Designing the User Interface of an Interactive Software Environment for Children

D. Grammenos, A. Paramythis, C. Stephanidis

Institute of Computer Science, Foundation for Research & Technology – Hellas
Science and Technology Park of Crete, Heraklion, Crete, GR-71110 GREECE
e-mail: {gramenos, alpar, cs}@ics.forth.gr

Abstract. This design experience paper reports on the preliminary design of the user interface of an interactive software environment for children 4-8 yeas old, which supports the creation of an 'electronic diary'. The paper focuses on the design rationale, and on the specification of the interaction with the system. More specifically, the outcomes of the interaction design process (i.e., a user interface specification) are presented, along with the issues and problems that had to be tackled, as well as the adopted solutions and the followed assumptions and conventions.

1. INTRODUCTION

The "Today's Stories" project (see Acknowledgements) aims to develop the social, communicative and emotional skills of children through a collaborative reflective activity based on the interesting events that take place during a day. This is to be accomplished by empowering children to create a diary of interesting events during the day. The creation of the diary consists of two stages. First, during the day, children can capture events by using a wearable camera (the KidsCam). The same event can be captured by different children, thus providing alternative views. Then, by the end of the day, the collected material is transferred (automatically, through a wireless network) to a workstation (the Magic Mirror), where children can view, annotate and edit it, by using an appropriate software tool (the Diary Composer), thus constructing their Today's Stories. The users of the Diary Composer are children between 4 and 8 years old, from two different countries (Denmark and Israel), as well as their teachers.

During the early phases of the project, requirements and context analysis activities, as well as brainstorming and discussion sessions took place and cooperative inquiry techniques (Druin, 1999a) were employed (whith children) in order to shape the requirements and the high-level concepts upon which the user interface was to be developed (Today's Stories Consortium, 1999; Koutra et al., 2000). This paper builds upon the outcomes of the aforementioned activities and reports on the preliminary design of the user interface of the Diary Composer, focusing on the design rationale, and on the specification of the interaction with the system.

The high-level goals of the performed design activity were to:

- Convey available functionality in a highly visual form and do not rely on textual representations, in order for the system to be language independent as well as usable by younger children that can not read, since "*highly visual menus and icons appear to be appealing to children and easy for them to understand and use*" (Wilson, 1988).

- Create an *open learning system*[1] (Jonassen et al., 1993) which can be adapted to suit the children preferences and cultural background (e.g., by adding self-made annotation symbols) and skills (e.g., by customising the functionality offered).
- Incorporate interactive elements and provide adequate metaphors (Erickson, 1990) that are intuitive both in terms of the function they represent, but also in terms of how they are to be operated upon (i.e., provide *affordances* (Norman, 1988)).
- Make the presentation and interaction appealing to children by making all the components of the user interface (inter)active and by providing feedback to indicate 'successful' interaction steps (Norman, 1988; Cooper, 1995), through animation (Baecker, Small, 1990) and audio effects (Mountford, Gaver, 1990) that on the one hand facilitate the comprehension of the concepts, and on the other hand promote and support exploratory styles of interaction.
- Create a 'forgiving' environment where there is no 'incorrect' or 'wrong' input and where active support and guidance is offered whenever needed (Cooper's "*Don't make the user look stupid*" and "*Make errors impossible*" design principles (Cooper, 1995)).
- Be gender-neutral (but also gender-customizable) and avoid the pitfalls of gender-oriented (and usually male-oriented) design (Furger, R., 1998).
- Avoid the use of cumbersome input devices (e.g., keyboard, mouse) and interaction techniques (e.g., double-click).

The ultimate goal was to provide a *transparent* user interface, so that children can focus their energies on their activities and not on the interface (Norman, 1990), since if the children's attention is focussed on the interface itself, then it gets in the way of exploration and knowledge construction (Winn, 1993).

2. DESIGNING THE USER INTERFACE

One of the problems encountered when designing the user interface, was that, although several related software products[2] targeted to the same user groups (a number of them are described in: Today's Stories Consortium, 1999; and in: Druin, Solomon, 1996) is available in the market, the relevant literature is relatively limited. Usually, what is described in literature is the final interface design and the high-level process followed to achieve it, but often a number of issues are neglected or missing, such as: (a) user interface design alternatives and decisions; (b) the design rationale; and (c) empirical data concerning the usability and value of the designs after testing them with children, and on how testing affected the redesign of the system.

Some helpful general user interface design guidelines for children were found in (Nicol, 1990) and (Druin, 1999a), as well as an illustrating example of a movie authoring and design system (but targeted to 12- to 14-year-old children). Furthermore, a comprehensive list of guidelines for the design of educational software is provided in (Temple University, 1999), but it is oriented towards course-based software.

[1] According to (Jonassen et al., 1993) *open learning systems* are those which are: (a) need driven; (b) learner-initiated; and (c) conceptually and intellectually engaging.

[2] Although the system described in this paper shares some common aspects with them, it differs considerably in respect to its philosophy, educational goals and offered functionality.

Along with the above, children- and education-related sources, design knowledge, experiences, and collections of guidelines on more ‘conventional’ user interfaces were also taken into account, such as (Galitz, 1997; Howlett, 1996; Norman, 1988; Shneiderman, 1998; Tognazzini, 1996; Weinschenk et al., 1997; Wood, 1998), as well as sources on cultural diversity issues (Galdo & Nielsen, 1996; Fernandes, 1995; Nielsen, 1990). Additionally, an interesting insight and guidelines on direct manipulation and drag and drop can be found in (Cooper, 1995).

The following sections describe the outcomes of the interaction design process (i.e., a user interface specification), along with the issues and problems tackled in the design phase, the adopted solutions and the followed assumptions and conventions.

2.1. Functional Specification

The following are the principal functional specifications of the Diary Composer (Today’s Stories Consortium, 1999):

The Magic Mirror ‘mode’
A Magic Mirror metaphor is used to familiarise children with the system. When the system is inactive it emulates a mirror by displaying on its screen the video input that is captured through a video camera positioned on top of the workstation.

The Video Explorer ‘mode’
When a child, or more, holding KidsCams approach the Magic Mirror, the Workstation switches to video review mode where all the video clips captured by the KidsCams that are in a short range are presented as thumbnails. The children can select, delete or send (to a friend) a video clip, or can review the clips of a previous day. If more than one video clips are related to the same episode, then they are presented linked to each other, so that a multiple perspective view of the event can be provided.

The Video Composer ‘mode’
When one, or more, video clips are selected from the Video Explorer, the system enters to annotation mode. Here children can collaboratively annotate the selected video clips. Children are also able to insert and remove picture, sound effects and voice annotations. Furthermore, a mechanism for extending the collection of the existing annotation symbols / sounds is provided. Video playback functions are also be supported (e.g., play, stop, fast), as well as volume control and a mechanism to close some of the opened video clips (in order to concentrate in one or two of them). Finally, children can delete video clips, switch the annotations on and off, save the annotated video clips and of course return back to the Video Explorer.

In addition to the above, on-line, context-sensitive help to all tasks is also provided.

Because of the nature of the system and the particular characteristics of the target user group, a touch screen was selected as the preferable input device, but mouse (or equivalent pointing device) input is also supported, in order to enable use of the system on ordinary PCs (e.g., for home use).

2.2. The Magic Mirror

The user interface works like a ‘mirror’. The input from a camera positioned on top of the magic mirror (workstation) is displayed (inverted to simulate a mirror) on the screen. When one or more children approach within a predetermined distance from the Magic Mirror (this is detected through their camera signals) the user interface automatically switches to ‘Video Composer’ through a ‘dissolve’ effect.

2.3. The Video Explorer

2.3.1. Requirements and constraints

According to the project specifications (Today’s Stories Consortium, 1999), a ‘timeline’ metaphor was adopted for representing the functionality related to reviewing the video clips stored in the KidsCam. A timeline is a horizontal line, spanning from the left to the right side of the screen, on which thumbnails of the video clips are placed according to the point in time at which they were captured (see Figure 1).

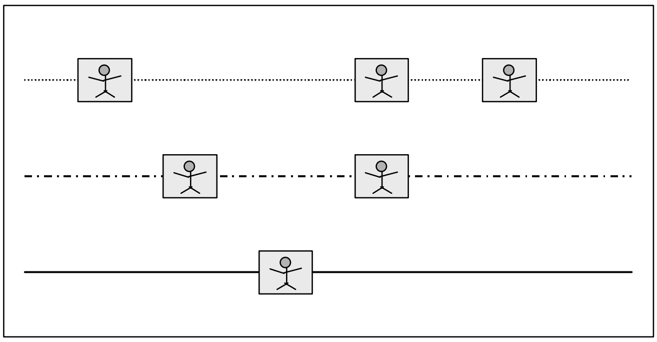


Figure 1: A timeline

A number of requirements concerned the use of the timeline metaphor (Koutra et al., 2000):

(a) Due to both technical and complexity reasons, the maximum number of concurrently visible timelines was limited to 3. All the available video clips from a single day should be visible in the timeline window, so as to avoid back and forth movement (i.e., scrolling) in the timelines.

(b) Children should be allowed to delete video clips from the timeline and have access to a previous day’s “Today’s Story”. [Initially, an upper limit of 4 video clips per timeline was assumed, but it was also required to accommodate a considerably larger number of video clips.]

(c) If a video clip was annotated, the annotated version (including maybe an icon representing the ‘annotated’ status of the icon) should be presented on the timeline instead of the original one.

(d) Video clips referring to the same episode should be explicitly linked and should act as a group (e.g., if any one of them was selected then all of them should be selected).

In the light of the above mentioned requirements and constraints the design of the Video Explorer’s user interface addressed a number of issues which are discussed in detail in the following subsections.

2.3.2. Content and meaning of the timelines

A timeline contains video clips that are captured within a single day. In other words, the timelines within the 'Video Explorer' represent a time frame of *today*. Two design options were considered:

(a) the timelines would always represent the same (fixed) time frame (e.g., from 8 a.m. to 2 p.m.); or

(b) they would represent a variable time frame which is proportional to the period in which the separate video clips were captured.

If option (a) was selected, in the case that all the video clips were captured in a short period (e.g., between 9 a.m. and 11. a.m.) more than half of the available space would be left blank (Figure 2). Since the available screen estate is quite limited and the size of video thumbnails should be maximised, the part of the timelines that have no thumbnails (the 'white space') should be minimised. Thus, option (b) was considered as more appropriate.

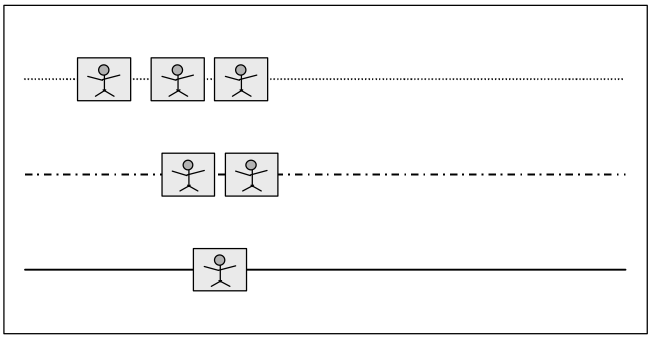


Figure 2: Fixed time-frame timelines window

An additional problem stemmed from the non-scrollable timelines approach (and was inherent in both of the above options): in the case that the video clips of different children concentrated on different time periods (e.g., child X shot 4 video clips between 9 a.m. and 10 a.m., while child Y shot 4 video clips between 1 p.m. and 2 p.m.) it was impossible for the timelines to accommodate both the large time range (from 9 a.m. to 2 p.m.) and the large concentration of thumbnails in little screen space without overlapping of the thumbnails (Figure 3). This problem is addressed by the Isis Story builder (Kim, 1995) through focusing on groups (*cliques*) of temporally close elements and letting the user work with only of them at a time – but this solution requires some kind of scrolling or switching between groups of thumbnails, which in this case was decided (following the project's requirements) not to be supported.

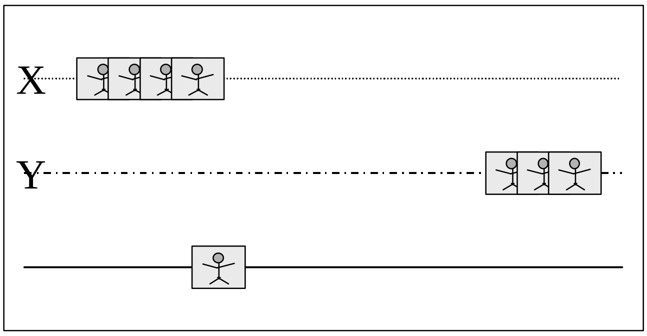


Figure 3: A problem with non-scrollable timelines

However, since the activity of video capturing was expected to be performed by children at scheduled (i.e., determined) points in time (e.g., during the break, or during a visit to a museum), it was expected that the various video clips will be, more or less, concentrated in close points in time. Thus, it was decided that the timelines should represent a variable time frame varying according to the period in which the separate video clips were shot.

To calculate this time frame the following formula was adopted:

- Start of the timelines = the quarter of an hour that is before (the time of the earliest video clip minus a quarter of an hour) – e.g., if the first video clip was shot at 8.35 a.m., then the start of the timelines would be the quarter of an hour that is before (8.35 – 15 = 8.20) which is 8.15 a.m.
- End of the timelines = the quarter of an hour that is after (the time of the latest video clip plus a quarter of an hour) – e.g., if the last video clip was shot at 1.35 p.m., then the end of the timelines would be the quarter of an hour that is after (1.35 + 15 = 1.50) which is 2.00 p.m.

An analogue time-scale is then adjusted on the timelines. The reason why quarters of an hour are used is that they can be conveniently represented. Furthermore, a quarter of an hour is subtracted and added to the start and end of timelines, respectively, in order to avoid having thumbnails at any one of the ends of the timelines.

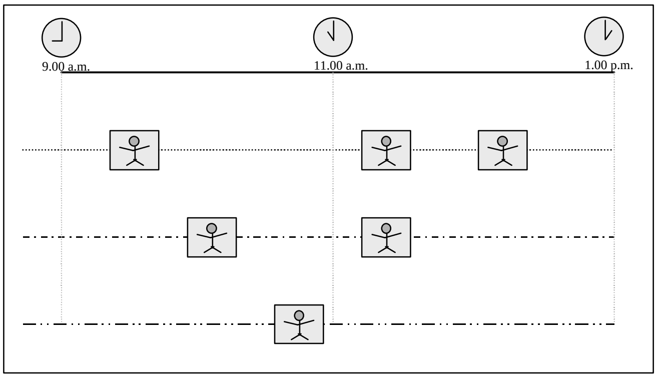


Figure 4: Representation of a time-frame

In order for the start and end time of the timelines to be easily understandable, a small clock (along with a time indication below it) is added over the start, medium and end of the timelines (Figure 4). Although this information might be redundant (or useless) for younger children that are not able to read or understand the time, it can be quite useful to older children (or to the teachers) in order to be aware of what the timelines represent, since this changes dynamically. Children that can not read have the option of hearing the time simply by touching on the clock. In any case, the actual usefulness of the clocks, as well as the appropriateness and timing of presentation for different age groups will be investigated during the evaluation of this preliminary design with children.

In addition to the above, the three timelines have different colours and patterns so that children can distinguish their own timeline. The same colours and patterns are used to frame the thumbnails, in order incorporate them on the timelines and provide a coherent image, but also make it possible to distinguish the 'author' of each video clip during the annotation phase. Furthermore, a small picture of the child's face (or another picture selected by the child) is presented at the beginning of each timeline (Figure 5).

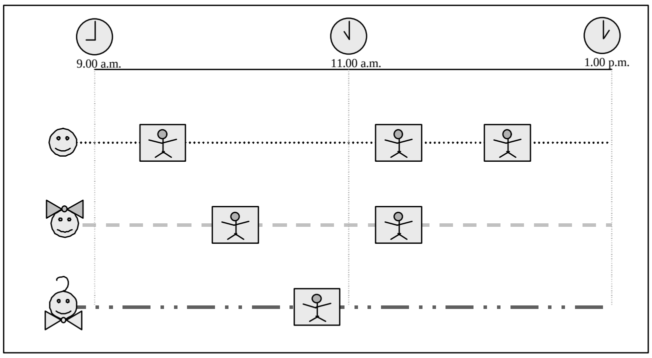


Figure 5: Distinguishing the timelines

2.3.3. Placement of the thumbnails on the timelines

An inherent problem of the thumbnails is that although they represent a single point in time (i.e., they correspond to a single pixel of a timeline), they have considerable size and span over several pixels. This means that if two consecutive thumbnails correspond to two close points in time, they will probably overlap. A way to minimise this problem is to minimise the size of the thumbnails. On the other hand, as the size of the thumbnails decreases, so does their clarity (it becomes more difficult to see their content), and their ease of use (it becomes more difficult to select them). Additionally, the size of a thumbnail can not be less than 64 x 48 pixels, in order for it to be large enough for being selected using a finger on a touch screen. Furthermore, the screen resolution for which the application is designed is 1024 x 768,and a part of the screen is devoted to a ‘toolbar’ (see below). As a consequence of these constraints, it is impossible to place more than 10 thumbnails on the same timeline.

Last, but not least, the fact that thumbnails belonging to different timelines but corresponding to the same episode (i.e., point in time) should be someway linked together imposes an additional constraint on the layout of the thumbnails, as well as to their size, since part of the screen should be devoted to visualise the linking.

Taking into consideration the above constraints, two different design approaches were proposed:

(a) a less general approach which has the advantage of being quite simple and easy to use and understand, and therefore more suitable for the younger children; and

(b) a more general approach, supporting the display of a larger number of thumbnails, but more complex, and therefore suitable for older children.

Following approach (a), some constraints arise concerning: (i) the number of the thumbnails; (ii) the time frame during which thumbnails were captured; and (iii) thumbnails’ distribution over time. Thus, (i) the maximum number of video clips that are transferred from the KidsCam to the Video Explorer is considered to be less than, or equal to, 4; (ii) all the video clips are captured in, more or less, the same (limited) time frame (e.g., from 9.15 a.m. to 10.45 a.m.); and (iii) all the video clips belonging to the same timeline are captured in intervals such that their thumbnails do not overlap when positioned on the timeline. These three constraints guarantee that the thumbnails can be placed on the timelines without incurring in the problems discussed in Section 2.2.2.

Two or more video clips referring to the same episode are surrounded by a coloured frame, with a semi-transparent background. Since there are only 3 timelines, the following 3 combinations can arise (Figure 6):

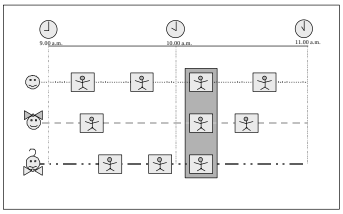
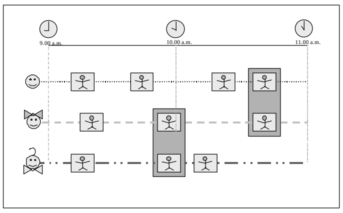
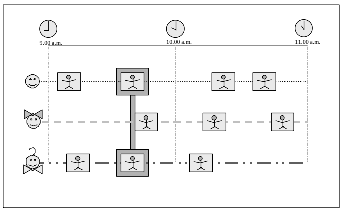
a. video clips from all the 3 timelines correspond to the same episode	b. video clips from adjacent timelines correspond to the same episode	c. video clips from the 1st and 3rd timelines correspond to the same episode
		

Figure 6: Alternatives for linking related video clips

As mentioned above, placing the thumbnails directly on the timeline might lead to overlapping between them. In the approach (b), the solution adopted to overcome this problem– is to place the thumbnails alternatively over and under the timeline, and link them to it through a line (Figure 7). If the minimum thumbnail size (64x48 pixels) is used, there is enough space for up to 20 thumbnails on a single timeline, a number that is possible to accommodate 'advanced' use of the system. The maximum thumbnail size that can be used is 128 x 96 pixels (this is constrained by the vertical size of the screen) and offers up to 10 thumbnails per timeline at an acceptable resolution.

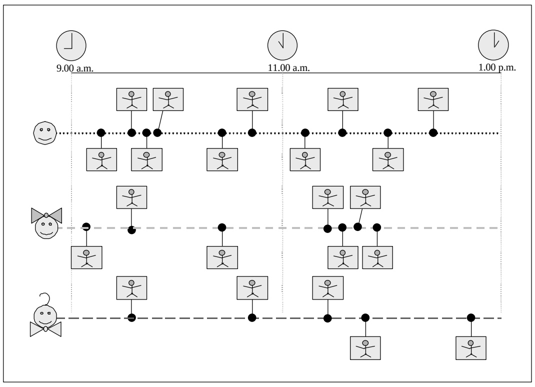


Figure 7: Linking thumbnails to timelines

When the minimum thumbnail size is used, a preview (or 'zoom') function is provided to help identifying thumbnails' contents: when one or more video clips are selected a larger version is presented (Figure 8). Then, if the child presses / clicks on the zoomed video clips, the Video Composer is called, while if the child presses / clicks anywhere else, he / she returns to the Video Explorer. A coloured circle with a semi-transparent background surrounds two or more video clips referring to the same episode. Since there are only 3 timelines, the combinations illustrated in Figure 9 can arise.

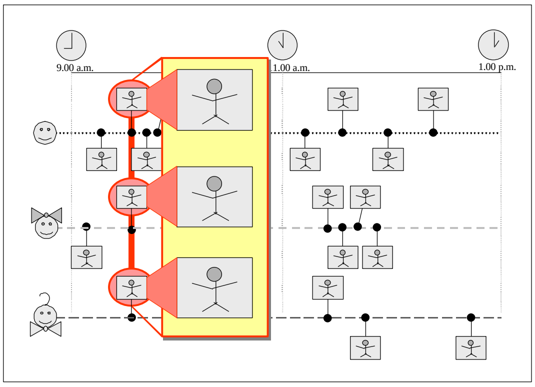


Figure 8: Zooming in the video clips

Due to the lack of literature and software relevant to the above tasks, this solution represents only a preliminary finding (based on expert knowledge in the HCI and educational software domains) on the usability and appropriateness for the target user groups. Thus, the optimal parameter values (e.g., maximum number and size of thumbnails per timeline) and adequacy of the solutions for different user groups, will be investigated and defined collaboratively with children.

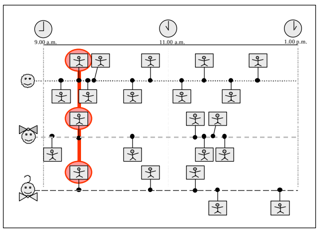
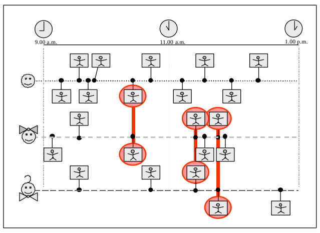
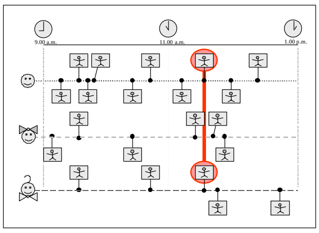
a. video clips from all the 3 timelines correspond to the same episode 	b. video clips from adjacent timelines correspond to the same episode 	c. video clips from the 1st and 3rd timelines correspond to the same episode 

Figure 9: Alternatives for linking related video clips

2.3.4. Additional functionality offered by the user interface

In addition to the above functions, children should be allowed to delete a video clip (and undo the deletion), send / give it to a friend, view a previous day's 'Today's Story' and get help. It is quite obvious that all these functions can not and should not be implemented directly on the timelines. Additionally, most of them are 'horizontal' in the sense that they are also used in other parts of the system (e.g., in the Video Composer). In a 'traditional' window-based user interface those functions would belong to a toolbar. Because of the overall style of the suggested user interface, instead of a typical toolbar, part of the screen is separated and 'populated' by interactive objects that provide the above functionality. These objects are:

Function	*Object*	*Description*
Get help	*character*	An animated cartoon character is ‘responsible’ for this task. If pressed, it moves around the screen presenting the available functions and explaining the interface. If dragged and dropped on a specific control / button it explains its function.
Delete / undelete a video clip	*trashcan*	• If a video clip is dragged and dropped on the trashcan, it is deleted. When the video clip is dragged over it, the trashcan’s lid opens and when the video clip is dropped in it a sound effect is heard and the lid closes. Furthermore, trashcan looks differently when it is empty and when it full (e.g., when full, it looks fatter than normal). • When pressed, the trashcan ‘spits back’ the last deleted video clip. Multiple undos are also supported.
Send / give a video clip to a friend	*box*	When a Memory Box[3] is near the Magic Mirror, an ‘open box’ icon appears. If a video clip is dragged and dropped on it, it is copied to the box and a small thumbnail appears on it. If the video clip in the box is dragged and dropped on the trashcan, it is deleted. When no Memory Box is near, an embossed icon of the Memory Box appears.
View previous ‘Today’s Story’	Tue 11 *calendar*	When pressed, a new screen opens presenting a calendar control, through which it is possible to select a previous day’s story. This icon has always the current date written on it. Probably this functionality will not be available to the younger children, since it requires some reading skills and knowledge related to the days, the months, etc.

2.3.5. Sketches and visualizations of the Video Explorer user interfaces

The above steps lead to the creation of the following sketch (Figure 10) of the Video Explorer user interface for approach (a):

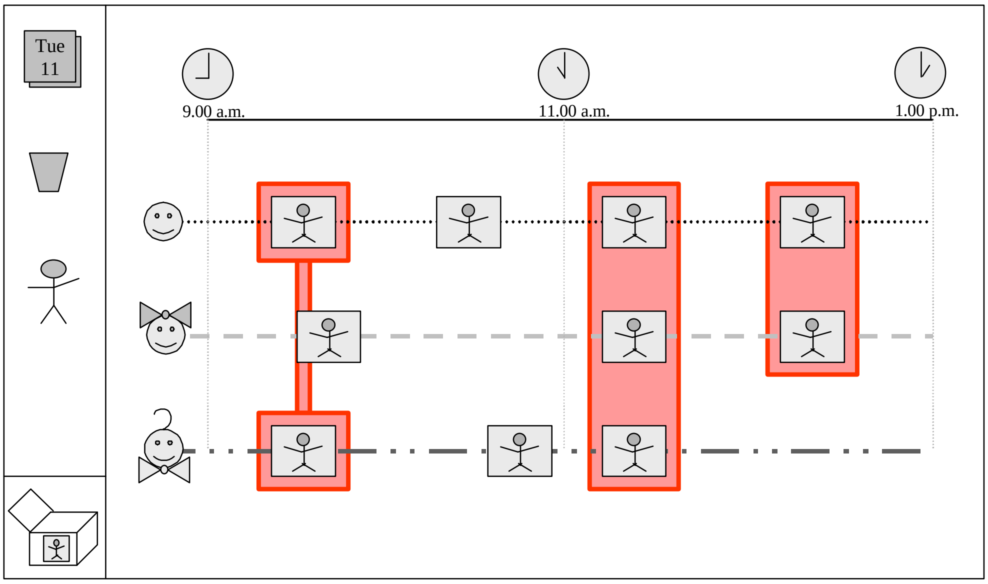


Figure 10: Sketch of the Video Explorer user interface – approach (a)

[3] A Memory Box is a special I/O device developed by the project. It is a wooden box equipped with infra-red that can store in it data and transmit its contents to a computer.

A sample visualisation of the Video Explorer user interface for approach (a) follows (*Figure 11*).

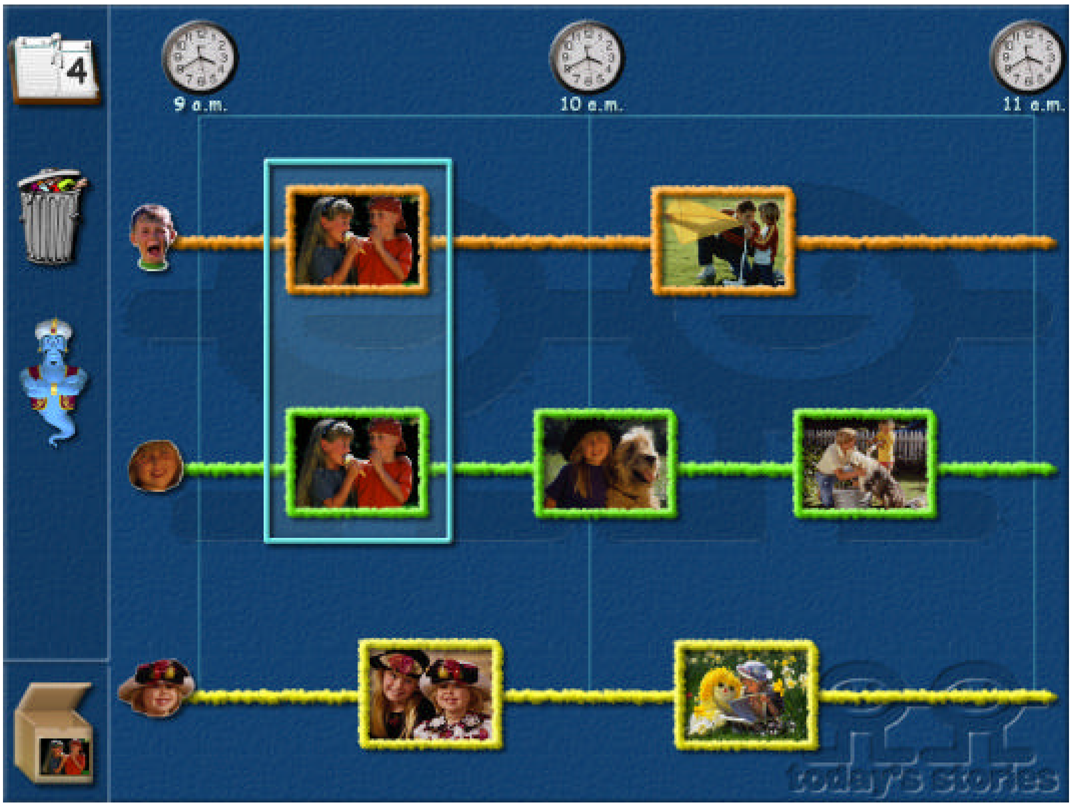


Figure 11: A sample visualisation of the Video Explorer user interface – approach (a)

The following (Figure 12) is a sketch of the Video Explorer user interface according to approach (b):

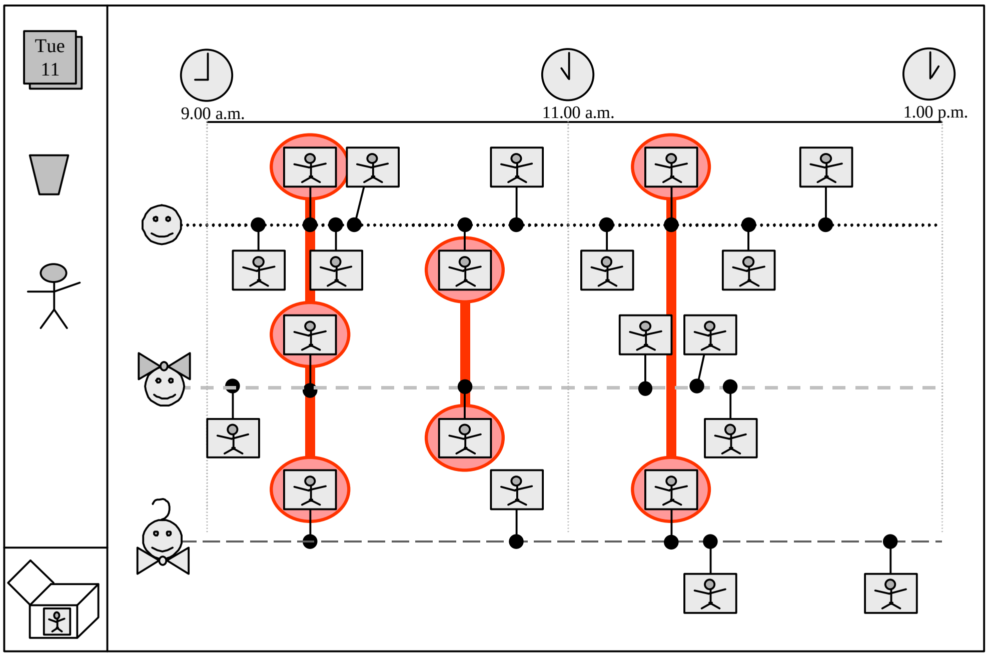


Figure 12: Sketch of the Video Explorer user interface – approach (b)

The above sketches and visualisations should not be considered as ‘final’ user interfaces. They are preliminary mock-ups that are used to illustrate some of the alternative design options. Their usefulness relies in the expressiveness of images compared to text, especially in order to: (a) communicate the design; and (b) constitute a common ground for the designers, the developers, and the other stakeholders (teachers, parents, children) for discussing and clarifying issues, and identifying problems or misunderstandings regarding the user interface and functionality of the future system.

2.4. The Video Composer

When a single thumbnail or a group of thumbnails is selected through the Video Explorer, the user interface switches to the ‘annotation mode’ (the Video Composer), in which children can view and annotate the selected video clips with pictures, sounds and voice commentary. This section provides a detailed analysis and the design specifications of the annotation user interface, and discuss intrinsic constrains, arising problems and their solutions.

2.4.1. Requirements and constraints

According to the project’s specification (Koutra et al., 2000), children should have access to functionality for video control similar to the one offered by popular video editing software (e.g., Play, Stop, Pause, Rewind, Fast Forward, Slow motion, Volume Control). Additionally they should be empowered to view the original video clip without annotations (i.e., be able to switch annotations on/off). In the case that more than one video clips are open, they should be treated as one, which means that there should be no separate controls for each of them. In this case, children should be able to 'close' (and later open) any of them, so that they can focus on just one or two video clips.

Support for three types of annotations was foreseen: image (i.e., symbols), sound (i.e., effects) and voice (i.e., spoken narrative). Image and sound annotations should probably be selected through some kind of ‘palette’. There should be no restrictions on the number of annotation symbols included in the final Diary Composer system. Annotation symbols could be grouped on separate palettes (e.g., emotions, actions, user-defined). This grouping should not be made explicit to the children and each palette should have a flat structure.

Image and sound annotations should be inserted (in a specific frame) by stopping (pausing) the video clip and then adding an annotation symbol on it. When the annotation procedure is over, the system should automatically add the annotation symbol in a predefined number of frames before and after the selected frame, ensuring that it is visible for a time period that is perceivable by the human eye / brain. Annotations should be automatically saved. Voice annotations should be treated in a different way, due to their nature. It should be possible to insert a voice annotation both while the video clip is playing and on a specific (paused) frame. In the first case the child should be able to add to the video clip a running commentary (in the same way that a sports caster describes a football game). In the second case, the video clip should be paused and a comment should be inserted at the specific point in time.

When more than one video clips are annotated concurrently, the new annotations should not replace the existing ones. Any previously existing annotations should be explicitly deleted / removed by the children.

Finally, children should also be provided with a function for removing annotation symbols, but there should be no function available for removing all the annotations simultaneously (i.e., ‘resetting’ the video clip).

In the following sections, the issues and design decisions concerning the design the user interface of the ‘Video Composer’ and stemming from the above requirements and constraints are discussed.

2.4.2. Supported video control functions and their representation

Concerning the video control functions to be offered in the Video Composer, two options were considered. The first option was to support all the functions available by common video editing programs (e.g., Play, Play Backwards, Stop, Pause, Rewind, Fast Forward, Slow Backward & Forward). The second option was to reduce the set of functions for the sake of simplicity and visual clarity, since the user interface is intended for use by young children. The minimum functions opted for to facilitate the annotation process are: Play (forward & backward), Stop (in the sense of ‘pause’), and Fast (forward and backward). When the system is to be used by very young children, these are reduced to just Play (forward & backward) and Stop.

Video functions are represented by conventional video controls, since (at least the older) children are quite familiar with them (see Figure 13). Another possible representation suitable for very young children makes use of graphics, such as a hare for fast forward, a turtle for slow forward, etc. In addition to the above, children can start and stop a video clip simply by touching / clicking on it. Once more, the aforementioned alternatives, as well as the actual level of video control needed to support the annotation task, will be investigated during the user trials.

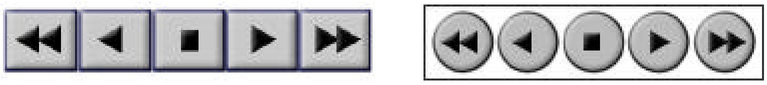


Figure 13: Examples of rectangural and round buttons for video control

When more than one video clips are open, only a single manifestation of the video buttons is visible (and not one for each video clip), since (according to the project requirements) they should all treated as one. The buttons are placed on a ‘control box’ (see Figure 14).

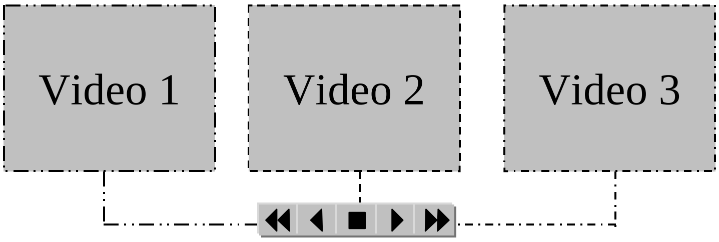


Figure 14: Example of video control box

Volume control is performed through the interaction object presented below (Figure 15):

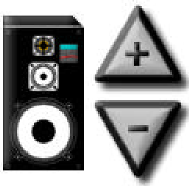


Figure 15: Volume control

If the ‘+’ button is pressed. the volume increases and a sample (short) tone is played. If the ‘-’ button is pressed the volume decreases and a sample tone is played. If the ‘speaker’ button is pressed, a sample sentence is played. Furthermore, in order to provide additional visual feedback, the size of the speaker increases / decreases when the ‘+’ and ‘-’ buttons are pressed.

When children wish to focus on just one or two of the video clips, they can close the other(s) through a special ‘handle’, attached on the side of each video clip. When this handle is pressed, the video clip shrinks to a thumbnail which also includes this special handle (Figure 16). The thumbnail is just a ‘still’ picture and thus is not affected by the use of the video controls. When the thumbnail or its ‘handle’ is pressed / clicked, it ‘grows’ back to its normal size and functionality. This function can be omitted when the system is to be used by younger children.

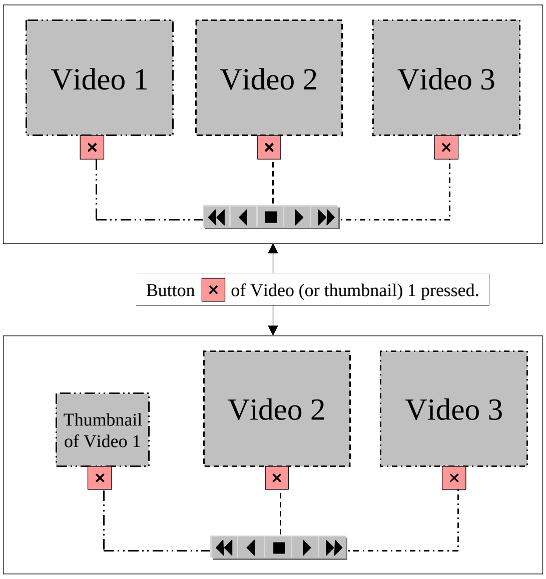


Figure 16: Closing and opening a video clip

2.4.3. Representation and functionality of the annotation symbols and palettes

The image palette has the ‘look and ‘feel’ of a collection of stickers (e.g., when an image is selected an ‘un-stick’ visual and sound effect is produced), since most of the children are quite familiar with stickers and can intuitively understand their purpose and functionality. When one of these pictures is selected a short description is heard. Two exemplary representations of the image palette are illustrated in Figure 17.

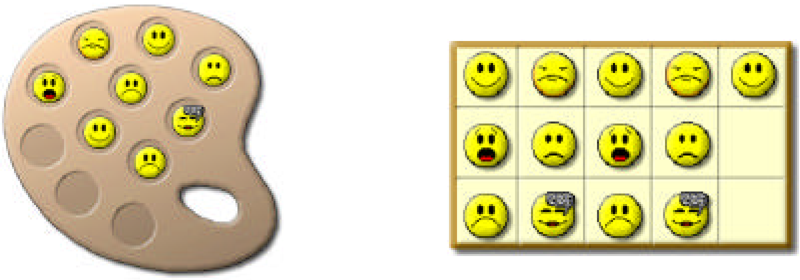


Figure 17: Examples of image annotation palettes

The sound palette looks significantly different than the image palette in order to avoid confusion. Actually, it looks like a CD that has pictures representing sound effects on it. These pictures have a different look than those of the image annotation palette; they all include a small CD icon (Figure 18). When one of these pictures is selected, the respective sound effect is played, along with a visual effect of the picture sticking to the mouse pointer.

Since image and sound annotations are not allowed while the video clip is playing, the image and sound palettes are (and look) inactive during that period and become activated as soon as the video clip is stopped.

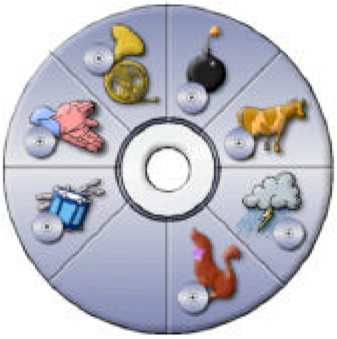


Figure 18: Example of a sound annotation palette

In addition to the above, an unlimited number of annotation symbols and multiple annotation palettes are supported through an easy and direct approach: a set of previous / next buttons reside close to each palette and enable the users to browse through the available palettes (see Figure 19). When one of these buttons is pressed, a new palette slides on top of the old one. Offering this functionality is important since "*the use of varieties of sign systems can enhance still further our understanding of the constructedness of knowledge, the value of considering multiple perspectives*" (Knuth & Cunningham, 1993). This functionality can be omitted when the system is to be used by younger children.

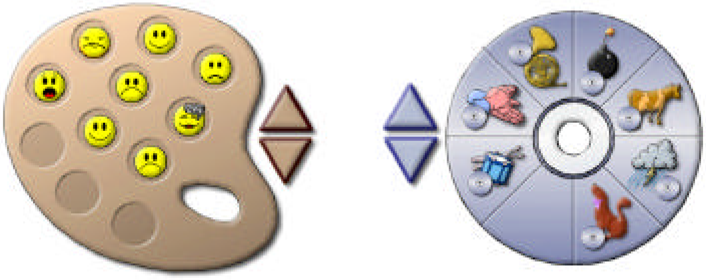


Figure 19: Examples of annotation palettes with previous / next buttons

The annotation palettes can be easily customized by the children, who can fill them with their own items (images and sounds) by using a Memory Box. If the content of a Memory Box is an annotation item, it can be dragged and dropped on an empty position of a (relevant) annotation palette, so as to be added on it. The palettes include empty positions by default, and additionally the children can create more empty positions by deleting existing items. Children can remove palette items, by dragging and dropping them to the trashcan. Undelete is supported through pressing / clicking the trashcan.

The final 'look and feel' of the palettes, as well as the annotation symbols that are presented on them, will be shaped collaboratively with children, to appeal to their individual preferences but also to their cultural background. Some suggestions for annotation symbols can be found in (Dreyfuss, 1984; Frutiger, Bluhm, 1998; Horton, 1994; Rudolf et al., 1977).

2.4.4. Video clip annotation

To add an image or sound annotation, first the video clip has to be stopped. Then:

a) To add an image annotation the child drags an image from the annotation palette and drops it anywhere on one of the video clips. The child can then move the annotation symbol and change its position on the video clip.

b) To remove an image annotation the child drags it from the video clip it resides upon and drops it anywhere outside the video clip. In case it drops it on another video clip, the annotation is transferred to it.

Sound annotations work in a similar way, with the difference that when the annotation symbol is pressed, the sound effect represented by the symbol is played instead of an 'un-stick' sound effect.

Speech annotations are treated differently. Firstly, a microphone control is used instead of a palette. The microphone control is animated (small 'beams' of sound appear) when active, to provide relevant feedback. Secondly, there are two alternative ways for inserting voice annotations:

a) 'Running' voice annotation (i.e., add to the movie a running commentary, in the same way that a sports caster describes a football game). The annotation starts and ends by pressing the microphone button. When the video clip is played back, the annotation is played along with it, substituting its original sound. This type of annotation is represented on the video clip by means of a microphone symbol.

b) 'Still' voice annotation: First, the video clip has to be stopped. Then, the microphone button is pressed. The annotation ends either by pressing the microphone button again or by starting the video clip. When the video clip is played back, it (automatically) stops at the annotated point, the comment is played (along with some feedback that the specific annotation is played – e.g., the annotation symbol flashes) and then the video clip is resumed. This type of annotation is represented on the video clip by means of a microphone symbol with a small stop button on it.

It is expected that in the final version of the user interface only one of the above approaches will be supported (to avoid confusion), but the decision of which will prevail will be made after testing of both approaches with actual users.

The presentation of annotations can be switched on and off, through the following two-state button (Figure 20):

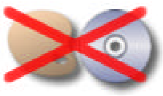
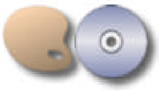
State 1 	When pressed, annotation palettes become inactive (and / or hidden) and video annotations are switched off. The button changes to *State 2*.	*State 2* 	When pressed, annotation palettes become active (and visible) and video annotations are switched on. The button changes to *State 1*.

Figure 20: Annotation switching button

2.4.5. Additional issues and functionality

A toolbar, which is quite similar to that of the Video Explorer, resides on the one side of the user interface. The only difference is that the calendar control is replaced by a control (looking like the 'back' button in Figure 21 or like a miniature of the timelines) for returning to the Video Explorer.

Figure 21: Sample 'back' button

2.4.6. Sketches and visualisations of the Video Composer user interface

The above analysis and discussion lead to the creation of the following (first) sketch (Figure 22) of the Video Composer user interface:

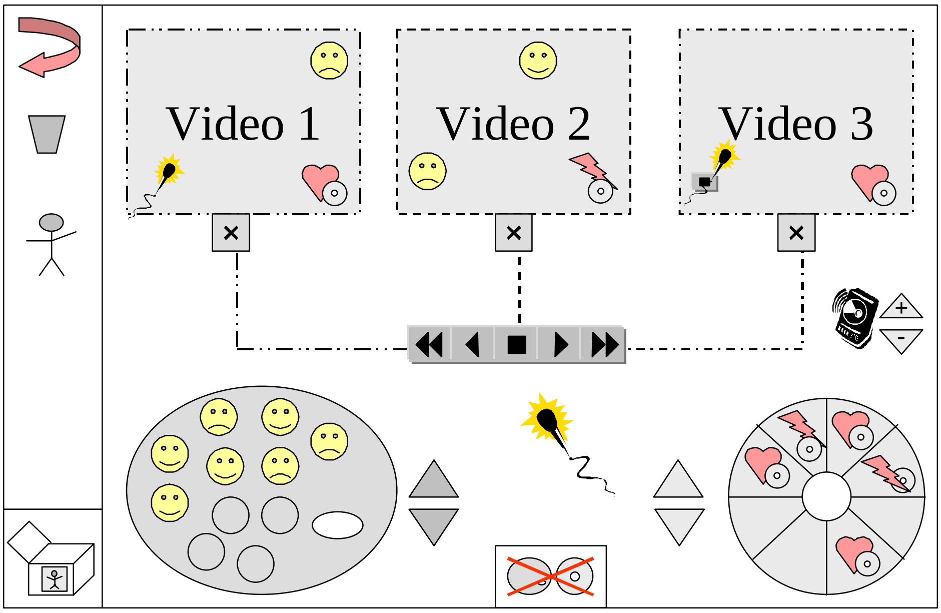


Figure 22: A first sketch of the Video Composer user interface

An expert usability evaluation (Nielsen, 1994) of the above user interface sketch revealed two possible drawbacks:

(i) the annotation palettes were too far from the video clips;

(ii) the video control and the open / close buttons were 'mixed' with the palettes.

These drawbacks were solved by reversing the position of the video clips and their controls (see Figure 23 below). As a result, the video clips provided a physical barrier that separated (and grouped) the video control buttons from the annotation functions.

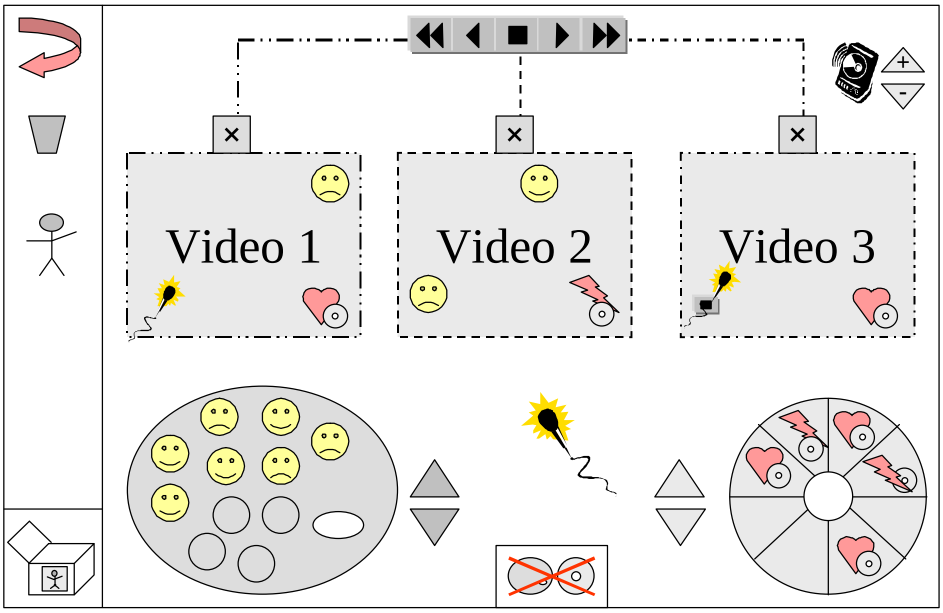


Figure 23: A sketch of the revised Video Composer user interface

The following (*Figure 24*) is a sample visualisation of the revised Video Composer user interface:

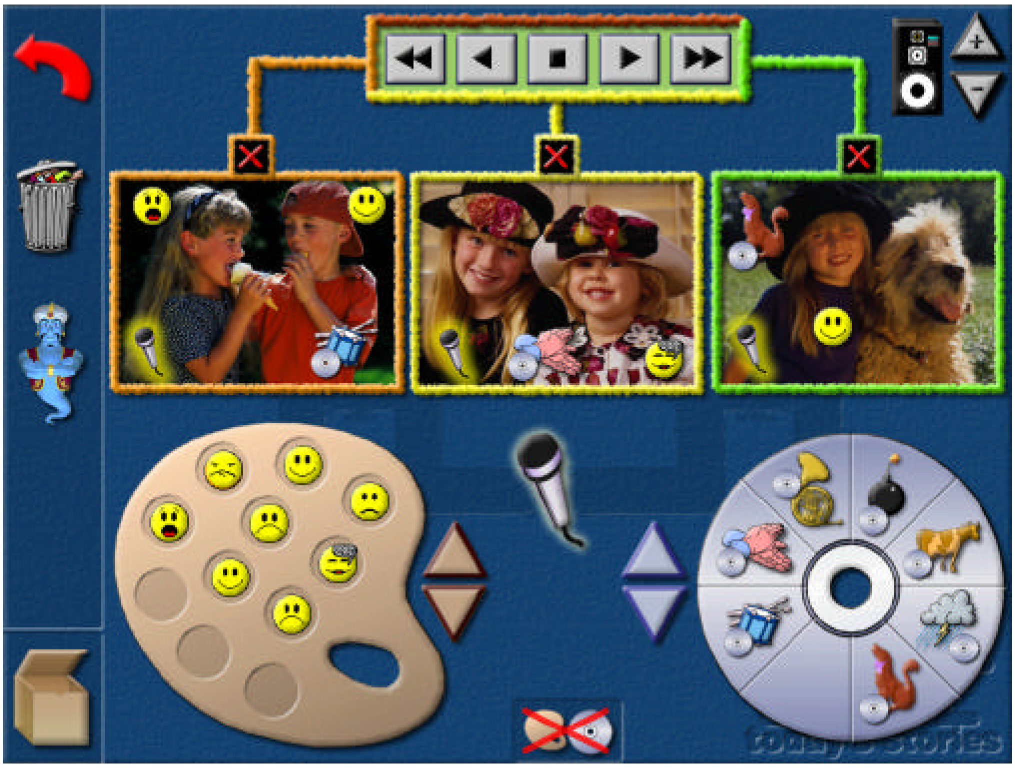


Figure 24: A sample visualisation of the revised Video Composer user interface

Since the Video Composer provides some complex functionality and might overwhelm or perplex very young children during their first contacts with the system, a simpler version of the user interface, containing only the 'bare necessities' for the video annotation task, has also been designed (Figure 25).

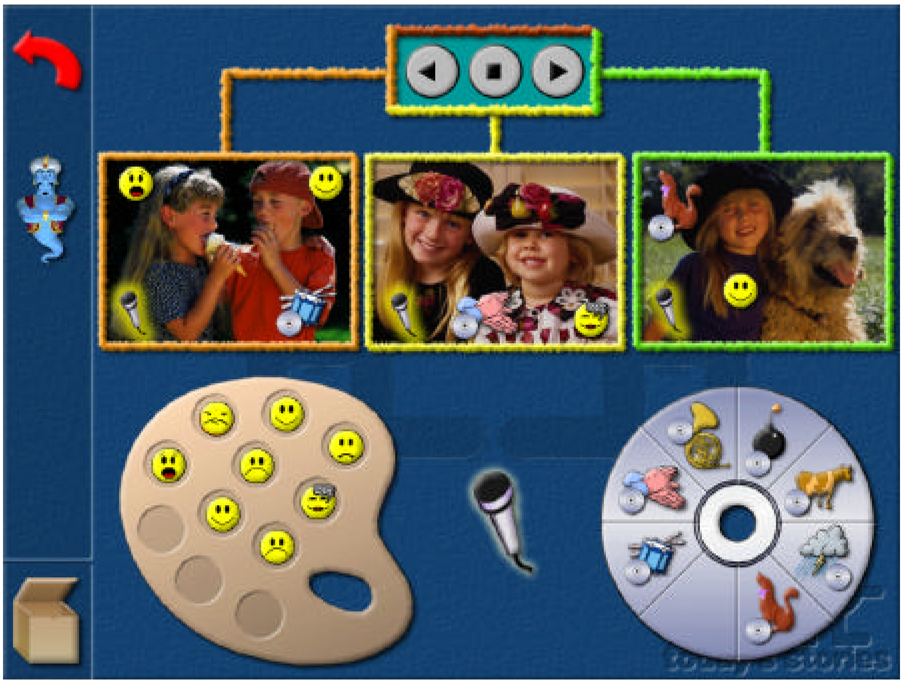


Figure 25: Alternative version of the Video Composer user interface for very young children

3. DISCUSSION, CONCLUSIONS AND FUTURE WORK

This paper reported on the preliminary design of the user interface of the Diary Composer software of the Today's Stories project. Since the user interface is intended for use by very young children (4-8 years old) the design relies highly upon visual rather than textual interaction. The resulting environment is intended to be aesthetically pleasing and appealing to children, while, at the same time, it aims to support simple, intuitive interaction through the adoption of adequate interaction techniques, the use of suitable metaphors to represent system functions, and the continuous provision of visual and audio feedback. Furthermore, the final system should be customised to the needs, requirements and preferences of the children, promoting self-expression and individuality.

The outcome of the work reported in this paper will be used to create prototypes which, following the principles of User-Centred Design (Norman and Draper, 1986), will be tested with, and evaluated by, actual users (i.e., children) (Hanna et al., 1997). These activities will eventually lead to refinements and elaboration of the design and overall improvement of the user interface, as well as to comparisons between design alternatives. Furthermore, they will help to ensure that the system behaves as it is expected to, and to assess whether the requirements and goals set during the requirements phase have been met. Some of the issues revealed through this design activity, and require further investigation and experimentation with actual users (i.e., children), include:

- the representation of time on the timelines;

- the maximum number, minimum size and the positioning of the thumbnails on the timelines;
- the level of functionality needed by, and adequate for, children of different ages;
- the customisation of the visual and sound components of the user interface for children of different age and/or with different cultural backgrounds (Galdo & Nielsen, 1996; Fernandes, 1995; Nielsen, 1990);
- the type and representation of annotation symbols that will be included on the annotation palettes.

ACKNOWLEDGEMENTS

The work reported in this paper has been carried out by the Institute of Computer Science of the Foundation for Research and Technology – Hellas under a subcontract commissioned by the Lambrakis Research Foundation, in the context of the project "Today's Stories" P29312 funded by the European Commission in the framework of the Intelligent Information Interfaces (i^3), Experimental School Environment Programme. The "Today's Stories" consortium comprises: Starlab (Belgium), Gøteborg University (Sweden), Maersk Mc-Kinney Moller Institute for Production Technology (Denmark), Nr. Brody Skole (Denmark), Computer Technology Institute (Greece), Lambrakis Research Foundation (Greece), The Center for Futurism in Education (Israel).

The authors wish to acknowledge the support and encouragement received from Dr. George Neofotistos, Lambrakis Research Foundation.

REFERENCES

Baecker, R., & Small, I. (1990). Animation at the Interface. In B. Laurel (Ed.), *The Art of Human-Computer Interface Design* (pp. 113-122). Reading, MA: Addison-Wesley Publishing Company, Inc.

Cooper, A. (1995). *UI Design About Face: The Essentials of user Interface Design*. Foster City, CA: IDG Books Worldwide

Dreyfuss H. (1984). *Symbol Sourcebook: An Authoritative Guide to International Graphic Symbols*. New York: John Wiley & Sons

Druin, A. (1999a). Cooperative Inquiry: Developing New Technologies for Children with Children. In *Proceedings of the Conference on Human factors in computing systems: the CHI is the limit (CHI '99),* Pittsburgh, PA, USA (pp. 592-599). New York: ACM Press.

Druin, A. (Ed.). (1999b). *The Design of Children's Technology*. San Francisco: Morgan Kaufmann Publishers, Inc.

Druin, A. & Solomon, C. (1996). *Designing Multimedia Environments for Children*. New York: John Wiley & Sons, Inc.

Erickson, T.D. (1990). Working with Interface Metaphors. In B. Laurel (Ed.), *The Art of Human-Computer Interface Design* (pp. 65-73). Reading, MA: Addison-Wesley Publishing Company, Inc., Fernandes, T. (1995). *Global Interface Design: A Guide to Designing International User Interfaces*. Boston: AP Professional.

Frutiger, A. & Bluhm, A. (1998). *Signs and Symbols: Their Design and Meaning*. New York: Watson-Guptill Publications

Furger, R. (1998). *Does Jane Compute? : Preserving Our Daughters' Place in the Cyber Revolution*. New York: Warner Books.

Galdo, M. & Nielsen, J. (1996). *International User Interfaces*. New York: John Wiley & Sons.

Galitz, O.W. (1997). *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and techniques*. New York: John Wiley & Sons

Hanna, L., Risden, K., & Alexander, K. (1997). Guidelines for Usability Testing with Children. *Interactions Magazine*, issue September + October 1997, 9-14.

Horton, W.K. (1994). *The Icon Book: Visual Symbols for Computer Systems and Documentation*. New York: John Wiley & Sons.

Howlett, V. (1996). *Visual Interface Design for Windows*. New York: John Wiley & Sons.

Kim, M.Y. (1995). Creative Multimedia for Children: Isis Story Builder. In *Companion Proceedings of the Conference on Human Factors and Computing (CHI '95)*, Denver, CO, USA (pp.37-38) New York: ACM Press.

Knuth, R.A., & Cunningham, D. J. (1993). Tools for Constructivism. In T.M. Duffy, J. Lowyck, & D.H. Jonassen (1993). *Designing Environments for Constructive Learning* (vol. 105, pp. 163-188). Berlin: Springer-Verlag, NATO ASI Series, Series F: Computer and Systems sciences.

Jonassen, D., Mayes, T., & McAleese, R. (1993). A Manifesto for a Constructivist Approach to Uses of Technology in Higher Education. In T.M. Duffy, J. Lowyck, & D.H. Jonassen (1993), *Designing Environments for Constructive Learning* (vol. 105, pp. 231-247). Berlin: Springer-Verlag, NATO ASI Series, Series F: Computer and Systems sciences.

Koutra, C., Kastis N., & Neofotistos, G. (2000). Interactive Learning Environments for Children: User Interface Requirements for a *Magic Mirror* and *Diary Composer* Environment, submitted.

Mountford, S.J., & Gaver, W.W. (1990). Talking and Listening to Computers. In B. Laurel (Ed.), *The Art of Human-Computer Interface Design* (pp. 113-122). Reading, MA: Addison-Wesley Publishing Company, Inc.

Nicol, A. (1990). Interfaces for Learning: What Do Good Teachers Know That We Don't?. In B. Laurel (Ed.), *The Art of Human-Computer Interface Design* (pp. 113-122). Reading, MA: Addison-Wesley Publishing Company, Inc.

Nielsen, J. (1990). *Designing User Interfaces for International Use*. Amsterdam: North-Holland, Elsevier Science Publishers.

Nielsen J. (1994). *Usability Inspection Methods*. New York: John Wiley & Sons.

Norman, D. (1988). The Psychology of Everyday Things. New York: Basic Books.

Norman, D. (1990). Why Interfaces Don't Work. In B. Laurel (Ed.), *The Art of Human-Computer Interface Design* (pp. 209-219). Reading, MA: Addison-Wesley Publishing Company, Inc.

Norman, D. & Draper, P. (1986). *User-centred Design: New perspective in Human Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Rudolf and Modley, Myers, W.R., & Comer, D.G., (1977). *Handbook of Pictorial Symbols: 3250 Examples from International Sources*. New York: Dover Publications.

Shneiderman, B., (1998). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, MA: Addison-Wesley Publishing Company.

Temple University. (1999). *Guidelines for the Design of Educational Software*. [On-line]. Available: http://www.temple.edu/dentistry/di/edswstd/title.htm

Today's Stories Consortium. (1999). *Deliverable D 3.3.1: Diary Composer Requirements and Concept Prototypes*.

Tognazzini, B. (1996). *TOG on Interface*. Reading, MA: Addison-Wesley Publishing Company.

Weinschenk, S., Jamar, P., & Yeo, C.S. (1997). *GUI Design Essentials*. New York: John Wiley & Sons.

Wilson, K.S. (1988). Palenque: An Interactive Multimedia Digital Video Interactive Prototype for Children In the *Proceedings of Conference on Human Factors and Computing Systems (CHI '88)*, Washington, USA (pp. 275-279). New York: ACM Press.

Winn, W. (1993). A Constructivist Critique of the Assumptions of Instructional Design. In T.M. Duffy, J. Lowyck, & D.H. Jonassen (1993). *Designing Environments for Constructive Learning* (vol. 105, pp. 189-212). Berlin: Springer-Verlag, NATO ASI Series, Series F: Computer and Systems sciences.

Wood, E.L. (1998). *User Interface Design: Bridging the Gap from User Requirements to Design*. Boca Raton, FL: CRC Press