Rewrite Rule

The third type of matching involves ‘voting’ characters in and out of scenes. We call this type of matching “rewrite rule” since this game was partly motivated by previous work into the use of graphical rewrite rules for childrens’ programming of simulations, as found in environments such as Agentsheets (Repenning et al., 1998). When a scene has been played, the entire “cast of characters” for that scene are available to be voted out. The characters that are on the left-hand-side of a tile are the characters that are voted out and the ones on the right-hand-side of that tile are voted in. Characters that are on neither the left or right side of the tile remain. A legal tile-placement is therefore any tile in which the character(s) on the left-hand-side of the tile are a subset of the characters that were present in the scene that was just played (see figure 9).

[Figure 9]

The version that has been implemented is an “implicit” matching type, meaning that all the characters that have been present during a scene can be considered for voting out. It should be apparent that it is also possible to create a rewrite game based on explicit matching. In such a game, it would only be possible to vote out characters that were visible at the end of a scene. The right-hand-side of a tile would specify the characters that would be voted in, but only for the *start* of the next scene, after which any characters could appear or disappear. When the user is simply matching to start and end-states of scenes, such as in the complete-explicit game, then the unpredictably of characters appearing and disappearing can add to the engagement of the game. However, it was felt that in a rewrite game, the appearance and disappearance of characters, who had been neither voted in nor out, would not be in-keeping with the spirit of this game-type. Therefore, we did not fully explore the possibilities for rewrite-explicit games beyond the initial evaluation stage, when it was found to be a difficult game to play.

4.4 Game-play Legalities for all Three Game-types

For all three game-types, in terms of game-play mechanics the legality of a given tile from a player’s perspective is determined by whether the *left-hand-side* of the tile fulfils the matching criteria for that game-type. However, there are further legality issues that must also be considered with regards to selecting scenes that match to the right-hand-side of any selected tile. In the simplest case, in order to adhere to the rules of the game, it is necessary that a scene exists to be played that has the properties dictated by the chosen tile, according to the matching criteria. However, it is further desirable that this scene is novel (in that it hasn't been played previously in the course of a game) and also that this scene adheres to any constraints that have been specified with regard to the legal ordering of scenes. These issues must all be considered during tile-set construction.

Each game-type can also be played according to different criteria, for example:

- Theme – it is possible to specify a theme to be common to all scenes played in a given game.
- Tile-type - the attribute that is used to determine the properties of scenes played e.g. characters, props etc.
- The number of tiles in a game.
- Undo/Redo – whether or not the tile-set allows tile choices to be made at points in the game, which the user can then go back and change to see how an alternative tile placement would play out.

In order to implement such a game we need to describe the content according to certain criteria (table 2). For example, we need to know which characters appear in a scene and also whether they are present at the start and end of a scene. Such attributes are relevant to the mechanics of constructing legal games. Certain scenes may be classified as having an underlying *theme.* We later introduce further attributes by which scenes are described that are used to provide narrative coherence to the selection and ordering of scenes.

[Table 2]

5. Overall Architecture

This section details the technical implementation of the model. To recap, we have taken existing visual content and subdivided it into meaningful units. These are then classified according to thematic properties and described by attributes such as the characters present in the scene. We have an interaction mechanism, the "domino-tile", by which a user can participate in the construction of novel experiences by choosing properties of scenes to be shown and thereby affecting the order in which scenes are played. Figure 10 shows the basic architecture of the model to this point.

[Figure 10]

6. Algorithms for Generating Legal Tile-sets for each Game-type

To implement the domino-tile interface for interacting with content we need to develop algorithms for generating legal tile-sets for the three games. Firstly, we instantiate the basic rules for interacting with content, that were earlier identified in *section 3.3*, with a tile placement being the equivalent of a command and the presentation to the user of a scene being the equivalent of a response. Assuming at this stage we are not applying any additional criteria to the ordering of scenes, the instantiated rules are as follows:

R1[†]: Every tile placement leads to a scene being played.
R2[†]: Each scene is played only once in any game.
R3[†]: At any given time, each available tile must be unique.

Assuming that the chosen property is whether or not characters are present in a scene, we illustrate the algorithms with the following scenes, *S1, S2* and *S3* (*figure 11*). For clarity, we omit properties of the scene that are not relevant for generating tile-sets. We also use abbreviations in place of full character names (see key in *figure 11*).

[Figure 11]

6.1 Tile Generation Function

The tile generation function is used during tile-set construction to determine the left and right-hand sides of tiles, based on the chosen property and on the matching type.

6.1.1 Tile Generation Function for Complete Explicit

[Table 3]

From the scenes, S1, S2 and S3, we can generate a table of all potential complete-explicit domino-tiles for the scene-set (see Table 3). The top row across the table holds the start-state information for a scene and the left-side down the table holds the end-state information for a scene. To make a transition between scenes - for example, to go from S1 to S3 - we first read down the left-hand column to find the *end* state of S1 (bi, bo, t-l) and then across to find the *start* state of S3 (bo, b-l). This creates the domino-tile with the left-hand-side “bi, bo, t-l” and the right-hand-side “bo, b-l”, as shown in the table.

6.1.2 Tile Generation Function for Complete Implicit

The tile generation function for complete-implicit is essentially the same as for complete explicit. However, for complete-implicit matching rather than generating tiles based on the start and end states of scenes it is necessary to consider only one state for each scene, e.g. the set of characters that appear in that scene. This is reflected in Table 4. For example, the domino-tile that makes the transition S1-S2 has a left-hand-side “bi, bo, t-l” (the set of characters that appear in S1) and a right-hand-side “bi, t-l, r-l” (the set of characters that appear in S2). This is reversed for the transition S2-S1.

[Table 4]

6.2 Algorithm for Complete Explicit and Complete Implicit

The algorithm for generating tile-sets for complete-explicit and complete-implicit differs only in the tile generation function. It is implemented as a depth-first-search. Given the limited number of scenes that we are working with, we do not consider it necessary in this case to use refined search techniques for tile-set generation.

To generate all possible legal tile-sets (for example, for the scenes S1, S2 and S3), we must examine, for legality, each tile-set for every possible scene combination. A scene-combination can be of any length, with the maximum length for n number of scenes being *n-1*. Each tile-set is generated by applying the tile-generation function for each transition in a path and storing the tile which is applicable for that transition. This method of generating tile-sets ensures that rule R1† for interacting with content is adhered to, i.e. that for every tile placed there is a scene available to play. Rules R2† and R3†, for interacting with content, are adhered to by the following means:

According to R2†, a scene-combination must not include duplicate scenes. Therefore, during tile-set construction when a scene has been used it is eliminated as a candidate for future use.

According to R3†, the tile-set must not include duplicate tiles. Each tile-set is generated by applying the tile-generation function for each transition in a path and storing the tile which is applicable for that transition. By applying the tile generation function at each stage during path-construction it is possible to assess the current tile to see whether it is a duplicate tile that would therefore violate interaction rule R3†. If a duplicate tile is encountered during tile-set construction, then the path cannot be expanded further.

In some cases a tile-set may contain two or more tiles with the same left-hand-side and differing right-hand-sides. This is permitted because it does not violate any of the rules for interacting with content. The only consequence of this is that depending on the order in which these tiles are placed by the user, the user may either backtrack to a previous choice point in order to be able to place all tiles, or else end the game with tiles in hand that cannot be legally played (figure 12). These are both acceptable game scenarios.

[Figure 12]

6.3 Rewrite Rule

Algorithmically, the rewrite game is more complex than either complete explicit and complete implicit. It is not possible to generate the tile-sets for the rewrite rule game in the same simple fashion as for the other two game-types, i.e. by simply creating tiles to make transitions between all possible scene-combinations and checking the tile-set for duplication. This is due to the possibility of tiles being generated that can be placed at a point in the game, other than the position they are intended for, that would lead to an illegal state - i.e. a situation where no scene exists which has the particular properties dictated by the choice of tile. This is a potential risk in the rewrite game due certain properties being "carried over" from one scene to the next.

Using the example of "characters" to illustrate this point, within the game any tile placement is legal when the characters on the left-hand-side of a tile have appeared in the scene against which the tile is being placed. Characters which are not "matched" by that tile but which were in that scene *must* then appear in the next scene, according to the game rules. Looking at the characters in scenes S1 and S2:

S1
has-characters bi, bo, t-l

S2
has-characters bi, t-l, r-l

The tile that makes the transition between these two scenes, according to the rewrite game rules, is "bo | r-l". Let's call this tile *t1*. This tile dictates that the subsequent scene has:

'*characters(s1) - lhs(t1) + rhs(t1)'*
= "bi, bo, t-l" - "bo" + "r-l"'
= "bi, t-l, r-l"
= *characters(s2)*

Of course, these are the characters which appear in S2 because the tile was explicitly made for the purpose of making this transition. However if, as happens to be the case in this example, the character on the left-hand-side of the tile is common throughout the scene-set, then this tile is also going to be matchable against a number of scenes. There is also the increased likelihood of other tiles being generated for a particular scene-combination that has only this character on the left-hand-side. This domino will not necessarily be eliminated as being a duplicate, since the probability is that the right-hand-side will differ. For example, we introduce the following scenes:

S4
Has-characters bo, r-l

S5
Has-characters b-l, r-l

Imagine we are generating a tile-set for the scene-combination S1-S2-S4-S5. We already have the tile "bo | rl" to make the transition S1-S2, now we also generate "bi, t-l | bo" for S2-S4 and "bo | b-l" for S4-S5. To recap, our tile set now looks as follows:

t1 = bo | r-l
t2 = bi, t-l | bo
t3 = bo | b-l

Within the game, the user is presented with this entire tile-set and given freedom as to which of any legally playable tiles they want to place at any point in the game. When S1 is played, the user can choose between tiles *t1* and *t3*, which are both legal choices. If *t1* is played, then the game can proceed as intended. However, if *t3* is played, then this dictates that the next scene to be played must contain:

'*characters(s1) - lhs(t3) + rhs(t3)'*
= "bi, bo, t-l" - "bo" + "b-l"'
= "bi, t-l, b-l"
= *characters(?)*

Since the characters “bi, t-l, b-l” do not appear within any scene in our scene-set, then it is not possible to find a scene matching the users choice.

In order to prevent the generation of tile-sets which enable these illegal situations to occur, the rewrite algorithm examines every possible scene-transition and compiles a list of dead-end rules for each scene based on the properties that are being used to compile the tile-set, e.g. characters in the scene. A dead-end-rule renders certain transitions illegal within a given path containing the scene to which the dead-end-rule applies. To illustrate how dead-end-rules are identified and applied, we use the following scenes:

S1
has-characters bi, rsf, gsf, bsf

S2
has-characters bi, bo, b-l

S3
has-characters bi, bo, rsf, gsf, bsf

S4
has-characters bi, bo, r-l

[Table 5]

Table 5 represents the entire tile-set for the set of scenes from S1 to S4. To create the tile that makes the transition Si – Sj, from the above table, the left-hand-side = Si-Sj and the right-hand-side = Sj-Si. In other words, to make the transition S2-S4, the left-hand-side of the tile is “b-l”, which is in the cell “2-4” and the right-side is “r-l”, which is in the cell “4-2”. We reverse this tile to make the transition S4-S2. The following creates the table values:

Si-Sj = characters in Si that are not in Sj
Sj-Si = characters in Sj that are not in Si

Starting with S1, which has the characters “bi, rsf, gsf”, we compile the dead-end-rule list for S1 by looking at each transition from all scenes *except* S1, e.g. S_2-S_1…. S_2-S_4, S_3-S_1….. S_3-S_4 etc. If the set of characters contained in any of these transition cells are *all* included as members of the scene S1, then that transition is not viable for paths containing S1. That transition is then included in the dead-end-rule list for S1. By applying these principles we create the following dead-end-rule lists for S1 – S4.

S1	S2-S2, S3-S2, S3-S3, S3-S4, S4-S4
S2	S1-S1, S1-S3, S3-S1, S3-S3, S4-S4
S3	S1-S1, S1-S2, S1-S3, S1-S4, S2-S2, S4-S4
S4	S1-S1, S1-S3, S2-S2, S3-S1, S3-S3

As a path is constructed, the dead-end-rule list for a path is updated by adding into it the dead-end-rule for each scene added to the path. A scene can only be added to the path if that transition is not ruled illegal by appearing in the dead-end-rule list for the path. This ensures that a legal tile-set is constructed.

7. Narrative Principles

All these algorithms can be run on the complete set of available scenes or else any scene-subset by a given criteria, such as “theme”. In general, these algorithms are easily extensible to include further criteria. For example, rather than simply recombining content in an arbitrary fashion it is possible to impose structure to the scene ordering according to certain attributes.

For example, consider the following levels at which content can be selected and re-organised:

Level 1 – content is selected and organised randomly
Level 2 – content is selected and organised according to thematic similarities. Examples include compilation episodes of a sketch show, “caught on camera” type television shows and blooper shows. There may be some underlying theme and episodes of local coherence across several scenes, but no real narrative that carries across the episode.

Level 3 – content is selected according to thematic similarities and described and organised according to narrative principles.

We are most interested in level 3, the narrative ordering of thematically related story units. Story-based content, such as we are using, lends itself particularly well to the application of narrative principles. There are several further reasons to support this choice.

Research suggests that narrative assists learning (Decortis et al., 2003). It provides coherence and by doing so, it makes the experience more understandable and therefore aids memory. We initially look at the possibilities for re-using content that is specifically aimed at children. Based on the premise that children's broadcasting can create a beneficial learning environment, this type of content commonly has some educational value incorporated into it which fits within the context of the story. In such scenarios, learning mainly occurs as a by-product of the viewing experience, with the primary focus being on the entertainment value. However, the benefit of interweaving narrative with educational content is that it makes the learning experience more entertaining and enjoyable than traditional fact-based teaching methods and this can help to maintain the child's attention. It would therefore be desirable to try to retain this association between narrative and learning to the largest possible extent within the model, as well as the educational aspects of the original content itself.

Therefore, we wish to use narrative principles to guide the selection, description and organisation of content, since we hypothesise that this will make the experience more engaging and memorable than the other levels of content presentation. On a more general level, the application of narrative principles to provide coherence and aid understanding is a benefit to a user of any age.

7.1 Plot Analysis

In order to understand how narrative principles can be applied, we have undertaken an analysis of the 65 Tiny Planets episodes, taking into consideration conventional narrative theories. Brooks (1996) identifies an abstract narrative structure based on earlier work by Branigan (1992). The abstract structure is comprised of the following narrative primitives:

1) speaker introduction – introduction of the character from whose viewpoint the story will be told
2) character introduction – introduction of other characters involved in the story
3) conflict – introduction of an obstacle to be overcome
4) resolution – overcoming the obstacle
5) diversion – a moment of tension relief, possibly comedic, which is incidental to the plot
6) ending – overall resolution to the narrative

As mentioned in *section 2.1*, the ordering of these primitives can be manipulated to produce narrative effect, so that the narrative ordering of events might differ from the underlying story. Additionally, there might be either a single conflict or else a series of conflict/resolution pairs within the conflict phase. Conflicts may have multiple resolutions and a resolution may have multiple conflicts. The choice of which to use affects the rhythm of the narrative, for example introducing many conflicts before any are resolved has the effect of building narrative tension.

Our own analysis of the Tiny Planet content revealed what we term our "minimum narrative plan" (figure 13). This plan is based upon a plot-structure that we found to be applicable to all episodes of Tiny Planets.

[Figure 13]

7.1.1 Theme Introduction

Each Tiny Planets episode has a theme, which tells the viewer the general purpose of the story. This theme is stated at the start of each episode by way of a voice-over. An example of such a theme might be "a flocker is having trouble pushing a giant ball up a steep ramp on the Tiny Planet of Stuff. Bing and Bong figure out an easier way to move it". This example is taken from a Tiny Planets episode entitled "Ramping up". Thereafter, though not necessarily at the very start of the episode, the following constituents will be introduced into the story: a flocker, a giant ball, a ramp, Bing and Bong.

7.1.2 Conflict and Resolution Attempts

Within each episode there is at least one conflict which must be resolved, through a successful resolution attempt, before the story can end. Analysis of the episodes revealed several possible conflict patterns. There may be one main conflict, for which many resolution attempts are made (ending in failure or partial success), before the conflict is successfully resolved. Or alternatively, there may be a series of conflicts, such that having resolved one conflict successfully, a further conflict is introduced.

In the case of "Ramping up", the main conflict is introduced when Bing and Bong have helped a flocker to push a ball up to the base of a large ramp and then try, but fail, to roll the ball up the steep slope onto the top of the ramp. Several further resolution attempts occur which end in failure, such as trying to use a ladder to get the ball on top of the ramp, before finally the conflict is successfully resolved when Bing and Bong roll the ball up the gradual slope of the ramp.

7.1.3 Postcompletion Events

The story never ends immediately after a conflict has been resolved. Once all conflicts introduced within a story have been successfully resolved, there are one or more postcompletion events. In the "Ramping up" episode, the postcompletion event takes the form of Bing pushing the large ball down the steep slope of the ramp and into the midst of some giant skittles, scoring a perfect "strike", i.e. where all the skittles fall down. Other common elements of each episode include the opening and closing credits, and arrival and departure on one of the Tiny Planets.

7.1.4 Sub-conflicts and Comedic Events

A further plot-element identified at this level is a sub-conflict. A sub-conflict is a conflict which occurs and is resolved but which does not progress the story in terms of the theme. Sub-conflicts often have comedic value or provide dramatic effect. In "Ramping up", the character Bong is trying to help the flocker by rolling the ball towards the base of the large ramp. However, Bong suddenly reverses direction and goes trundling past a bemused flocker and Bing, going the wrong way. Bing and the flocker indicate to Bong that she should roll the ball in the opposite direction towards the ramp, thereby resolving that conflict.

7.1.5 Sub-plots

In addition to each episode having a main plot, there may also be a sub-plot. A sub-plot is a self-contained plot occurring within the context of the main plot. For example, in "Ramping up" while the character Bing is trying to find ways to get the large ball up onto the top of the ramp via the steep slope, the other main character Bong realises that the solution lies in rolling the ball up the gradual slope on the other side. The conflict that Bong faces is that she cannot get Bing to listen to her idea.

7.1.6 Themes

Looking at the series as a whole, rather than analysing each individual episode, we have identified a number of themes which reoccur across multiple episodes. These themes reflect an educational aspect to each episode from the series. A number of episodes involve "shapes", for example showing how triangular shapes cannot get through circular holes. The shapes involved may be either 2D or 3D, but the principle is the same. Another common theme is "heavy objects", where a character must find a method for moving, for example, a large ball. Solving such problems often involves the use of "levers", which is a type of "sub-theme" for heavy-objects.

7.2 Description of Content According to Plot Level for Narrative Re-ordering of Scenes

The narrative plan specifies the most basic plot structure to which each newly created narrative must adhere in order for it to achieve narrative coherence. Thus, constraints are included in the model as to the permitted ordering of scenes according to the narrative plan. In order to implement this, it is necessary to describe each scene according to certain attributes.

Each scene may firstly be described according to the "theme" to which it relates, such as "shapes" or "heavy-objects". By including a theme attribute in the content description we can select sub-sets of scenes according to a chosen theme and enhance the narrative by ensuring that all scenes shown within that narrative have a thematic similarity. This has the additional benefit of enabling the educational value from the original series to be retained. Some scenes are generic in terms of the event they convey so not all scenes have a theme-description.

As discussed previously, our content has been sub-divided so that each scene describes one primary event. Each event correlates to the individual plot elements that comprise the narrative plan. Therefore, we can now classify each individual scene according to its theme and to its *plot-element-type*, as described below:

- *Theme-introduction* - A scene is described as a theme-introduction if it introduces concepts that pertain to one of the themes identified as reoccurring across multiple episodes, but is not itself a conflict-introduction, resolution-attempt or conflict-resolution.
- *Conflict-introduction*, *conflict-resolution* and *resolution-attempts* - A scene can be described as a conflict-introduction if it clearly introduces a conflict for which there is a matching conflict-resolution, i.e. the original resolution from that same episode. In effect, conflict introduction and resolution attempts are essentially the same. For our purposes, the conflict introduction is a term applied to the first time a conflict is introduced in any particular narrative-ordering of units. Subsequent scenes within this narrative ordering may be failed resolution attempts from the same episode and for the same conflict or else failed resolution attempts for the same theme but from different episodes. An example of this would be to show a triangle-local failing to get through a round hole as a conflict introduction. From the same episode, a subsequent resolution attempt might show a round-local failing to get through a triangular hole. From a different episode it would be possible to show a box local getting stuck in a hoop (which occurs in at least two different episodes). The ultimate resolution in this case would be to see the triangle local getting through the triangular hole. Therefore, a scene may be described as both a *conflict-introduction* and a *resolution-attempt.* However, some resolution attempts may not be conflict-introductions if there is no clear scene showing the conflict being resolved.
- *Post-Completion Events* – A post-completion event may either be the original post-completion event for a conflict-introduction/conflict-resolution pair or else some generic scene in which characters are seen to 'say goodbye' or 'leave'.

The minimum narrative plan can be enhanced by applying further narrative and directorial techniques in both the selection and ordering of content. One example of a directorial technique is the use of comedic moments to relieve dramatic tension, thus affecting the pacing of the plot. These are equivalent to the *diversions* identified by Brooks. These can easily be incorporated into the model by identifying scenes that have a comedic value and including knowledge about comedic timing and placement of such scenes within the model. We add this plot-level attribute into our content description to enable the content to be described and re-organised according to narrative principles (see figure 14). The length of time (i.e. the number of units) between introduction of a conflict and its ultimate resolution can be varied to provide more or less tension, corresponding to more or less failed resolution attempts being shown. The minimal narrative plan used in the model assumes that story time and narrative time do not deviate. This decision was taken because of the age of the children using the demonstrators. The Tiny Planets episodes themselves also conform to this. An alternative narrative plan could be used in which the two times differ. For example, the resolution of a conflict could be shown near the beginning, followed by the stages that had led to this event happening. For similar reasons, more sophisticated plot twists involving multiple conflicts that could be used to create suspense are also not implemented in this model but could be easily incorporated into alternative versions. Overall, the narrative plan, the description of content according to plot-element-type and the principles of theme-description can be easily tailored to apply to alternative story-based content.

[Figure 14]

7.3 Updated Algorithm

We can now update our algorithm using the narrative principles to provide constraints to the ordering of scenes to ensure plot coherence. The simplest plot structure must have the elements "theme introduction" (TI) "conflict introduction" (CI), "conflict resolution" (CR) and "post-completion event" (PC) in this order, to maintain plot coherence (see figure 15).

More complex plot structures must still maintain this ordering, but could have additional plot-elements such as comedic-events. Additionally, themes may be introduced over the course of several scenes, this set of scenes comprising a *theme* phase (TP).

[Figure 15]

A further requirement in providing plot consistency is that the conflict-introduction and conflict-resolution in any potential plot-structure must be a pair, taken from the same episode, such that the first conflict introduction that is shown in a game has a corresponding resolution for which the tile-set provides the possibility of placing tiles such that this resolution is shown. A *conflict phase* (CP) may include a number of resolution attempts in between the conflict introduction and its ultimate resolution.

Figure 15 shows a representation of the scene-subset during tile-set construction. The circles represent the sub-set of clips which match the chosen theme. These clips are then classified according to their plot type and four clips are then chosen as "anchors" (highlighted in grey in the diagram). The anchors are TI, CI, CR and PC. All other clips that could be a theme-introduction are classified as belonging to the theme introduction phase (TP) and like-wise, those that are classified as being conflict-introductions/resolution-attempts belong to the conflict introduction phase (CP). These are the white circles in the TP and CP boxes.

Comedic clips do not fall into either of these categories but may match the chosen theme. These scenes are therefore also available for use within the tile-generation phase. These are shown as white circles in the "comedic events" box.

In the diagram, the arrows depict transitions between scenes. A transition is essentially a domino-like tile that has a left-hand side which matches the scene the arrow comes from and a right-hand-side that matches the scene the arrow points to.

Ordering rules are used during tile-generation to ensure that illegal moves are not possible for a constructed tile set. TI is the starting scene of the game, i.e. the one that is chosen to be played before the first tile must be placed. From TI, it is possible to go directly to CI, or else to visit several scenes within TP or to have a comedic event. Within a legal tile-set there must always be a tile which enables the conflict introduction to be shown. Once in the conflict phase, it is possible either to move straight to the resolution, which leads directly to a postcompletion event (the game is over and there should be no tiles available to go "back" in the game) or else to show more scenes from within the conflict phase, or else comedic events. However, it is not possible to re-enter the theme phase from the conflict phase. A further constraint is that scenes cannot be shown twice.

We can now add the following requirements for generating legal-tile sets, such that within a legal tile-set there must be no tiles available that allow the following illegal situations to occur:

- The first scene shown within the conflict phase must be the conflict introduction. Therefore, it must not be possible for a user to place a tile against any scene within the theme-phase that forces any scene from the conflict-phase, other than the conflict-introduction, to be shown.
- Narrative constraints dictate that conflicts must be introduced before they can be resolved. Therefore, an illegal situation might occur if a user was able to place a tile that necessitated that the conflict-resolution be shown prior to the conflict introduction being played
- The conflict-resolution signifies the end of the conflict phase. At this point, the only possible move is to a post-completion event. There must be no legal tiles available to play once this has been shown.

7.3.1 Technical Implementation

With regards to the technical implementation of the above, the application of narrative constraints requires that certain transitions are rendered illegal for any given path-set. For a given set of scenes we can generate a list of illegal transitions so that they are not used during path construction. The rules for identifying illegal transitions are:

- If the state is a theme-introduction, eliminate all transitions from that state to any state in the conflict-phase other than any conflict-introductions that have matching conflict resolutions.
- If the state is a conflict-introduction, resolution-attempt, conflict-resolution or post-completion event, then eliminate all transitions from that state to any state in the theme-phase

The algorithm must now occur in three phases to ensure that plot-ordering is maintained. The first phase produces the set of legal-paths from theme-introductions to conflict introductions. Conflict introductions are "goal" nodes in this phase. The second phase adds to each of these paths, in turn, the legal paths from the conflict introduction of that path to the conflict resolution of that path. Conflict resolutions are "goal" nodes in this phase. The third phase produces transitions from a conflict-resolution to a post-completion event for each path. At any stage, a potential path is discarded if it is found to contain an illegal tile for that particular path. The output of this algorithm is the set of legal transition paths for the scene-set. This update to the algorithm is applicable for all three game-types.

Figure 16 shows the updated architecture with the narrative engine included.

[Figure 16]

8. Implementation – Tiny Planets Demonstrator

The Tiny Planets demonstrator is an implementation of our model, Scene-Driver, for re-organising content. It incorporates narrative constraints, the domino-tile interface and uses content from the Tiny Planets television series. In accordance with the use of narrative principles for reorganising content, the primary goal for the user is, in this case, to view a completed narrative (rather than in the generic version where the primary goal was to view the maximum amount of available content). Additional user-defined goals might be to achieve the objective of viewing an entire narrative by placing the minimum or the maximum number of tiles (e.g. either seeing only the essential narrative elements or else seeing every single available scene, including all non-plot essential scenes, such as comedic moments). The demonstrator has been developed in order to test the model both from a technical perspective and in terms of usability by children of the appropriate age for the Tiny Planets content, at whom this implementation of our model has been targeted.

There are two basic versions of the demonstrator, one specifically developed to be used within evaluation studies and one developed to implement algorithms for generating tile-sets and re-organising content according to narrative constraints and selected theme. In the version which was used in evaluation studies, the scene-ordering is pre-calculated and the graphics for the domino-tiles are all pre-generated and of high standard. Pre-calculation of scene-ordering enables control for factors such as game length when running the experiment and it was thought that the high graphical quality would enhance the appeal for children taking part in the studies. The second version of the demonstrator is fully dynamic in terms of choosing novel orderings for scenes and for this reason the domino-tiles are created dynamically so that it is not necessary to pre-generate every single possible domino-tile. The benefit of this dynamic version of the demonstrator is that it can easily be adapted for re-use of different content types simply by utilising a different scene-library and scene annotations, as well as individual graphics for characters and props (for generating the domino-tiles). However, some graphical quality is sacrificed in order to achieve this.

We have additionally incorporated some directorial elements in order to improve continuity between scenes. These directorial elements take the form of "transition scenes". These are scenes which have been specially constructed to be played between one scene and another. They show characters leaving and arriving between scenes. Within the version used in the evaluation studies, these scenes were all pre-authored shockwave movies. In principle, it should be possible to write a program with either shockwave or another programming language to generate the appropriate character movements at run-time in order that transition scenes could also be included in the dynamic version of the demonstrator.

This section provides a series of screen-shots, taken from the version used in the evaluation studies, to illustrate some features of the Tiny Planets demonstrator.

When starting the game, a scene is played on a "television" in the centre of the screen. In figure 17 this scene involves Bing and 3 flockers trying to push a ball up a ramp. In the top-left of the screen is an "inter" tile. An "inter" is a tile played by the computer to reflect what has happened in a scene. They are used in both the complete explicit and rewrite games for continuity reasons and also to aid the child in matching tiles to scenes. In a complete-explicit game the "inter" takes the form of a different coloured tile, that matches the start and end state of the scene which has been played. In a rewrite game, the inter takes the form of a "cast-tile", which shows the cast that had been present within the previous scene and which characters are therefore available to be "replaced" in the next scene. In figure 17 the "inter" shows that the scene both started and ended with Bing and 3 flockers. At the bottom of the screen is a set of 8 available tiles.

[Figure 17]

In this scenario, a legal tile has been chosen with Bing and 3 flockers on the left-hand-side and 6 locals on the right-hand-side. Therefore, Bing and the 3 flockers must leave and the 6 locals must arrive (the transition is shown in figure 18). The next scene will start with the 6 locals. If an incorrect tile had been chosen then a cross would appear to indicate the wrong choice and the scene would have been replayed.

[Figure 18]

Once the transition has finished playing, the next scene that has been selected by the choice of tile is played and the "inter" appears which matches the start and end-state of the clip that is being played (Figure 19).

[Figure 19]

A screen shot of a completed game (one in which all tiles have been placed) can be seen in figure 20. The row of tiles represents the collaborative narrative created by the child and Scene-Driver. Within some narratives, there may be points at which there are two or more tiles with identical left-hand-sides, but differing right-hand-sides, are available to be placed. The user's tile-choice at this point will affect the course of the narrative. However, once the choice has been made the user is permitted to "go back" to the choice-point, to see how an alternative tile-choice would have played out. In such scenarios, a "back-button" appears to indicate the possibility to return to a previous choice-point.

[Figure 20]

The dynamic version of the demonstrator, which implements the algorithms for generating tile-sets is run on a PC and is written in the LISP programming language, utilising functionality that enables the generation of web-pages (the demonstrator is played via an Internet Explorer browser window). The version used in evaluation studies is a scaled-down version of this, which uses pregenerated graphics and a predetermined path as well as including the extra functionality for recording the data from the studies.

Additionally, the version that was developed for use in the evaluation studies has also been demonstrated in Shockwave ™ and as an interactive DVD in order to demonstrate that, in principle, different platforms might be used to deliver the game. The Shockwave version is an 8 tile version of the complete-explicit game, using the same scene-library, transition scenes and graphics as the version that was used in the evaluation studies, but with the functionality written in Lingo. The DVD version was authored on a PC and also uses the same scene-library, transition scenes and graphics as the other version. However, in the DVD version each screen that a child views is essentially a menu and each domino acts like a DVD menu option. When a tile is selected, this takes the child to the next "menu" where they can see the scene that matches their tile-choice and choose their next tile. Since DVD players lack the functionality required to dynamically generate menus, it is necessary to explicitly author every possible "menu" response to a child's tile choice at each stage of the game. Therefore, similarly to the version used in the evaluation studies, the narrative path of a given game is predetermined.

The DVD version has been implemented as a simple 3-domino demonstrator using a child-friendly DVD remote-controller developed by Berchet ™ (see figure 21). This device consists of a sheet which overlays 9 touch-pads which are the equivalent of activating the digits 0-9 on a regular DVD remote control. Careful placement of graphics on the overlay sheet ensures that when each graphical element is pressed the touch-pad beneath is activated. The action for pressing a particular touch-pad is pre-authored at every stage of the game to either take the user to the next "menu" where the applicable scene is played or else to a screen which indicates that an invalid tile has been chosen. It is possible to have multiple different overlay sheets for any given DVD, each with a different set of tiles. Placing a sheet on the device causes the DVD to skip to a new chapter with a new game on it.

[Figure 21]

9. Evaluation Studies

Scene-Driver has been evaluated throughout the course of its development in four separate studies, with the results of each evaluation being used as part of the development process. The purpose of each study was as follows:

1. Study one used mock-up cardboard dominoes to investigate the ability of children to use the domino-tile interface for each of the three game-types.
2. Study two was essentially a re-run of the first study but using the Tiny Planets demonstrator to enable the children's tile-choices and errors to be recorded. An engagement metric was also introduced in this study that was carried forward and used in the subsequent studies.
3. Study three was similar to the second study, but with a larger sample-size and evaluating children who were a year older than those who took part in the previous study. The primary purpose of this study was to assess the performance of these older children in playing the rewrite game compared to the complete-explicit game.
4. The purpose of study four was to compare the playing of a narratively coherent game with a game in which the clips were selected and played in an arbitrary way.

Due to specific issues pertaining to the design and running of experiments with young children, all four studies were designed under the guidance of and conducted by a child psychologist. Additional benefits of this included the possibility of the child psychologist to contribute some informal observations on each study in addition to the collected data and to identify possible additions or improvements to the experimental design, the design of the demonstrator used in the studies or the game itself, for use in future studies.

We present an analysis of the methods and results for each of these studies below.

9.1 Study 1

The aim of this study was to gain some general feedback of the child's understanding and enjoyment of the experience and to assess their ability to play the three different types of game. Children were tested in three conditions: complete-explicit, complete-implicit and rewrite-rule.

For each condition, the materials consisted of a set of 8 "mock-up" cardboard domino tiles, a set of 9 scenes and a further "scene-domino-scene" example. The scenes used were edited from Quicktime movie files. In addition, the complete-explicit and rewrite-implicit conditions both had a set of cardboard "inters" constructed which were to be placed after each domino that a child had placed.

For practical considerations relating to the conducting of the experiment, each set of dominoes and scenes were, respectively, constructed and selected such that in whichever order the dominoes were played (given that, at times, there were choices) there was a specific scene to be played for that domino choice. This ensured that there would always be a scene available to play if a child had placed a correct domino and that locating of the scene would be an easy task for the experimenter

The study was conducted with 20 children aged between 6 and 7. A laptop was used to display the scenes to the child and children were video-taped as they took part in the experiment.

Each child was tested individually. They were first shown the scene-domino-scene example and had the matching procedure explained. They were then presented with the 8 dominoes which were available for placement and the first scene was played. In cases where a child had progressed through the game to a point where they could match no more dominoes, yet still had tiles remaining due to earlier alternative choices having been available, the child was taken back to the “choice-point” and allowed to make the alternative domino selection.

9.1.1 Results of Study 1

The first evaluation was a pilot study aimed to highlight any issues that should be considered during the development of the Tiny Planets demonstrator and to inform the design of the next study, using this demonstrator. Therefore, the results of study one were based on informal observation of the children playing the game rather than on empirical data analysis.

The overall impression was that the children were engaged in playing the game and found the experience to be enjoyable. Children who were already familiar with the Tiny Planets television series tended to understand the principles of the games slightly quicker than those for whom the characters and the television show were novel. There were no apparent difficulties in understanding the game mechanics in terms of being able to “go back” and make alternative tile choices at some points in the game. In this first study, it was not possible to fully evaluate the extent to which children understood the relationship between the tiles on the table and the clips on the screen. Inevitably, there was a lack of spontaneity between the child playing a tile and watching the next scene, since it was necessary for the experimenter to intervene to play this scene. It was decided that better data would be obtained in further experiments using a fully automated system that allowed the child to have full control and for the experimenter to take a back-seat.

Children showed a good overall ability to match tiles to scenes in all three game-types, with complete-explicit appearing to be the easiest, rewrite-rule the hardest and complete-implicit somewhere in between.

9.2 Study 2

Development of the Tiny Planets demonstrator meant that in the second study children could directly control the game without the need for intervention from an experimenter to play scenes. A further benefit of automating the game was the ability to record data, such as number of errors and the choices made when there was more than one possible match. The main purpose of the second evaluation study was to test the technical implementation of the demonstrator, to evaluate rating scales and metrics for measuring enjoyment and to assess children’s ability to use the tool to play the game.

The study was conducted with 9 children (3 female and 6 male) aged between 6 and 7. This study had the same three conditions as the first: complete-explicit, complete-implicit and rewrite rule and 3 children took part in each condition. The tile-sets and clips were also the same. As in the first study, a “scene-domino-scene” example was used in each condition for the purpose of explaining how each game worked.

Two metrics were used for assessing the child’s enjoyment of the task. A widely used enjoyment measure is the smiley-face rating scale. This scale was developed for childhood pain assessment and was initially intended for use in hospitals. It consists of a number of cartoon faces ranging from very sad to very happy (see figure 22).

[Figure 22]

This scale has been used to measure children’s enjoyment on a variety of tasks. However, Airey et al. (2002) suggest the possibility that children might instinctively show a preference for smiley faces and may further have difficulty associating a sad face with their own opinion of a task. For these reasons, this scale may not always give an accurate assessment of enjoyment. They go on to suggest a different enjoyment metric, which they call a “sticky-ladder” rating scale (figure 23). The sticky ladder is made of a material to which discs backed with Velcro may be attached. Rather than pointing to pictures on a scale, children can express their views by physically sticking items onto the ladder to rate them in terms of preference. Both the general and specific meaning of the discs can change depending on the nature of the task. Also the child is asked to, in effect, calibrate their own scale. For example, if the sticky-ladder us being used to assess enjoyment of a game, then each disc represents a game. The child is asked to specify the meaning of the disc that represents their most favourite game and their least favourite game, these being placed at the top and bottom of the ladder, respectively. This ensures that the child understands what each disc represents. When the child is asked to make a comparison between these discs and the game that is being assessed, they are comparing "like with like" and are able to make a self-assessment of their enjoyment relative to two other games that they are familiar with. The Sticky Ladder rating is therefore relative to other games that the child has played, rather than being an absolute measure of enjoyment.

[Figure 23]

For the study, a smiley-face scale was constructed and printed out onto sheets of paper and a "sticky ladder" was made, with one of the discs constructed to represent the Tiny-Planets game. The purpose of using both metrics was to enable comparison between the two types of scale for use in future studies.

The tool was installed on a laptop which had a mouse attached and each child was videotaped as they played the game. Again, each child was tested individually. They were first taken through the example for the condition they were being tested in. When the child had adequately demonstrated that they understood how the game worked and how their tile choices would affect the scene that was played, the experimenter proceeded to a "login" screen where the name, gender and age of the child were recorded. After this, the child was allowed control of the computer and the experimenter was on hand to guide the child if they were in obvious difficulty.

When all tiles had been placed, the experimenter asked if the child had enjoyed the game. The sticky ladder was then produced. A blue disc was handed to the child and they were asked to think of their least favourite game that they have played. They were then asked to imagine that the blue disc was their least favourite game and to stick it onto the bottom of the ladder. This procedure was repeated with a red disc, representing their most favourite game, and they were invited to stick the red disc at the top of the ladder. Children were able to interpret game to mean any kind of game that they played, rather than just being other games played on the computer (e.g., playground games such as hopscotch, board games etc. were permitted). When both discs had been placed they were given the Tiny Planets disc. The experimenter then pointed at the blue disc and the red disc in turn and asked "if this is your least favourite game and this is your most favourite game" (with "this" being replaced by the name of the games the child had stated) "where on the ladder would you put the tiny-planets game that you've just been playing?" Their choices were recorded. The children were then shown the smiley-face scale and asked to describe the faces. Once the child had described the sad face and the happy face, they were asked "thinking about the game you have just played, which face would you choose to describe the experience of playing the game?" Again their choices were recorded.

It was considered that only a small sample size was necessary to assess the efficacy of the two rating scales and to see if any changes needed to be made to the tool or to the experimental design prior to running a study on a larger sample size.

9.2.1 Results of Study 2

Similar to the previous study, the majority of the results are based on observation due to the purpose of the study and the small sample size. With regards to the objectives of testing technical aspects of the Tiny-in demonstrator and assessing the children's ability to use the tool, there were no apparent difficulties.

In terms of understanding the process of tile-matching, the results reflected those seen in the first study. It appeared that comprehension was good in the complete-explicit and complete-implicit games, but that children in the rewrite condition required more guidance from the experimenter than those in the other two conditions. Figure 24 shows the percentage of correctly placed tiles in the three conditions.

[Figure 24]

It seemed that the ability to correctly match tiles was reflected in the enjoyment rating subsequently given on the sticky ladder (Figure 25). This suggests that enjoyment may in part be dependent on whether or not the child was able to immediately understand the game rules or if they required some assistance from the experimenter. With regards to the smiley-face rating scale, *all* children chose the happiest face, which suggests that this rating scale is less effective than the sticky ladder at rating a child's preference within this age-group.

[Figure 25]

Overall, these results suggested that whilst the two complete-matching games are suitable for children of this age-group, the rewrite game may be more suited to older children.

One intended benefit of using the Scene-Driver tool was that children could directly control the game without the need for intervention. It was assumed that this would assist the children in understanding the relationship between their tile choice and the scene that was shown. Some observational data obtained from this study was that some children, when prompted to explain a tile choice (either correct or incorrect), would point to a character (or characters) on a domino and say that they wanted to see that character. This suggested that these children fully understood how they could manipulate the content of scenes by their tile-selection.

9.3 Study 3

The purpose of the third evaluation study was to essentially re-run the previous study with children who were one year older than those tested in the second evaluation study, to investigate whether older children were more competent at playing the rewrite-rule game. Since the results of the previous study suggested that the complete-explicit game was the easiest to play and that the rewrite-rule game was the hardest to play, it was decided to test only these two games in the third study, since this should show up the biggest differences in ability.

The study was conducted with 40 children aged between 7 and 8. Twenty children played with the complete explicit and twenty played the rewrite game. To evaluate how much children enjoyed the game, both self-report and observational measures were used.

The following self-report measures of enjoyment were used:

1. Sticky ladder
2. Smiley face scale

The following observational measures of enjoyment, for both playing the game and watching the clips, were used:

1. Laughing
2. Clapping
3. Talking about character
4. Talking about clips
5. Making sounds from clip
6. Recall (mentioning events that occurred in clips when asked what they remember)

The following observational measures of ability to play the game were used:

7. Number of errors made (i.e. choosing incorrect tile)
8. Requests for help
9. Physical help provided by the experimenter
10. Verbal prompting by experimenter
11. Clicktil ("clicking ahead" on dominoes before scene has finished playing)
12. Thinktil ("thinking ahead", i.e. moving mouse over next domino prior to scene finishing)

9.3.1 Results of Study 3

The statistical analysis of the data, comparing the complete explicit and rewrite conditions, is shown below.

Table 6 shows the mean ratings given by children for the two rating scales (sticky-ladder and smiley-face scale), when they were playing the "rewrite rule" game and the "complete explicit" game. The highest possible enjoyment rating is *5* on the smiley-face scale and *7* on the sticky-ladder scale. Therefore, Table 6 shows that both games rated quite highly in terms of enjoyment according to either scale. There was no significant difference between the conditions in terms of how much the children said they enjoyed the game, according to either rating scale.

[Table 6]

Table 7 shows the observational enjoyment data from children playing the "rewrite rule" game compared with children playing the "complete explicit" game. An observational score was obtained for each child by counting the number of times each measure occurred, e.g. the number of claps, smiles etc. There were no significant differences in any of these measures, they also have very low means.

[Table 7]

Table 8 shows the data from the observational measures of ability. There were two significant differences with these measures. Children in the rewrite condition made significantly more requests for help and required significantly more verbal prompting. Thus it does appear the rewrite was more difficult for the children, although this was not shown in either the self reported or observational measures of enjoyment. It would therefore appear that whilst older children still had some difficulties with the rewrite game, their enjoyment was not as affected by the difficulty level as it was for younger children. However, given the low sample size of the previous study, it is necessary to be cautious about making comparisons.

[Table 8]

9.4 Study 4

The fourth evaluation was conducted in order to investigate the narrative properties of Scene-Driver. The study compared the playing of a narratively organised game with a game in which the clips were selected in an arbitrary way. Additional to the previous studies, at the end of the game each child was shown their personal “compilation” episode that they had constructed by playing the dominoes. This episode began with the opening credit sequence from Tiny Planets and ended with a slightly abbreviated version of the closing credits from the show. These credits were altered so that at the very end, the child’s own name appeared under the “directed by” credit. The purpose of this compilation episode was to enable comparisons to be made between having the child as an active participant in constructing a new narrative compared to having the child merely view a previously reorganised episode.

To achieve these aims, the experiment was run with three conditions:

1) Interactive narrative – The child reorders content via the playing of the domino game then watches their personal episode. Content reorganisation is narrative-driven and there is a thematic similarity across all scenes.
2) Interactive non-narrative - The child reorders content via the playing of the domino game then watches their personal episode. Content reorganisation is non-narrative-driven and there is no thematic similarity across scenes.
3) Domino-only narrative – The child plays the same domino game as in the interactive-narrative condition, but only matches domino-to-domino, there are no scenes shown at this stage. Following the playing of the domino game, the child views the “narrative” version of the compilation episode.

In order to assess narrative versus non-narrative ordering of scenes, rather than the game mechanics and children's ability to match tiles, all three domino games were of the complete-explicit game-type. This game type was chosen because it was consistently found to be the easiest of the three game types for the children to play. Since there was also a thematic similarity amongst scenes in the narrative version of the game, it was decided that the non-narrative version should use a different scene-set. It was therefore necessary to control for certain variables between the two versions, such as the duration of scenes and how frequently characters appeared and the length of time that they were on screen.

Forty eight children aged between 7 and 8 participated in the study. There were sixteen participants in each condition, eight of which were male and eight of which were female. In all three conditions, once the compilation episode had been viewed, the children were asked to give a self-assessment of their enjoyment of the experience via the sticky ladder. They were then presented with a recall task. The purpose of this task was to enable an assessment to be made as to whether narrative ordering and coherence assisted memory. The recall task took the form of an identity parade of six characters, three of whom had been present in the episode and three which they had not seen. Since the clips shown in the non-narrative game were different to those shown in the narrative games, the identity parades necessarily differed in terms of the characters that could be shown. Therefore, in choosing the three characters to include that had been present in some of the clips shown, it was necessary to try and control for factors such as the overall length of time the characters appeared on screen, across how many clips they appeared and also if they were similar to other characters in such a way that confusion might occur. Figure 26 shows the identity parade for the two narrative conditions. The positioning of the characters in terms of those which were present and those which were not was the same for all three conditions.

[Figure 26]

9.4.1 Results of Study 4

There were no significant differences in the enjoyment ratings given on the sticky ladder in any of the three conditions (see Table 9), the ratings were consistently high. This is most probably due to the same game-type, of the easiest difficulty level, being used in all three conditions (previous studies suggested that game difficulty affected the enjoyment ratings given).

[Table 9]

Interesting results were achieved in the comparison between narrative and non-narrative versions of the game in the recall task. Table 10 shows the mean number of characters *correctly* identified as having been present in the scenes that were viewed. There were no significant differences between the conditions in the correct identification of characters that had been shown in the scenes. Table 11 shows the mean number of characters *incorrectly* identified as having been present in the scenes viewed, these characters being the ones which had not in fact been present in any of the scenes shown to the child. This showed a significant difference - $F(2,45) = 6.5$, $p < 0.05$. A Tukey (HSD) post-hoc comparison revealed significant differences between the Interactive Non-narrative and Interactive narrative conditions ($p < 0.01$) and also between the Interactive Non-narrative and Domino-only narrative conditions ($p < 0.01$).

[Table 10]

[Table 11]

In both the interactive narrative and interactive non-narrative conditions, children saw all scenes twice. One viewing occurred as they were constructing the narrative and placing their tiles and the second viewing occurred as they then watched the episode they had constructed. In the Domino-only narrative version, each scene was viewed only once. It is thus interesting to note that in spite of this potential drawback, children in this condition were less likely to make “false positives” in the identity parade than children in the non-narrative condition. This suggests that narrative ordering does have a positive influence on memory.

10. Comparisons between Scene-Driver and Existing Interactive Narrative Models

Now that the Scene-Driver model has been fully introduced, we can compare it to some existing models of interactive narrative that were earlier introduced in *section 2*, in order to identify the similarities and differences. The first similarity, common amongst most models of interactive narrative, is that available narrative resources are divided into segments or lexia. The second similarity, comparable to the triangle interface of Gorbet et al. (1998) and the sculptural hypertext interfaces for Card Shark and Thespis (Bernstein, 2001) and StorySpinner (Hooper and Weal, 2005), is the way in which the reader unfolds the narrative in a particular way by selecting from presented choices, i.e. either cards or tiles. Thirdly, also comparable to the sculptural hypertext models, is that a selection by the reader can consequently alter the availability of other, so far unused, options. Finally, like

StorySpinner, Scene-Driver enables the user to backtrack to previous choice points in order that the entire story can be played out.

The following outlines some differences between Scene Driver and the existing interactive narrative models. First, Scene Driver has a narrative model which specifies the stages through which the narrative should progress. According to the minimal plan, the narrative should progress through the stages of theme introduction, conflict introduction and confliction resolution. This differs to the models of Murtaugh (1996), Srinivasan (2002) and Rocchi and Zancanaro (2003) which have no plot model for reorganising content.

Second, the narrative model in Scene-Driver is centralised. In order to change the plot ordering to which a Scene-Driver narrative must adhere requires only that this centralised narrative model be modified. Similar to Scene-Driver, in sculptural hypertexts the desired narrative ordering of lexia can be assured. For example, in Card Shark cards constituting a conflict introduction would require that the theme introduction phase be the current state of the story as represented on the blackboard. Selecting one of the conflict introduction cards would result in the blackboard being updated. However, in sculptural hypertext systems there is no centralised representation of the narrative model. A change to the narrative model would require altering the constraints and effects that each individual card had. In cognitive dimensions terms, the representation is viscous (Green, 1990) as many independent changes have to be made to alter the narrative model.

Third, in Scene Driver constraints as to the applicability of each story segment are reflected in the game rules via which the reader drives the narrative. Options that cannot be selected due to narrative constraints cannot be used because the tiles by which they are represented cannot be played at that stage of the game. This provides the user with an alternative model as to why options are available and unavailable at particular stages. This contrasts to Card Shark, for example, where cards are simply disabled if preconditions are not met

Fourth, unlike any existing sculptural hypertext systems, Scene Driver has a look-ahead algorithm. This ensures that the collection of tiles given to the user can be successfully used to produce a complete narrative of a specified length. This also ensures that at any stage all tiles that are playable according to the chosen game rules will produce a scene that is consistent with the narrative plan.

Fifth, in Scene Driver the tiles by which the next step in the narrative are selected can, as well as the scenes themselves, alter the story state. For example, when adopting the rewrite rule game, the properties shown on the left hand side of the tile (e.g. characters) are voted out and those on the right hand side of the tile are voted in. In existing sculptural hypertext systems the story state can only be changed by the story segment, not by the method of selection. In the original Card Shark, each card available for selection represented a specific lexia. Story Spinner uses Tarot cards as the method of selection, and each Tarot card may potentially point to more than one available lexia. If there is a choice of lexia, then criteria are used to select the most appropriate. Similarly, Scene Driver potentially has a one-to-many mapping of tiles to clips. For example, this is used when a tile is playable at different narrative stages depending on the other choices made. However, Scene Driver is unique in having a selection mechanism that can itself alter the story state.

11. Conclusions

Scene-Driver comprises a set of tools for the sub-division and reorganisation of content according to chosen constraints. The model specifies the following generic requirements for content re-use:

- The sub-division of content into a library of discrete units, each unit depicting one main event.
- The selection and ordering of content, from the library of units, according to a chosen criteria to ensure a coherent presentation.
- Annotation of content according to both visual and conceptual properties.
- An interface via which the user can traverse the content.

The model has been implemented using content from the Tiny Planets television series. This implementation instantiates and extends the generic model in the following way:

- Tiny Planets content is sub-divided into units, each depicting one main event.
- Content is selected and ordered according to narrative criteria.
- Content is annotated according to visual properties, such as characters and props, as well as conceptual properties, such as the plot-element-type of the main event and the underlying theme.
- The user traverses content via the domino-tile interface.

In essence, the Tiny Planets demonstrator is an interactive narrative-driven game using “domino” like tiles, in which a child actively collaborates with Scene-Driver to construct novel narratives. In line with studies into narrative and memory, evaluations conducted with the demonstrator comparing narrative with non-narrative versions of the game, have shown evidence that the introduction of narrative constraints assists memory. The evaluation studies have also yielded positive results in terms of both self-assessed and observational measures of the children’s enjoyment. Although none of the four evaluations addressed longevity of interest in the game, such issues could be tackled in future studies.

A second demonstrator has been developed which implements algorithms for generating tile-sets to create legal games within narrative constraints. A legal game is one in which, each time a user plays, the scenes are presented in a coherent fashion according to a minimum narrative plan, regardless of the user’s tile choice. The algorithm generates tile-sets of varying lengths, enabling a user to specify how many tiles they want to have available in a given game. Game difficulty can also be varied by either allowing a user only a single tile-choice at each stage of a game (the simplest scenario) or by permitting a user to choose from two or more tiles at certain points in the game, to which they can retrace their steps and try alternative tile-choices.

The Scene-Driver model itself is not limited to use with children’s television content. Rather, the content descriptions, interaction mechanism and ordering-constraints are flexible to permit alternatives to be utilised, depending on the targeted user and the content-type. With regard to content type, it is apparent that certain types of content will be more suitable than others to be implemented within the model. For example, static images depicting objects, people and events can lend themselves equally well to the types of descriptions needed for use in the model, whereas abstract art may not. Documentary-type factual or historical content may be used in place of fictional material. A further possibility might be to provide tools for users to construct appropriate annotations of their personal repositories of digital video and images stored on their PC.

In terms of the interaction mechanism, we have previously described two levels at which this can be conceptualised, with a command being based either solely on the next state or else being based on both the current and the next state. It is possible to extend the interaction device further, for example by including additional properties on the interface that the user must consider when matching tiles. These properties might encode narrative principles so that narrative ordering is made explicit via the interface. Alternatively, narrative principles may be omitted or replaced with different criteria for reorganising content. In particular, for older children and adult users, there is the possibility to replace the domino tiles with a more appropriate interaction method, without the necessity to be purely icon-based or to constrain the options to the same extent that is desirable for younger users.

Additionally, different platforms may be used for delivering the game. In fact, we have successfully built prototype demonstrators of the game in both Shockwave ™ and as an interactive DVD using a special Berchet ® DVD remote controller, aimed for use by children (http://www.dvd-kids.com).

Taking into consideration all of the above, it can be seen that Scene-Driver provides a solid foundation for the creation of interactive games re-using content to create engaging novel experiences.

Acknowledgements

The Tiny-in project was funded by EPSRC grant GR/R54378/01. We would like to thank Pepper’s Ghost for providing the Tiny Planets content and for offering support during the project. We also wish to thank the school children and teachers who cooperated in the evaluation studies.

References

Adams, E., 1999. Three problems for interactive storytellers. Consulted February 20, 2003. http://www.gamasutra.com/features/designers_notebook/19991229.htm

Airey, A., Plowman, L., Connolly D., Luckin R., 2002. Rating children's enjoyment of toys, games and media. Paper presented at 3rd World Congress of the International Toy Research Association on Toys, Games and Media, London.

Bernstein, M., 2001. Card Shark and Thespis: Exotic tools for hypertext narrative. Proceedings of the twelfth ACM conference on Hypertext and Hypermedia, Århus, Denmark, pp. 41-50.

Bocconi, S., 2004. Semantic-Aware Automatic Video Editing. In: 12th ACM International Conference on Multimedia - Doctoral Symposium, pp. 971-972.

Branigan, E., 1992. Narrative Comprehension and Film. Sightlines, ed. E. Buscombe. Vol. 1. Routledge, New York.

Brooks, K. M., 1996. Do story agents use rocking chairs? The theory and implementation of one model for computational narrative. Fourth ACM international conference on Multimedia, Boston, USA, ACM Press, pp. 317-328.

Cavazza, M., Charles F., Mead, S.J., 2002. Character-based interactive storytelling. Interactive Entertainment, pp. 17-24.

Chatman, S., 1978. Story and Discourse: Narrative Structure in Fiction and Film. Cornell University Press, London.

Davis, M., 2003. Editing Out Video Editing. IEEE MultiMedia, Volume 10, Issue 2, pp. 54-64.

Decortis, F., Rizzo, A. and Saudelli, B., 2003. Mediating effects of active and distributed instruments on narrative activities. Interacting with Computers. Volume 15, Issue 6, pp. 801-830.

Genette, G., Lewin, J. E., 1983. Narrative discourse: An essay in method. Cornell University Press, Ithaca, NY.

Gorbet, M., Orth M., Ishii, H., 1998. Triangles: tangible interface for manipulation and exploration of digital information topography, in proceedings of CHI '98, Los Angeles CA, pp. 49-56.

Green, T. R. G., 1990. The cognitive dimension of viscosity - a sticky problem for HCI. In D. Diaper and B. Shackel (Eds.) INTERACT '90. Elsevier.

Hayes-Roth, B., Brownston, L., Sincoff, E., 1995. Directed improvisation. Stanford University Report KSL-95-04.

Hooper, C. J., Weal, M. J., 2005. The StorySpinner Sculptural Reader. Proceedings of the sixteenth ACM conference on Hypertext and Hypermedia, Salzburg, Austria, pp. 288-289.

Jacob, R. J., Ishii, H., Pangaro, G., Patten, J., 2002. A tangible interface for organizing information using a grid. In Proceedings of CHI 2002, New York, ACM Press.

Landow, G., 1992. Hypertext: The Convergence of Contemporary Critical Theory and Technology. Johns Hopkins UP.

Mateas, M., Stern, A., 2002. A Behaviour Language for Story-Based Believable Agents. IEEE Intelligent Systems. pp. 39-47.

Mazalek, A., Davenport, G., Ishii, H., 2002. Tangible Viewpoints: A Physical Approach to Multimedia Stories. In Proceedings of ACM Multimedia '02, Juan-les-Pins, France, ACM Press, pp. 153-160.

Monaco, J., 1977. How to read a film: The art technology, language, history and theory of film and media. New York: Oxford University Press.

Murray, J.H., 1997. Hamlet on the Holodeck: The Future of Narrative in Cyberspace. New York: The Free Press.

Murtaugh, M., 1996. The automatist storytelling system: putting the editor's knowledge in software. Masters Thesis, MIT Media Lab.

Nack, F., 1996. AUTEUR: The Application of Video Semantics and Theme Representation in Automated Video Editing, Ph.D. thesis, Lancaster University.

Repenning, A., Ioannidou, A., Ambach, J., 1998. Learn to Communicate and Communicate to Learn. Journal of Interactive Media in Education, 98 (7) [www-jime.open.ac.uk/98/7].

Rocchi, C., Zancanaro, M., 2003. Generation of video documentaries from discourse structures, in Proceedings of the 9th European Workshop on Natural Language Generation.

Ryan, M. L., 2001. Beyond Myth and Metaphor: The Case of Narrative in Digital Media. Game Studies, volume 1.

Ryokai, K., Cassell, J., 1999. StoryMat: A Play Space for Collaborative Storytelling. In Proceedings of CHI '99, ACM.

Ryokai, K., Vaucelle, C., Cassell, J., 2003. Virtual Peers as Partners in Storytelling and Literacy Learning. Journal of Computer Assisted Learning, Volume 19, Issue 2, pp. 195-208

Schank, R., 1990. Tell me a story – A new look at real and artificial memory. New York: Macmillan.

Srinivasan, R., 2002. Village voice: Expressing narrative through community-designed ontologies. Masters Thesis, MIT.

Szlias, N., 1999. Interactive drama on computer: beyond linear narrative. In Proceedings AAAI Fall Symposium on Narrative Intelligence, North Falmouth MA, AAAI Press, pp. 150-156.

Thorndyke , P., 1977. Cognitive structures in comprehension and memory of narrative discourse. Cognitive Psychology, 9, pp. 77-110.

Ullmer, B., Ishii, H., Glas, D., 1998. MediaBlocks: Physical Containers, Transports, and Controls for Online Media. In Proceedings of SIGGRAPH '98, Orlando, Florida USA, July 1998, ACM Press, pp. 379-386.

Walker, J., 1999. Piecing together and tearing apart: finding the story in *afternoon*. Proceedings of the Tenth ACM Conference on Hypertext and Hypermedia, Darmstadt, Germany, pp. 111-117.

Weal, M. J., Michaelides, D. T., Thompson, M. K., de Roure, D. C., 2003. Hypermedia in the ambient wood. New Review of Hypermedia and Multimedia, 9, pp. 137-156.

Figures

Figure 1: Tiny Planets Characters. The two central white characters are Bing (the larger creature) and Bong and the other characters are a mix of flockers, locals and a robot.

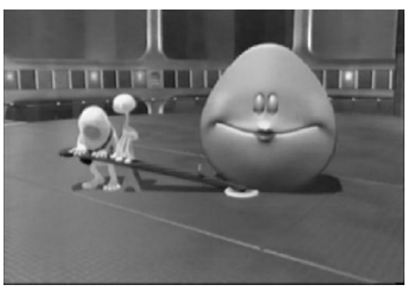

Figure 2: A screenshot from the Tiny Planets series. The main characters, Bing and Bong, are using a seesaw as a lever to move a large heavy 'local'.

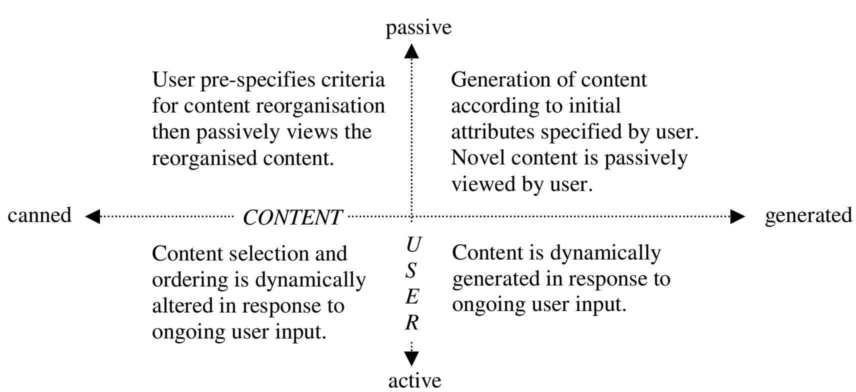

Figure 3: Potential approaches for content re-use

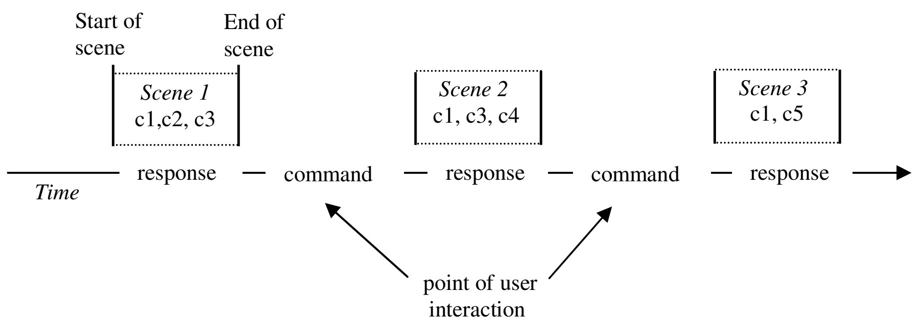

Figure 4: A chain of commands and responses

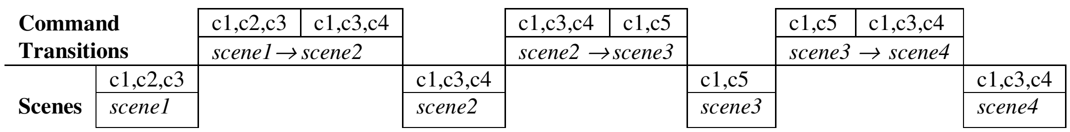

Figure 5: Example of commands that specify transitions from the current to the next scene

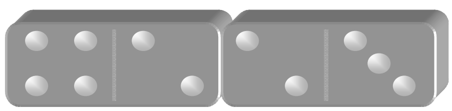

Figure 6: Standard domino matching with two tiles

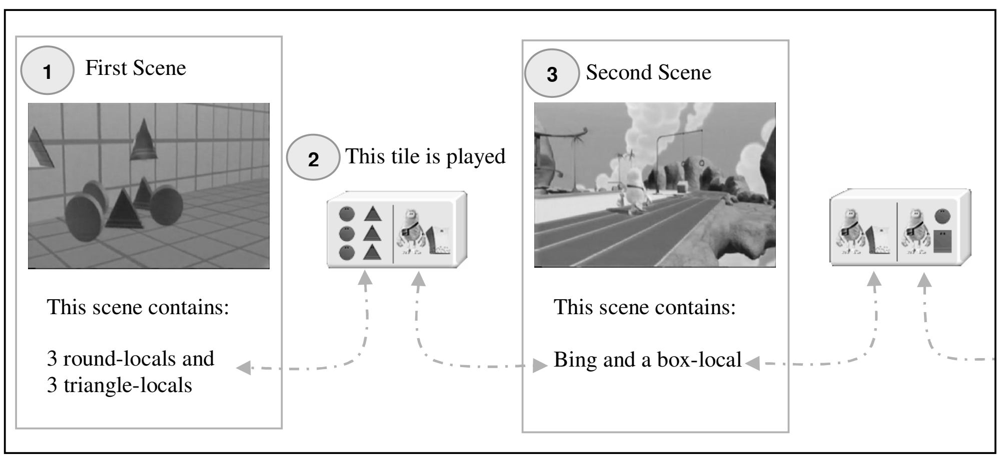

Figure 7: Legal tile matching in Complete-Implicit game. 1) The first scene is played. This contains "3 round-locals, 3 triangle-locals", so a legal tile must have these characters on the left-hand-side. 2) A legal tile is placed against this scene. The right-hand-side of this tile contains "Bing, box-local", thus specifying that the next scene must contain these two characters only. 3) A scene containing only "Bing, box-local" is played.

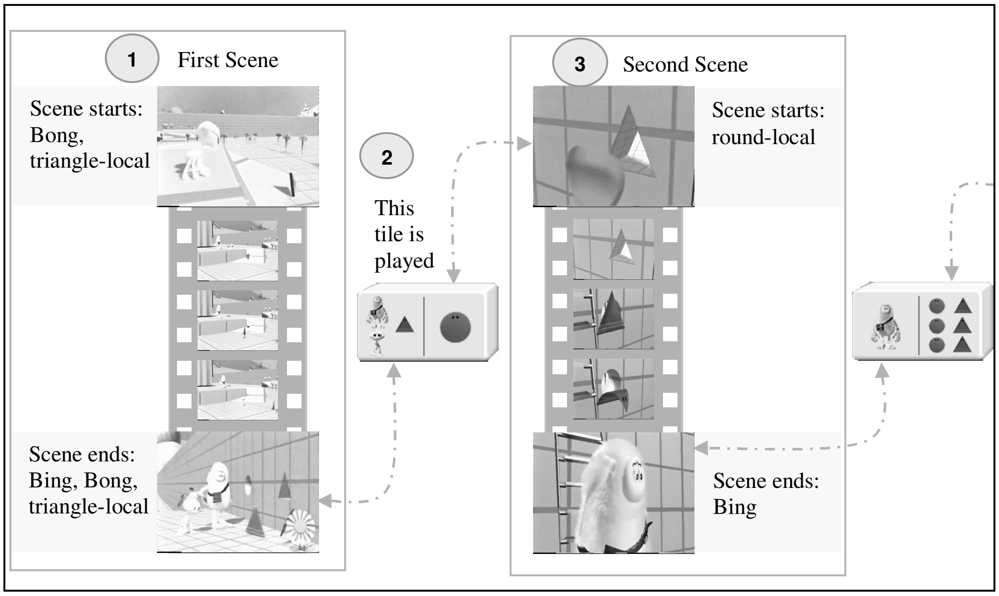

Figure 8: Legal tile matching in Complete-Explicit game. 1) The first scene is played. This scene starts "Bong, triangle-local" but *ends* "Bing, Bong, triangle-local", so a legal tile must have these three characters on the left-hand-side. 2) A legal tile is placed against this scene. The right-hand-side of this tile contains "round-local", thus specifying that the next scene must *start* with this character. 3) A scene which starts with "round-local" is played.

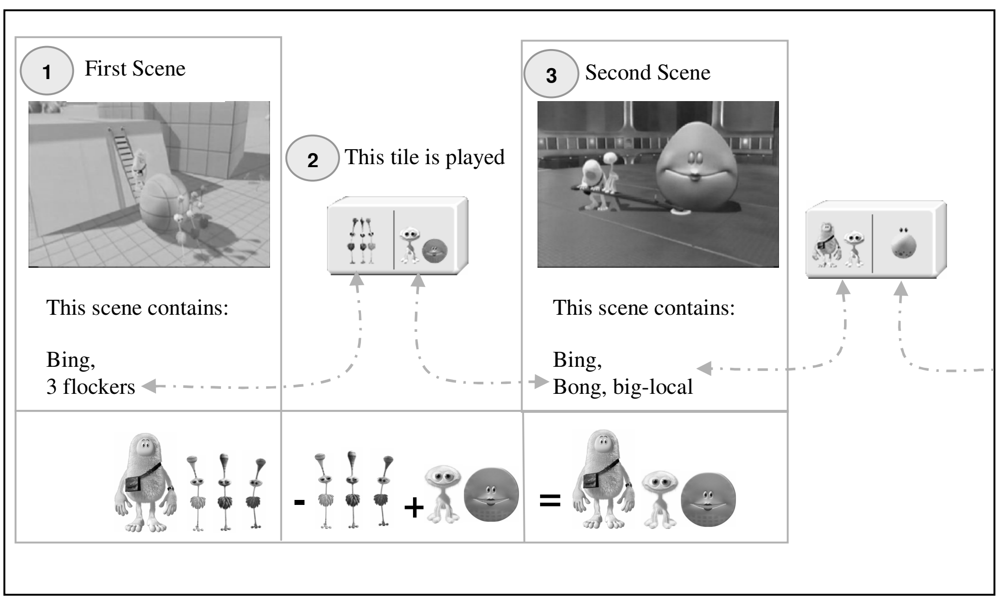

Figure 9: Legal tile matching in Rewrite-Rule game. 1) The first scene is played. This scene contains "Bing, 3 flockers", so a legal tile must contain any or all of these characters. 2) A legal tile is placed against this scene. This tile contains "3 flockers" on the left-hand-side and "bong, big-local" on the right-hand-side, thus specifying that in the next scene "3 flockers" must be replaced by "bing, big-local", while Bing remains. 3) A scene containing "Bing, Bong, big-local" is played.

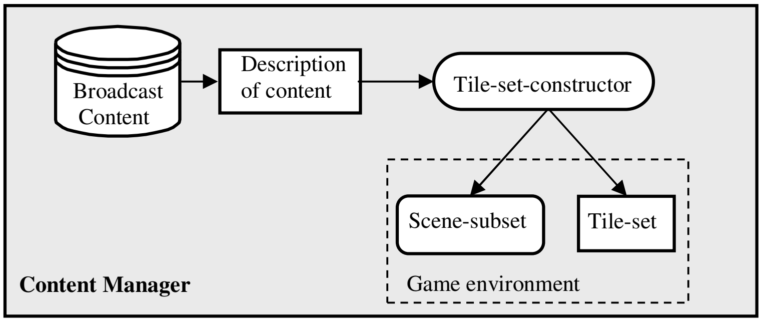

Figure 10: The content manager. Content description enables the tile-set-constructor to select a scene-subset according to a chosen theme and to generate a legal tile-set from this subset. A legal-tile-set and the scene-subset form part of the game environment.

S1 (Scene 1)
has-characters bi, bo, t-l
has-start-state bo, t-l
has-end-state bi, bo, t-l

S2 (Scene 2)
has-characters bi, t-l, r-l
has-start-state r-l
has-end-state bi

S3 (Scene 3)
has-characters bi, b-l
has-start-state bo, b-l
has-end-state b-l

Key (for all character examples in this section)

bi = Bing
bo = Bong
t-l = triangle-local
r-l = round-local
b-l = big-local
rsf = red-stuff-flocker
gsf = green-stuff-flocker
bsf = blue-stuff-flocker

Figure 11: Three scene descriptions taken from the Tiny Planets content.

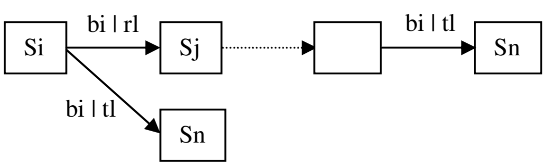

Figure 12: Choice-points in tile-sets. Some tile-sets may contain two or more tiles with the same left-hand-side and differing right-hand-sides. The top and bottom rows show two alternative paths through a set of scenes. If the 'bottom row' choice is made, then the user will need to backtrack and make the alternative tile placement in order to view all scenes in this scene set.

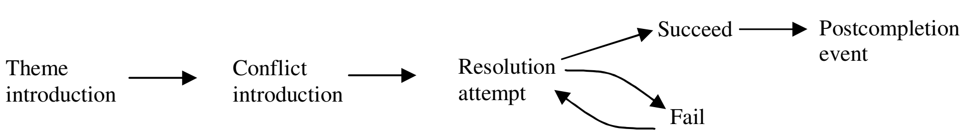

Figure 13: The minimum narrative plan, developed from analysis of Tiny Planets episodes and narrative theory

SCENE	
has-file-location	62-40-42
has-description	Bing tries to use a ladder to lever a ball up onto the top of a ramp, but fails
has-characters	Bing, bong, blue-stuff-flocker, green-stuff-flocker, red-stuff-flocker
has-props	ladder, large-ball
has-actions	Lever
has-start-characters	Bong, blue-stuff-flocker, green-stuff-flocker, red-stuff-flocker
has-end-characters	Bing
has-themes	heavy-objects
has-plot-element-type	conflict-introduction, resolution-attempt
Has-conflict-resolution	62-49-54
has-episode	62

Figure 14: Introduction into scene-description of attributes for determining plot-element-type (attributes are highlighted).

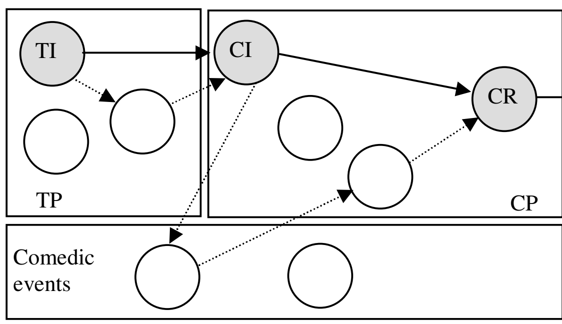

Figure 15: A graphical representation of narrative-driven path construction. Each circle represents a scene of a specific plot-type. The solid arrows depict the minimum narrative path, the dotted arrows depict one of many other potential pathways.

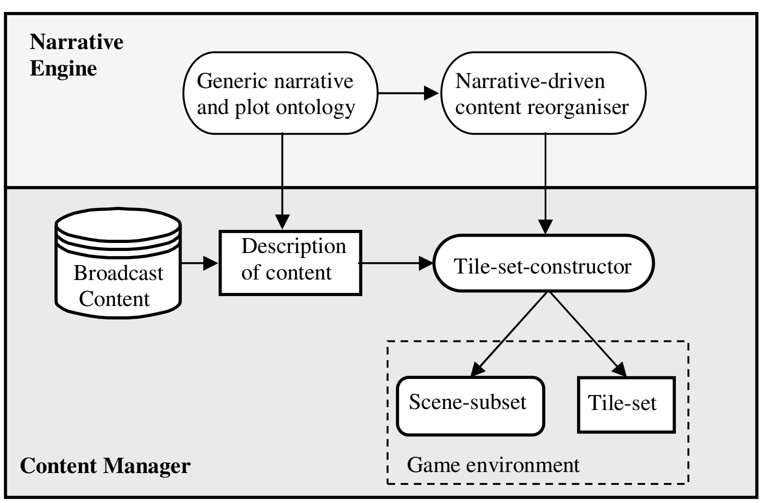

Figure 16: The Narrative Engine and Content Manager. Content description enables the tile-set-constructor to select a scene-subset according to a chosen theme and to generate a legal tile-set from this subset, according to the narrative constraints specified by the narrative engine. A legal-tile-set and the scene-subset form part of the game environment.

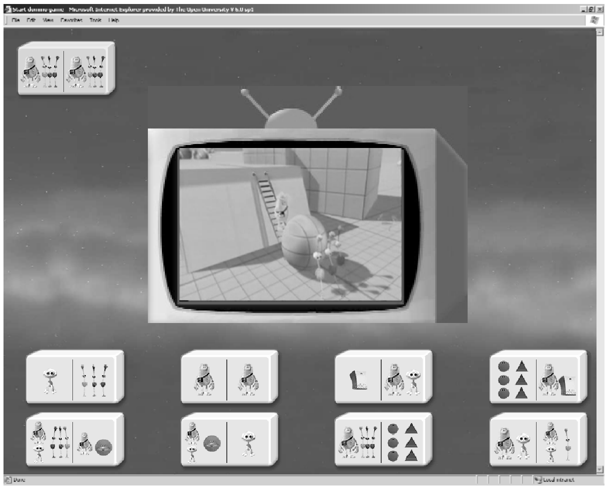

Figure 17: Starting the game (taken from complete-explicit)

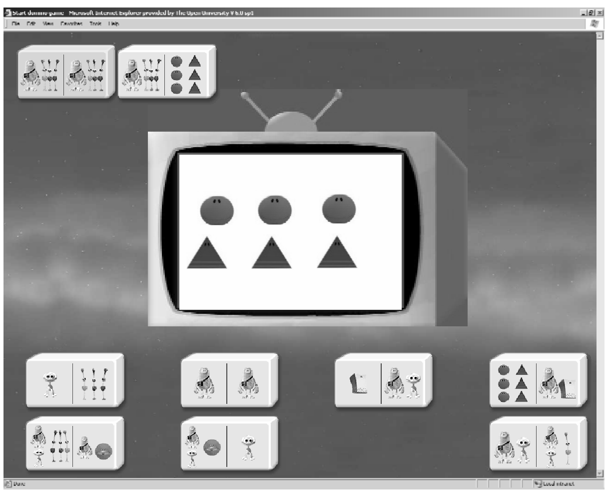

Figure 18: The transition showing the 6 locals arriving on the television (taken from complete-explicit)

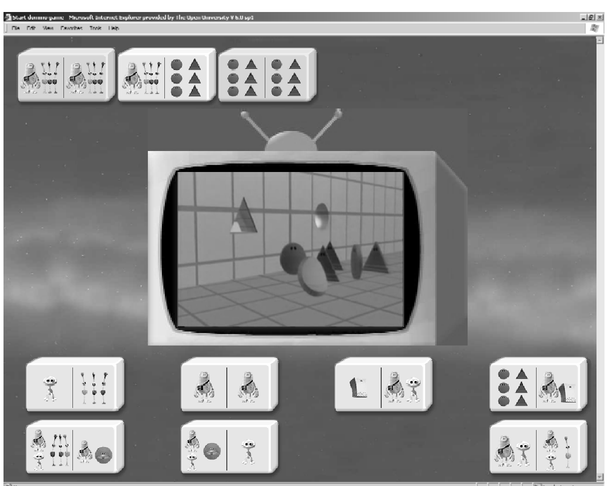

Figure 19: The next scene (taken from complete-explicit)

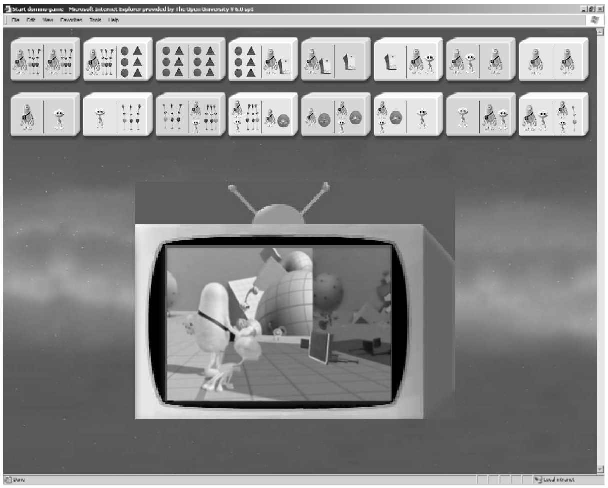

Figure 20: A completed game (taken from complete-explicit)

Figure 21: Child-friendly DVD remote-controller from Berchet ™, with simple 3-domino overlay sheet to demonstrate potential for implementing Scene-Driver game.

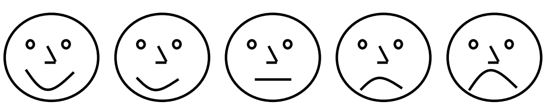

Figure 22: The smiley-face rating scale.

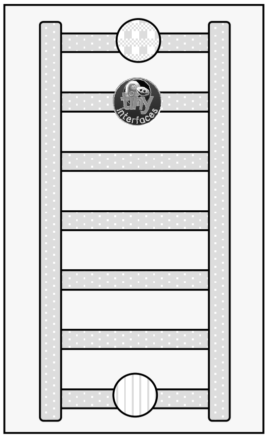

Figure 23: The sticky-ladder rating scale. The discs are ordered by the child from top (most favourite) to bottom (least favourite).

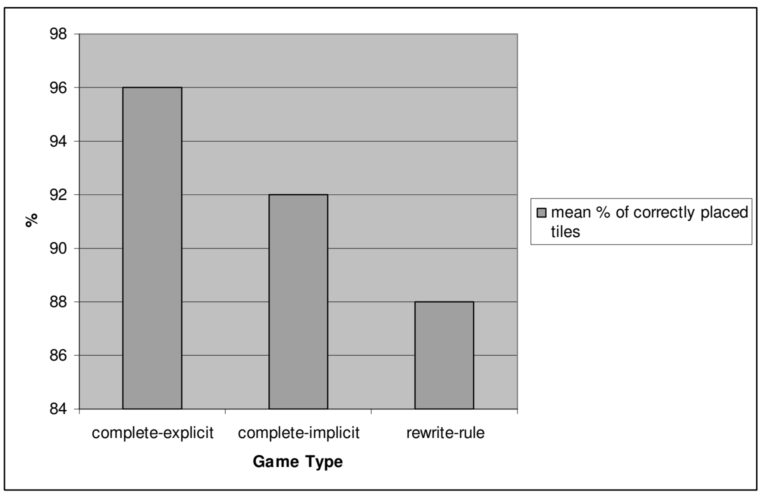

Figure 24: Percentage of correctly placed tiles in each condition of the domino game

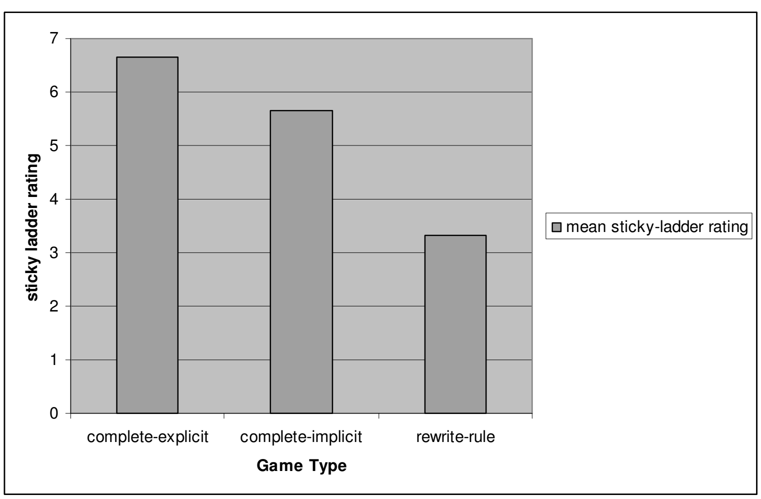

Figure 25: Mean rating on sticky ladder in each condition of the domino game

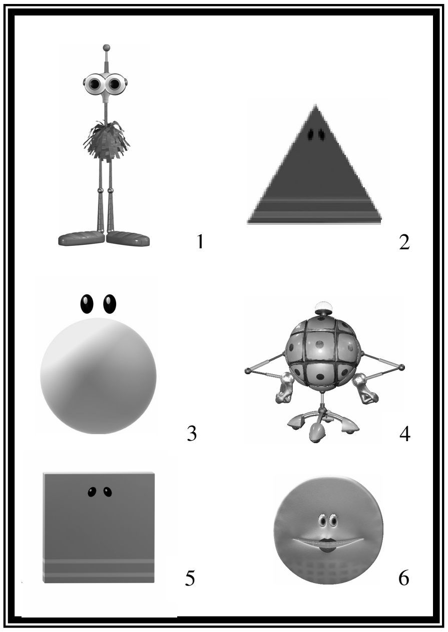

Figure 26: The identity parade for the narrative conditions. Characters 2, 3 and 5 were present in scenes viewed by the children, the other characters did not appear in any scenes viewed by the children.

Tables

No.	Command	Response
1	Has-property-x: c1, c3, c4	Scene 2
2	Has-property-x: c1, c5	Scene 3
3	Has-property-x: c1, c3, c4	Scene 4

Table 1: Example of commands that specify characteristics of the next scene

SCENE	
has-file-location	62-40-42
has-description	Bing tries to use a ladder to lever a ball up onto the top of a ramp, but fails
has-characters	Bing, bong, blue-stuff-flocker, green-stuff-flocker, red-stuff-flocker
has-props	Ladder, large-ball
has-actions	Lever
has-start-characters	Bong, blue-stuff-flocker, green-stuff-flocker, red-stuff-flocker
has-end-characters	Bing
has-themes	Heavy-objects
has-episode	62

Table 2: Scene described by some attributes relevant for constructing legal games. The file-location provides a unique identifier for locating the actual media for the scene, specifying the episode from which the scene is taken and the frame-numbers which the scene spans.

		S1	S2	S3
	Start-states / *End-states*	bi, bo, t-l	r-l	bo, b-l
S1	bi, bo, t-l	x	bi, bo, t-l | r-l	bi, bo, t-l | bo, b-l
S2	bi	bi | bo, t-l	x	bi | bo, b-l
S3	b-l	b-l | bo, t-l	b-l | r-l	x

Table 3: Scene state information for generating tiles for Complete Explicit game

	S1	S2	S3
has-characters	bi, bo, t-l	bi, t-l, r-l	bi, b-l

Table 4: Scene state information for generating tiles for Complete Implicit game

	S1	S2	S3	S4
S1	{ } S1 - S1	rsf, gsf, bsf S1 - S2	{ } S1 - S3	rsf, gsf, bsf S1 – S4
S2	bo, b-l S2 - S1	{ } S2 - S2	b-l S2 - S3	b-l S2 – S4
S3	bo S3 - S1	rsf, gsf, bsf S3 - S2	{ } S3 - S3	rsf, gsf, bsf S3 – S4
S4	bo, r-l S4 - S1	r-l S4 - S2	r-l S4 - S3	{ } S4 – S4

Table 5: Scene state information for generating tiles for Rewrite Rule game

	Software Rewrite		Complete	
	Mean	Std Deviation	Mean	Std Deviation
Smiley	4.60	.68	4.40	.68
Sticky	5.60	1.67	5.35	1.35

Table 6: Self-report measures

	Software Rewrite		Complete	
	Mean	Std Deviation	Mean	Std Deviation
Smiling	.1	.2	.1	.2
Clapping	.0	.0	.1	.2
Character	.9	1.8	.5	1.0
Clips	.8	1.5	.6	1.1
Sounds	.3	.7	.1	.3
Recall	1.05	1.50	1.05	1.36

Table 7: Observational measures of enjoyment

	Software Rewrite		Complete			
	Mean	Std Deviation	Mean	Std Deviation	t (df = 38)	p
Number of errors	1.0	1.1	.8	1.0	0.47	NS
Requests for help	2.7	1.5	1.7	1.5	2.17	< 0.05
Physical help	.2	.4	.1	.3	0.47	NS
Verbal prompting	4.8	2.8	2.6	2.0	2.17	< 0.05
Clicktil	.0	.0	.4	1.4	-1.32	NS
Thinktil	.2	.5	.2	.5	0.31	NS

Table 8: Observational measures of ability

Interactive Narrative		Interactive Non-Narrative		Domino-only narrative	
M	SD	M	SD	M	SD
5.3	1.1	5.1	1.2	5.3	1.2

Table 9: Sticky-ladder ratings

Interactive Narrative		Interactive Non-Narrative		Domino-only narrative	
M	SD	M	SD	M	SD
2.6	.63	2.4	.62	2.3	1.0

Table 10: Identity parade – correctly identifying characters that were present in scenes

Interactive Narrative		Interactive Non-Narrative		Domino-only narrative	
M	SD	M	SD	M	SD
.00	.000	.56	.73	.13	.34

Table 11: Non-identity parade – identifying characters as having been present in the scenes when these characters had not been shown