User Interface Design for Keyframe-Based Content Browsing of Digital Video

DUBLIN CITY UNIVERSITY
Ollscoil Chathair Bhaile Átha Cliath

by

Hyowon Lee

A Dissertation Presented in Fulfilment of the Requirements for the Ph.D. Degree

Supervisor: Prof. Alan F. Smeaton

School of Computer Applications

January 2001

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy in Computer Applications, is entirely my own work and has not been taken from the work of others.

Signed: ____________________ Date: __________.

Hyowon Lee

Acknowledgements

My PhD research has been the largest part of my past few years. During this time, I have been reading, pondering and writing. But it was not just my own work – it involved many people directly and indirectly, and I'd like to show my appreciation to them.

My supervisor, Prof. Alan F. Smeaton, has been most supportive and I cannot thank him enough for his consideration since I came to DCU. I am very lucky to have him as my supervisor, his guidance and understanding has been the most crucial part throughout my study.

I thank Prof. Catherine Berrut for her research support in detail. She helped me focus the direction of my research very much, and I could not have done this much of progress without her help.

I thank Drs. Noel Murphy, Seán Marlow and Noel O'Connor, the principal investigators in the Centre for Digital Video Processing, for their kindly and useful advice, paper corrections and opinions in our numerous meetings.

I thank Dr. Jonathan Furner in UCLA for supporting me during the first year of this research. His introducing me to the video retrieval field led me into studying in depth this exciting area.

I also thank Kieran Mc Donald, for his strong support during my research. He has been giving me million dollar comments and many of them are reflected in my thesis. I very much appreciate the frequent discussions he initiated, ideas he suggested when reviewing books and papers, I do not see any better research environment than what I have had with him. I thank Paul Browne, for his strong support since he came to our Lab. His everyday feedback and much workload he took charge for me is much appreciated. I am really lucky to have these two guys as my work colleagues.

I thank Cathal, Tom, Aidan, Criostaí and Evan – the gang of supportive and friendly colleagues in my bay. I'd also like to give my appreciation to Jer, for his friendly support. I learned a lot of things from the discussions we've had, often during the early morning coffee time, which became one of my favourite time of the day. Also, other postgraduate students current and past have been all very supportive and nice to me – I thank them all.

I also thank the people who participated in the evaluation stage of this research. Their interesting and useful comments made this thesis much better.

Finally I thank my home folks, though nearly on the other side of the globe, who have been my constant source of energy throughout the journey. Studying abroad is hard, but I am glad to have my family who made it less hard. I thank Marion in Toronto and my aunt in California for their support.

Hyowon Lee

January 2001

Abstract

With the technological advancements in computing and networking today, many interesting and potentially useful applications are being developed and turned into products. Digital video is one of those application areas that is getting more and more attention as the consumer market is likely to be huge, including professional video editing and home entertainment. However, most digital video systems so far have been developed based on technologically based ideas and possibilities, thus providing very little usability and poorly designed user interfaces. Having no long-term user base or usage context make it difficult to develop a new system using a user-oriented methodology where a target usage specification is made and a system developed accordingly.

In this research, the design of user interface to a digital video system is studied in depth, particularly focusing on keyframe-based video content browsing interfaces. By analysing and identifying important elements in designing keyframe-based browsing interfaces, we construct a well-defined "design space" where a specific user interface can be designed by simply selecting a set of available options. This analytic approach makes the user interface design, which can be difficult to clarify due to its fuzzy nature, much more systematic and manageable. The usefulness of the constructed design space is demonstrated in the context of diverse devices such as desktop PCs and PDAs (Personal Digital Assistants) sharing the same data, and also diverse characteristics of the users. Specifically, video browsers suitable for a PDA are designed, and also various video browsers supporting different user preferences are designed and implemented using the design space. A qualitative evaluation is conducted on the designed browsing interfaces, to improve their usability and to gain further insights into the design of such browsers and refine the underlying design space.

Table of Contents

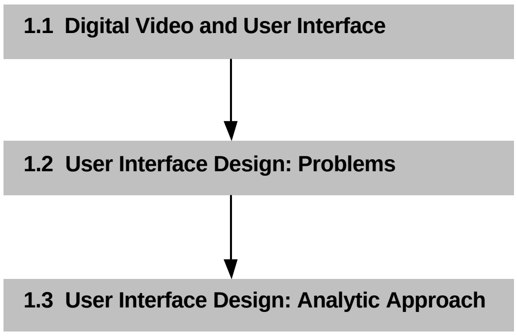
Table of Figures

Table of Tables

CHAPTER 1

Introduction

One of the greatest of the current trends in digital video, multimedia, and computer products in general is their diversity - diversified platforms, diversified devices through data sharing, as well as diverse user needs and preferences. All these contribute to making the design of the user interfaces to these systems complicated and potentially expensive. A design approach called analytic design can address these problems. The subject of this thesis is to construct a design space and evaluate its soundness, in the particular area of browsing of digital video content.



1.1 Digital Video and the User Interface

Digital video is one of the areas where technology is opening up huge possibilities for future usage of video content - video clip databases for broadcast companies, video editing systems for film producers and various home video entertainment systems such as DVD, Web TV, Video on Demand are some of the applications coming up in the near future. Commercial companies' interest in digital video is great because the potential market in the exploitation of video is huge. In domestic environment, for example, most households own at least one conventional TV and VCR and use them in their everyday lives.

Already starting to appear on the market are set top boxes such as TiVo [TiVo] and ReplayTV [ReplayTV], which record broadcast TV programmes in a digital format on their internal disks. These products are currently the digital version of the conventional VCR and get rid of the problems inherent in conventional systems, such as degrading video quality and managing a number of video tapes. DVDs are spreading very quickly, with their high quality video contents and attractive extra features such as director comments, multiple subtitles, changeable camera angle, and direct access to various pre-indexed scenes. Video on the web is also becoming a reality, though is still at the infancy stage of its development, using video streaming technology which allows playback of digital video files on the client while receiving the file data. With upcoming higher bandwidth networks, this will become much more reliable and faster as if playing from local data storage.

However, as with any other field that is driven largely by technological development, it is often the consideration on the human side of the picture that is neglected or overlooked when a new digital video product or service is developed. The *user interface to digital video* - often becoming overwhelmingly complex with all its multimedia elements and control widgets on the computer screen or TV - is surely a good example at the moment of where the technological possibility dictates the shaping of the end product or service, with the human-computer interaction and usability side often reflecting the engineering and feature-driven approach with some minor cosmetic work done at the end of the development. While all sorts of new technologies are implemented and thrown out to the market while yet further versions of them are being developed, it is often the end users and customers who suffer from the poor usability of such technology-driven systems. Products which are freshly launched with technological promises are usually poor in considering their users, because being first to market normally means no time for proper user studies in the development of that product. Conventional user interface development process where one starts from a task specification and description becomes difficult because no such system environment with established user group to consult exists. The result is often a product with very surprising functionalities but with a poorly crafted user interface. People buy products expecting to use all the functionalities incorporated in them, but the difficulty in learning how to use or manipulate them can make them a waste of time and effort, cause frustration and eventually stop users from using it. Forgotten functionalities due to their difficult usability are often found in many cases. Many families who own a home VCR do not use or do not bother using the automatic recording function by pre-setting the recording, not because it is not a useful functionality, but simply because it is difficult to pre-set. This is a typical case of a useful functionality becoming unusable because of its poorly designed user interface. Computer systems can be another case of a poorly designed interface causing frustrations. With so many complex functionalities incorporated into a single machine, with the screen full of icons, buttons and other "administrative debris" [Tufte 90, p89] all so easily causing user-driven errors and harsh error messages, it is no wonder we see such phrases as "computer phobia." For a technology product based on something like digital video, if such a system is to be accepted and used smoothly in our everyday lives, the user interface of the product has to be carefully considered.

1.2 User Interface Design: Problems

Both in companies and in academic circles, user interface design is still an area not yet given the proper consideration which it deserves. When conducting research in the area of user interface design, it is often strongly felt that there is a lack of a body of knowledge where such research can be started. A recent shift of direction in interface design has been from considering the design stage as a final, wrapping up process where some cosmetic work is done to attract users, into a more integrated approach where the requirements of a target user group are clearly captured at the beginning of a development, and then the system is developed according to these requirements specifications, constantly checked throughout the development of the system. This integrated, user-centred approach has been very much emphasised in recent years with its reward visibly experienced in companies in the form of system success in the market. A new job position called "usability engineer" has appeared, a profession within system development in charge of making sure that target user group's requirements are met in the final product by way of various newly proposed usability engineering methods and tools. Also many new independent consulting companies have been formed that specialise in the whole range of usability concerns for other system development companies including initial requirements capturing, integration of the requirements in the subsequent development stages, prototyping a system and conducting various usability evaluation methods. Many large software companies such as Microsoft, IBM and Sun Microsystems now have separate departments exclusively dealing with usability and user consultation. Emphasised is the development process reflecting the target users' requirements and wishes, and thus "iterative refinement" becomes an important part of the development process. Iterative refinement recognises that the initial design of a system cannot be perfect, and thus refines it by exposing it to target users and getting feedback, this cycle repeating until a satisfactory design comes out at the end. This is illustrated in Figure 1.1 below.

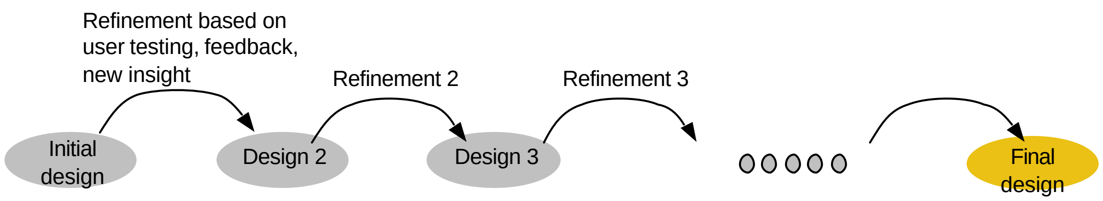


Figure 1.1 – Iterative refinement of the system's user interface

Although this shift of emphasis on integrating the "user" side into system development is a desirable movement, currently practised usability methods with iterative refinement are still largely based on trial-and-error having no theoretical backup or basis to think about user interface design. The *starting point for design is too arbitrary* and as a result the subsequent iterative refinement stages become laborious and expensive. This will be less of a problem if the target environment has been well established and understood, as more conventional user interface development process can capture in fine detail the initial user requirements and usage context and the initial design can start based on it. However, if designing an interface to a system whose environment is still not known and focusing on possible future development such as this thesis's theme digital video, the starting point can be more difficult to get right initially. This results in an unpredictable refinement process, as illustrated in Figure 1.2 below. What should be the initial design to start refinement from? How can we reduce the refinement stages by starting from the nearest possible final design from the beginning? While iterative refinement through usability testing involving real users is one important element of user-centred system development, the need for having proper theoretical and methodological grounds to provide a proper starting point for design is being felt more and more.

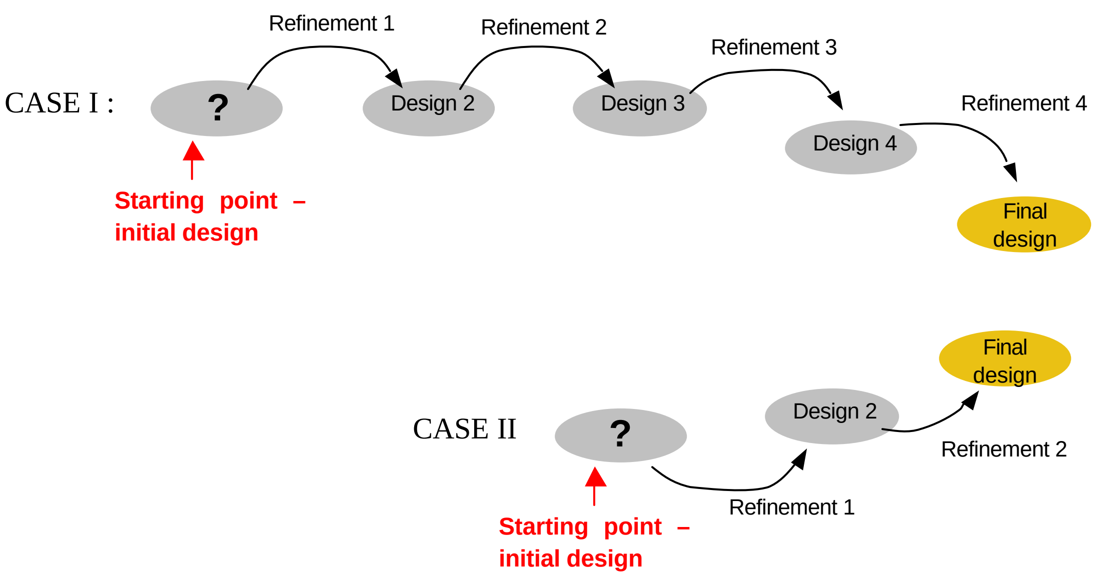


Figure 1.2 – Starting point determining the subsequent refinement work and final design: case I took more refinement stages to reach the similar final design result than in case II

Researchers and practitioners developing computer systems for end-users will generally agree that designing a good initial user interface is a highly creative process [Shneiderman 98, p89] and requires intuition and an almost artistic sense from the designer. This means that we should not stifle this aspect of design by emphasising a rigid theoretical and practical framework. Also it is true that designers get ideas from other existing interfaces and do design similar elements in their own creations, which is a kind of mimicry but can be viewed as constructive [Sanderson 97]. At the same time, with so many possibilities and uncertainties - mainly coming from the diversity of needs and contexts which we come to acknowledge these days - it is a daunting task when we try to design a "good" user interface, to go about comparing this design to another, and to improve it.

An interface designer will usually have some preferred conception or rough sketch on what the interaction and screen design should be like, partly based on past design experiences or what s/he has seen designed for other systems, and also partly based on some original creative ideas. It is true that designers tend to include design features that they have known from experience before, even when those features made the interface less usable [Smith *et al.* 00, p114]. When it comes to interfaces to digital video systems, what comes into mind will probably be a set of small pictures displayed on the screen, with some control widgets and a timeline bar at the bottom, and clicking on the picture starts playback of the video sequence from that picture's depicted moment.

One of the main problems in user interface design is its ambiguous notion of "good" and "bad" design. How do we know a good interface? How do we compare one interface to another? How can we come up with a new interface? The situation gets worse when the interface we are developing is a highly interactive one with multimedia objects presented and handled, providing infinite possibilities and infinite ramifications for implementation.

Platform diversification is another factor which is emerging strongly in technology and adds to the problem of user interface design. As computer hardware quickly develops and people adopt it into their work, a system that shares its data among different devices such as a desktop PC, a computer kiosk, a PDA (Personal Digital Assistant) and a mobile phone have recently appeared. As more and more computational power and network bandwidth are becoming available for these devices, this allows developers to be able to provide powerful applications with various potentially useful functionalities to run, and providing appropriate user interfaces to different devices is becoming an important issue. We cannot simply re-use the same interface with the same interaction styles as used on a desktop PC with a large screen with a mouse and keyboard, for a small, handheld PDA or a mobile phone. Different devices require different interaction styles and paradigms, requiring extra effort from the designers.

Furthermore, adopting a user-centred development to capture a target user group's requirements, developing a specific interface reflecting these requirements, and developing various interfaces suitable for different kinds of devices and then iteratively refining them by getting users to test use them - all this effort still cannot guarantee that all users will be satisfied with the final interface. Even when developing the same application for the same device, we come to realise that we cannot simply provide one refined and "optimised" user interface, hoping that everybody will be satisfied with this particular interface. People have different degrees of knowledge and preferences, and these personal differences make a big difference in their satisfaction with using a system.

1.3 User Interface Design: An Analytic Approach

To address these problems with user interface design mentioned above, there have been attempts and efforts to streamline and turn the fuzzy, unpredictable and ill-defined interface design concept into a more structured and formalised process, exemplified by "design space analysis" [MacLean *et al.* 89] and further adaptations and variations derived from this (for example, [Stary 00] [Graham *et al.* 00] [Calvary *et al.* 00]). This family of approaches to structured design have different emphases on procedural aspects and methods, but their underlying idea is following the following steps:

1. analyse and identify important facets or dimensions that are essential in designing the user interface under concern;
2. identify a list of possible design options or values for each of the dimensions identified above, forming a large but clearly defined *design space*;
3. identify particular usage contexts that work as constraints for selecting the right combinations of design, then
4. apply the identified set of constraints to the design space, thus coming up with a specific user interface.

This way, designing an interface becomes less of a fuzzy, artistic work and more of a concrete, simple decision-making process where a well-defined "design space" provides a rich but clear set of possible options and selecting them from each dimension constructs one specific interface. Selecting the "right" options from all the possible set of options becomes possible by applying some concrete usage context to filter out or limit possible combinations. Once a specific interface is selected from the design space, iterative refinement can follow against that interface. This is illustrated in Figure 1.3 below:

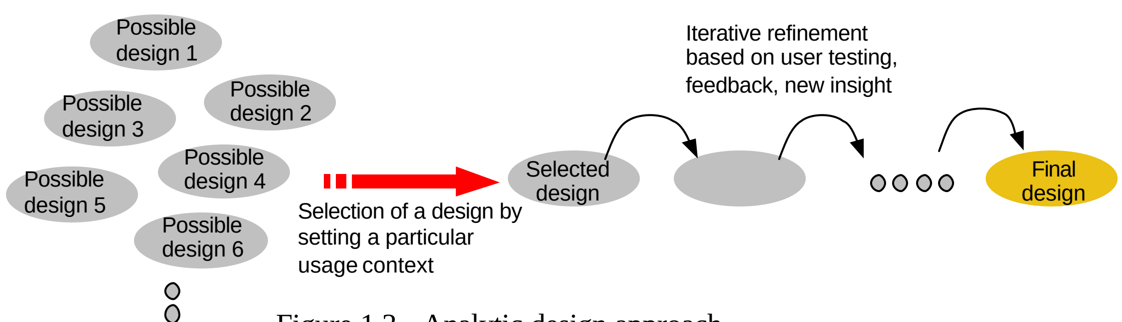


Figure 1.3 - Analytic design approach

This means that firstly various possible designs are considered, addressing the problems in the design process mentioned above: the initial interface can be designed simply by selecting the right set of options from the design space. Comparing different interfaces is a matter of locating the interfaces in the design space, clearly showing that interface A is the result of selecting a set of options (a1, a2, a3,…) in the space, whereas interface B is the result of selecting a different set of options (b1, b2, b3,…). Because the design space covers literally all possible designs, it can produce within it the interfaces for any different devices. Selection of design options can be made for a desktop PC interface with large display and a mouse, or for a small mobile phone interface. The design space also covers possible interfaces for users' individual differences, again a matter of selecting the right set of options. The bottom line is that once a design space is properly constructed, *multiple user interfaces* can be designed from this space by considering particular usage contexts. The difficulty in this design approach then is to construct the design space by considering all the possible design elements and attributes, itself a process which can be very complicated. Also selecting the right set of design options can be a difficult process, in which target usage and users have to be identified and relating interface elements to target usage and users can be ambiguous and uncertain. Alternative design opportunities always exist but with this general approach, design process becomes more manageable.

In summary, this approach attempts *to design not a specific interface, but to design a possibility space where a specific interface can be specified in various ways*. This is the same as the idea of increasing the level of abstraction so that the currently thinkable artefact can be viewed as one of the many possible instances that can be derived from the abstraction [Smith *et al.* 00, p115]. Imaginable is the kind of future system where the user-interface is dynamically determined depending on the usage context at a given time with a given user. This is one of the core elements of the concept of the "unified user interface" [Akoumianakis *et al.* 00], directed towards developing an open, flexible user interface that can adapt to various user preferences, abilities, skills and needs, on the fly. Whether a specific design can be realised from the possible space *automatically* or not is not the main concern of this thesis, but it is acknowledged that this will become a very important technical concern for such systems that brings up a specific interface among many possible ones.

The particular application domain of user interface concern in this thesis is *video content browsing* using keyframes extracted from the video sequence (explained in depth in Chapter 2). Use of other useful data from video (such as audio and subtitle) to enrich information to the users is not considered in this work, although we are planning this in the near future. As mentioned above, a single user interface cannot support the diverse context of usage and users, and this applies to video content browsing interfaces, too. There are various novel content browsing interfaces proposed and adopted in many experimental and commercial video systems at present, but most of them use a particular interface, without considering what other possible interfaces there could have been. In this work, it was considered particularly important to be able to understand the attributes and characteristics of the video browsing interfaces under consideration, rather than coming up with one specific interface and trying to refine and optimise that for a particular target usage. It was considered that this is a necessary concern from an examination of the recent trends and future perspectives in this area. The structure of this thesis is illustrated in Figure 1.4 below, from the view point of an analytic design process.

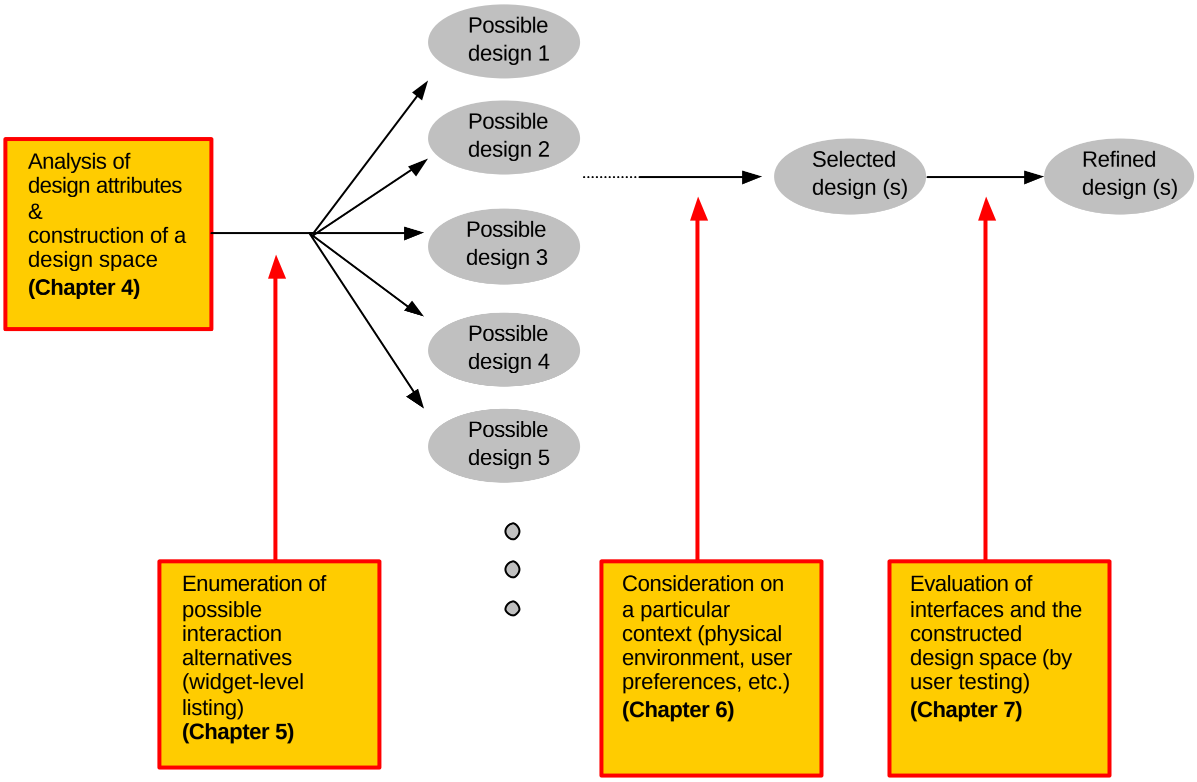


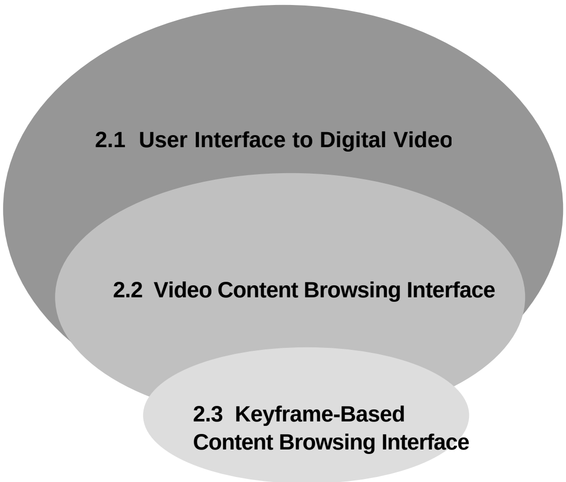
Figure 1.4 - Approach in the design of video browsing interface and corresponding chapters in the thesis

This thesis starts by reviewing current ideas and example interfaces for video content browsing (Chapter 2), which provide important clues on constructing the design space in later chapters. Following that, a digital video system called Físchlár is introduced (Chapter 3) as background information, which will be used throughout this thesis as the browsing interface testbed and evaluation. Analysis then starts by identifying important attributes of video browsing interfaces (Chapter 4) largely based on the current ideas used in the literature, and enlists possible design options and interaction alternatives for each of the attributes, forming a design space where a set of design options can be simply picked up (Chapter 5). Then a particular context for the target browsers' use is considered to select one or some of the possibilities (Chapter 6), where concrete examples were examined for device concern and user concern. Here a simple user classification is done to select suitable interfaces for different user groups. Then finally, the evaluation of the design of the selected individual interfaces is conducted (Chapter 7). The evaluation is user-oriented and because it is intended to reveal interesting and useful issues in users' interaction with the video browsing interfaces that might have been overlooked or not addressed in the previous parts of the thesis, it is also the evaluation of the constructed design space, its dimensions and values, interaction alternatives and overall the way we formed the whole idea of video interface design around the design space.

CHAPTER 2

User Interfaces for Video Content Browsing and Keyframe-Based Browsing: a Review

Reviewing past and current studies, experimentation and observing the systems available today is an important part of this research. As will be mentioned in Chapter 4, having a close look at the available and "thinkable" user interfaces has become an important clue in analysing the dimensions of interfaces and especially identifying the right set of design options for each of the dimensions. This chapter reviews the published digital video interface work of other projects and companies, focusing from the overall interface to content browsing down to various keyframe-based browsing interfaces.



2.1 The User Interface to Digital Video

Concerning the user interface side of system development, digital video information retrieval systems naturally lead to more complex front-end interfaces, as the rich information attributes (visual, audio, and other meta information) of the video medium, their indicators and controlling widgets have to be used and displayed on the screen. As found in most areas of computer system development, the user interface concerns for digital video retrieval systems both in the research and the commercial sectors are often not an aspect of central importance. In practice, much emphasis and interest has gone on developing video data-specific storage and compression methods such as MPEG-7 as well as automatic content analysis primitives such as camera shot boundary detection and object tracking through frames. The outcomes from these technical considerations are that they often dictate how the user interfaces should be like for the resulting video systems. When the enormous possibilities in digital video are perceived and implemented from the technical point of view, it is hardly surprising to see these trends of technologically-driven system development being so dominant.

To be able to identify the current trends in user interfaces in digital video research and products, some typical user interface features that have been designed and implemented are identified, largely based on our review of current video retrieval systems, both experimental and commercial, as summarised below. This list of features which now follows, will be used later to construct a table of systems and which features they provide, thus indicating the various emphases of these systems.

- The 'Cataloguing tool' is a distinct interface approach designed for the cataloguers or indexers of video material. A 'Manual cataloguing tool' provides a human cataloguer or indexer with an easy way to navigate video sequences frame-by-frame and allows video segmentation and text annotation. A 'Semi-automatic tool' is basically the same kind of tool as the manual one, but is particularly designed to simplify the indexer's work by automatically segmenting and arranging the content, allowing an index to view and make changes or add further annotations. This tool usually provides a playback screen and a segmented shot list with keyframes along with text annotation fields, etc. The 'Threshold adjusting before automatic segmentation' is set to an automatic shot detecting sensitivity so that more appropriate segmentation of video can be done depending on the video content. For example, Excalibur's Screening Room [Excalibur] allows the indexer to pre-select a particular film genre (animation, drama/comedy, documentary, etc.) for the video in question before automatic segmentation is conducted and this helps to produce more precise segmentation result.
- 'Keyframe-based sketch-drawing' is a content-based query tool where the user defines static visual features (colour percentage, texture, shape, etc.) in a drawing tool and the system matches this user-drawn sketch against video keyframes from within the database. Basically this is a tool found in the popular content-based still-image retrieval systems such as QBIC system [Ashley *et al.* 95], but quite frequently adapted to video retrieval systems to retrieve keyframes. Though often criticised for not being capable of addressing motion or audio attributes of video information (such as in [Dimitrova 95] and [Iyengar & Lippman 98]), this could be one of the complementary search tools based on a video's visual characteristic.
- 'Histogram manipulation' is a query tool where the user can modify the histogram of a

keyframe's visual features such as its colour and then request other keyframes with a similar histogram. This could be useful for users who first of all know what a histogram is and how to interpret it, by helping specify low-level visual features very specifically. Query-formulation is often the most problematic stage for novice users as it can be difficult for them to figure out, for example, how to use a sketch-drawing tool or histogram tool. One solution offered is the QBE (Query-By-Example) method where example items are presented to the user and s/he can simply query asking for similar items to the one s/he has selected. An approach based on 'Keyframe-based QBE (Query-By-Example)' is such a method where the user can browse through a set of keyframes from video sequences and request a new set of keyframes that are all similar to any particular one.

- 'Motion-based sketch-drawing' extends keyframe-based sketch-drawing by including user specifications of object and/or camera motion (in addition to objects' static characteristics) as an input to searching for video sequences. This motion query tool comes from the idea that a good IR system should provide search tools based on the medium's attributes (in the video medium's case, audio, image and motion attributes). One of the difficulties in providing this feature is that composing motion-based query pictures could be a complicated task for the user, as s/he has to be able to be quite specific in defining not only objects and their characteristics, but also the movement of those objects and different camera motions as well. Saving a composed query for later reuse or for other users can be a way to reduce this problem (as in MovEase system [Ahanger *et al.* 95]), as well as 'Motion-based QBE (Query-By-Example)' which allows the user to browse through a number of scenes with similar visual and motion characteristics.
- Also listed in some of the publications which form part of our review of the area are various Video browsing methods, and these can be sorted roughly in order of their degree of abstraction. Representing a video sequence as a single line of textual description (usually by manual annotation) or as a single thumbnail extracted from a representative keyframe are two common video abstractions used in most video systems. Transcript displaying is also one way of representing a video's content, though in many systems such a transcript may not be readily available unless there is some means of automatically generating transcripts from video content.
- 'Keyframe lists in chronological order' is the common abstraction method that displays a set of keyframes within a shot/scene/programme, and this is often called 'storyboard.'
- 'Options for different granularity of keyframes set' is often used in a networked environment to help the user have a look at the content of video before having to download large size data. Users usually have options for storyboards sampled at different rates. This idea of presenting different levels of detail in keyframe browsing will be further elaborated later in this thesis.
- 'Interactive hierarchical keyframe browser' is a particular video browser tool that can show all keyframes in a video, hierarchically – following a particular portion of interest brings up more

detailed keyframes for the user.

- The 'Keyframe slide show' is one form of video abstraction that shows a set of keyframes in one location temporally. Screen 'real estate' can be saved with this method, because only a single rectangular space for all keyframes is required, at the expense of the user having to keep on looking at the changing keyframes constantly.
- 'Video summary playing' is a moving extension of 'Keyframe slide show'. It consists of a condensed version of video sequences much the same as those movie trailers, showing only the most interesting (or important) sequences of a video. Though the resultant sequence is still something which has to be played (i.e. it is time-dependent), the length of time the user has to watch is reduced, thus saving browsing time. The idea of automatically condensing down the video sequence into a short trailer version has been further investigated in [Lienhart *et al.* 97] [Christel *et al.* 99].
- 'Playback' is the normal VCR-like tool that plays video sequences, as seen in almost all video systems. In some cases, a user's ultimate goal in using a digital video system might be watching an entire long programme using such a playback tool, or it might be just previewing a small, low quality playback to locate a sub-section of video material, with the analogue full-version sequence requested for later off-line playback.
- Browsing support such as synchronised presentation of more than one video abstraction (transcript + playback, keyframe list + playback, etc.) and text/scene search capability in this synchronised mode can also be achieved within the one system.
- 'Intelligent keyframe selection' indicates any content-based method that selects one or more frames from a video sequence, so that those selected frames are chosen to represent the video content. The use of keyframes within a browsing interface will be further explained throughout this thesis. This issue of keyframe selection is included here because whatever frames are selected and displayed on the screen will surely affect the user's browsing performance and behaviour – this is predominantly a technical issue but is also an interface concern, too.

These various interface features for video retrieval systems have been studied and experimented in research projects and companies with different emphases in different areas. Based on the feature list above, we now construct a table to show a matrix of the above-mentioned interface features and some of the example systems that provide those features (Table 2.1).

Feature \ System	SWIM	MediaSite (Informedia*)	VISION	WebSEEk	Internet CNN NEWSROOM	VideoSTAR	FRANK	CAETI*
Cataloguing tool								
Semi – automatic cataloguing tool		O						
Manual cataloguing tool						O		
Threshold adjustable before automatic segmentation								
Textual query								
Natural language (or Keyword) input	O	O	O	O	O	O	O	O
Category or Keyword list browsing	O	O	O	O	O	O		O
Use audio information for indexing/searching		O	O		O		O	O
Automatic full transcript generation**		O						
Visual query								
Keyframe-based sketch-drawing	O							
Histogram manipulation				O				
Keyframe-based QBE	O			O				
Motion-based sketch-drawing								
Motion-based QBE	O							
Video browsing method								
Textual description	O	O	O	O	O	O		O
Transcript		O			O		O	O
Single keyframe	O	O	O		O			
Keyframe list in chronological order (storyboard)		O					O	O
Option for different granularity of keyframes set								O
Interactive hierarchical keyframe browser	O							
Keyframe slide show				O				O
Video summary playing		O						
Playback	O	O	O	O	O	O	O	O
Transcript + Playback synchronisation		O					O	
Keyframes + Playback synchronisation		O					O	
Text search with Playback and/or Keyframes sync						O	O	
Intelligent keyframe selection	O	O		O				O

*This project is not yet completed at the time of writing, thus could facilitate more features in the near future.

**Partial transcript generation such as word spotting is included in previous 'Use audio information for indexing/searching.'

Table 2.1 - Feature-System matrix

Feature \ System	Pop-Eye / OLIVE*	VideoQ	MovEase	NeTra-V	EUROMEDIA*	Screening Room	VideoLogger
Cataloguing tool							
Semi – automatic cataloguing tool					O	O	O
Manual cataloguing tool							
Threshold adjustable before automatic segmentation						O	O
Textual query							
Natural language (or Keyword) input	O	O			O	O	O
Category or Keyword list browsing	O	O			O		O
Use audio information for indexing/searching	O					O	O
Automatic full transcript generation**	O						
Visual query							
Keyframe-based sketch-drawing			O				
Histogram manipulation							
Keyframe-based QBE							
Motion-based sketch-drawing		O	O				
Motion-based QBE		O		O			
Video browsing method							
Textual description		O	O		O	O	O
Transcript	O					O	O
Single keyframe	O	O	O	O	O	O	O
Keyframe list in chronological order (storyboard)	O				O	O	O
Option for different granularity of keyframes set					O	O	
Interactive hierarchical keyframe browser							
Keyframe slide show							
Video summary playing							
Playback	O	O	O	O	O	O	O
Transcript + Playback synchronisation							
Keyframes + Playback synchronisation	O						O
Text search with Playback and/or Keyframes sync							O
Intelligent keyframe selection					O		

*This project is not yet completed at the time of writing, thus could facilitate more features in the near future.

**Partial transcript generation such as word spotting is included in previous 'Use audio information for indexing/searching.'

Table 2.1 (Continued) - Feature-System matrix

Table 2.1 shows some of the current trends within the area of interface development for video browsing and access and the provision of these features within experimental and commercial video systems at the moment. Observing the distribution of the circles in the table shows that:

- Most systems use a textual query interface and few provide any form of visual query interface, probably indicating the need for further research in this area;
- Most systems use keyframe(s) as their video browsing method;
- Playback is provided in all listed systems indicating that playback is regarded as a most important interface feature, and
- Whereas most systems provide more than one video browsing method (often transcript + playback and/or keyframe + playback), browsing aids such as synchronisation between different browsing methods is often not facilitated.

It is understandable that individual systems do not provide all possible interface features, as different systems will be designed to cater for different target users and domains. However, it is observed that there are cases of a feature being provided when some related other features are absent.

To help us think about user interfaces for digital video IR systems more clearly, it is useful to break down the interface into several elements supporting different user actions (of course it has to be borne in mind that at a later stage the evaluation of the whole interaction, rather than a part, will have to be done, as is discussed later in Chapter 7). Here the idea of identifying different stages of a user's information seeking behaviour becomes useful - starting with deciding which information source to use, searching for a document in the decided collection, searching for a wanted part in a document, reading that point in the document, then going back to searching, and so on. This is described in, for example, the seven stages of action [Norman 88, p48], the various information-seeking sub processes [Marchionini 95, p49], the four-phase search process [Shneiderman *et al.* 97] and the eight sequences of interaction cycle [Hearst 99, p263]). Although these stage models do emphasise the unpredictable and non-purposeful change of directions between the stages, thinking in terms of the individual stages is useful in helping us to clearly consider the user interface because each stage of information seeking usually requires different interface support. Figure 2.1 below illustrates a simple categorisation of supporting UIs, roughly responsible for different stages of information seeking models.

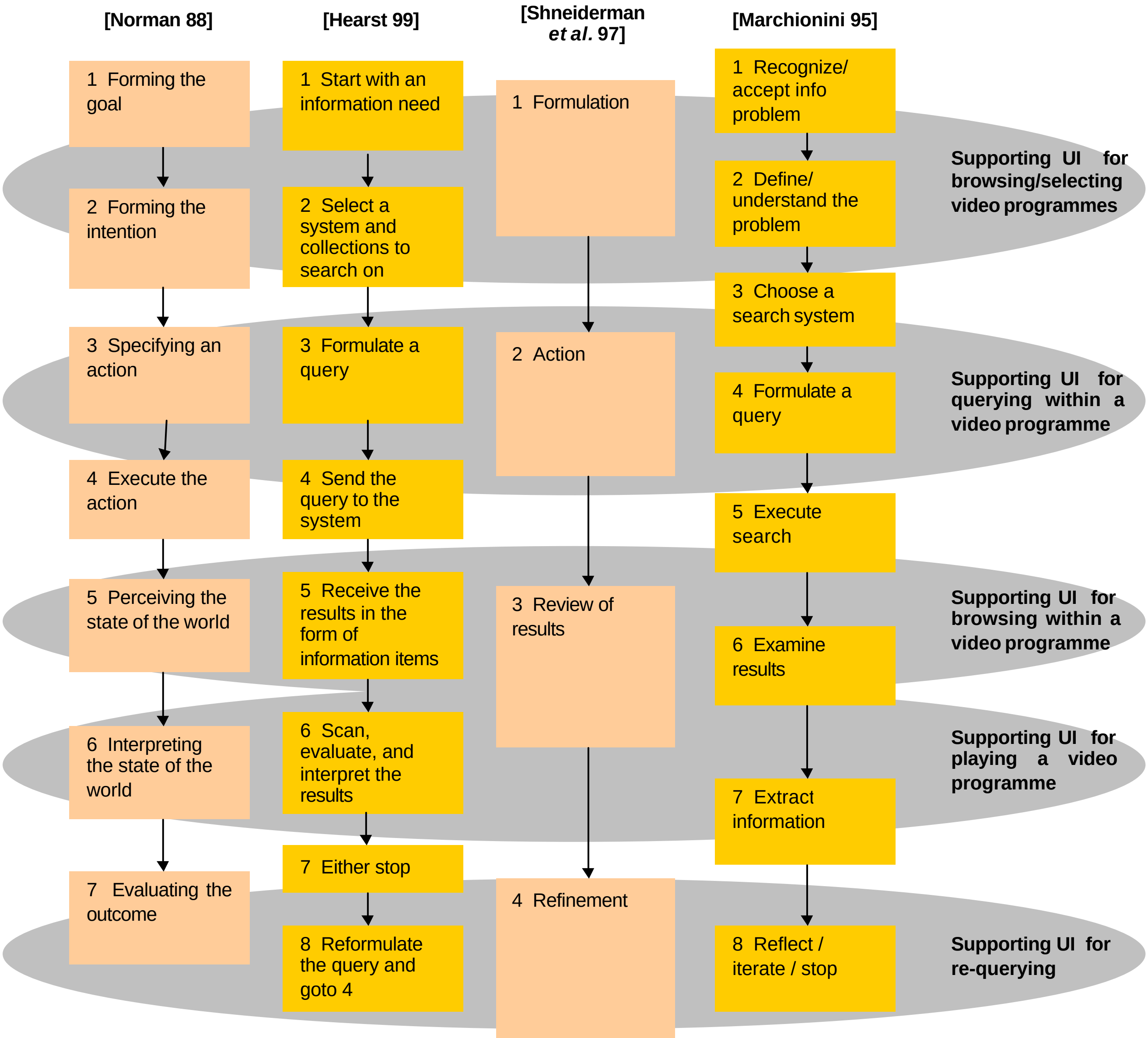


Figure 2.1 - Information seeking stages and the corresponding interface elements that support them

On the right-hand side of the Figure, roughly grouped are the supporting user interface elements by the stages of information seeking. They are:

- Interface elements to support browsing / selecting video programmes (as a collection)
- Interface elements to support querying within a video programme (content querying)
- Interface elements to support browsing the content of a video programme
- Interface elements to support watching (part of) a video programme
- Interface elements to support re-querying the video library and/or within a video programme

Different systems targeted for different usage will have different emphasis on some of these elements than others. For example, a large video clip database designed for searching by news broadcasting staff will have a more elaborate library querying interface to help efficiently retrieve a wanted video clip, whereas a home digital video system will have stronger interface features in the areas of content browsing and playing.

It appears from our literature study that all of these different interface elements are in great need of far more investigation and experimentation than has been done to date. Only a small number of interface studies and experimentation can be found in the literature for different interface facilities, grouped as follows:

INTERFACES FOR BROWSING / SELECTING VIDEO PROGRAMMES:

Searching through a whole video library is often done with a query formulation interface by some form of meta data of the video such as title, date and description, as in other text retrieval interfaces. Most systems with very large video collections will require an interface for querying and browsing the collection. The Físchlár application [Mc Donald *et al.* 01] displays a main interface having an organised video item arrangement in different folders to easily sort, filter and browse as in common email management software. FilmFinder [Ahlberg & Shneiderman 94] is a good example of a novel visualisation of a query result with a set of slider bars as a user's query filter and the resultant films dynamically and immediately reflected as the user drags the slider bars. When there is a large number of items to be displayed, novel visualisation methods originally studied for bibliographic data collection (such as in [Veerasamy & Belkin 96] [Hearst 95] [Rao *et al.* 95]) can be adapted for visualising a video collection. Information space visualisation is one of the relatively well investigated area and many interesting ideas have been demonstrated. A good example of this is the Informedia project [Christel & Martin 98] where the video items from the collection as a query result was visualised in a scatter-plot display, with the relationships among video items in terms of query terms visible, as Figure 2.2 shows. Whether the unit of retrieval is an individual scene, a video clip or a complete programme, visualisation of the

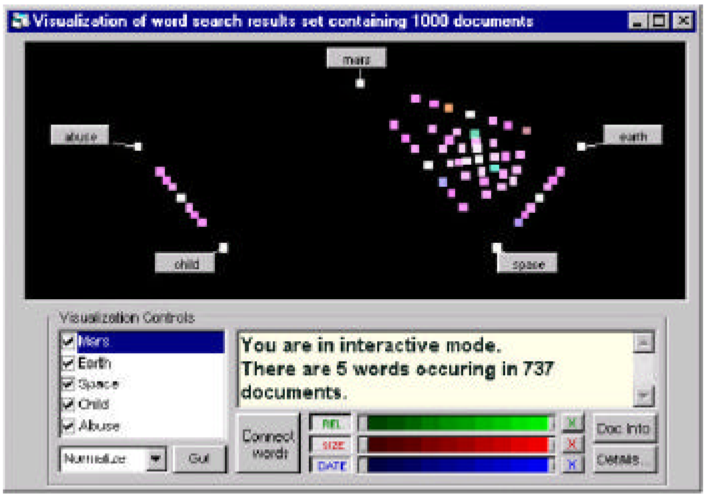


Figure 2.2 - Video query result visualisation interface in the Informedia Digital Video Library

user's query result is a good way of informing the user of the characteristics of the retrieval thus providing some clues on how to go about re-querying to more precisely retrieve the wanted video items or to commence viewing some of the retrieved objects.

INTERFACES FOR QUERYING VIDEO CONTENT:

When a query is to be specified by a user in terms of actual video content attributes, more specialised interfaces than those used to capture text-based queries are required. Manually constructed video databases such as VideoSTAR [Hjelsvold *et al.* 95] have a text-based query tool where the user can select different content attributes such as person, events and locations selectable from the list, all pre-indexed by a human indexer. When it comes to automatically indexed video collections, VideoQ [Chang *et al.* 97], NeTra-V [Deng *et al.* 98] and MovEase [Ahanger *et al.* 95] are some of the scarce examples of content-based querying interfaces where the user can query either by drawing on a canvas, objects with certain shapes and colours, specifying motion, or by selecting example video clips from the screen. The DICEMAN query application [Dunlop & Mc Donald 00] is an example of an elaborate query interface for digital video content where the user can compose various elements from the pre-defined set of elements and attributes such as persons, overall colour, movement, the order of objects appearing in the sequence, and so on.

INTERFACES FOR BROWSING VIDEO CONTENT:

Browsing video content is a very new concept and can be compared to a text retrieval interface feature for viewing abstracts or full-text. In video retrieval interfaces, what is known as "browsing video content" would be equivalent to FF/REW (Fast Forward / REWind) on a home VCR machine, or some selective keyframe browsing on a digital video system. The interface which is used for browsing video content is the main focus of this thesis, and this is further detailed in the next section.

INTERFACES FOR WATCHING VIDEO CONTENT:

Watching or playback of video may or may not be the final goal of a user's video searching interaction, but in the context we have known so far, playback is often the last stage of the user's interaction with the video and the interfaces supporting this are already common in the form of video player software, such as RealNetworks RealPlayer [RealPlayer], Windows Media Player [Media Player] and the Apple QuickTime Player [QuickTime] - basically all these have the same playback interface with buttons for play / pause / stop / rewind, volume control, and a timeline bar indicating the current point of playback in the context of the overall whole video clip. Digital recording set top boxes such as TiVo [TiVo] and ReplayTV [ReplayTV] are also playback centred and have remote control interfaces supporting playback features.

INTERFACES FOR RE-QUERYING:

Recognising the user's information seeking in interactive systems where the user keeps on modifying his/her goals as well as information needs while interacting with the systems, the facility for re-querying based on the current interaction status has become an important element for many IR systems. In video systems, early experimentation on systems such as WebSEEk [Smith 96] and SWIM [Zhang *et al.* 95] provided rudimentary interfaces for re-querying based on initial query results. More recently, the aforementioned DICEMAN query application [Dunlop & Mc Donald 00] provides an interface that automatically composes a query panel based on a user's selection of a particular clip from the result set display, from which the user can further modify the query specification to refine his/her second or subsequent query.

2.2 Video Content Browsing

There are currently a handful of different video content browsing ideas proposed or used that allow browsing of different granularities of abstraction of video content in different styles, either manually or automatically constructed. However the idea of browsing the content of video sequences is such a novel concept that has come to the fore only since technological developments have made all sorts of video data processing on desktop computers feasible, and as of now this area needs to be explored and considered in far more depth and variety.

PROBLEMS OF SEQUENTIAL BROWSING

With conventional VCR machines at home, "browsing" the video content has so far meant simple FF/REW (FastForward/REWind) with jog-shuttle controller or buttons. Most software video players also have the same kind control functions with Play/Pause/Stop buttons and a timeline bar indicating the current point of playing, similar to Figure 2.3. The problems of fast forwarding and rewinding have been noted in several places in the literature [Taniguchi *et al.* 95] [Elliott & Davenport 94] [Arman *et al.* 94, p97] [Christel *et al.* 97, p21] [Yeo & Yeung 97, p44] [Boreczky *et al.* 00, p186]. Apart from the technical problem of the large size of a video file to be transferred on a networked environment, more fundamentally there is a problem of the sequential, single access-point, linear nature of the video medium when a person is trying to access the content: constrained by time, as we play faster and faster it becomes more and more difficult to recognise the content, with the cost of going back to re-view taking longer and longer. If one is searching for a particular scene in the video, the user is faced with fast forwarding while struggling to keep concentrating on the fast-moving sequence. As the playing point

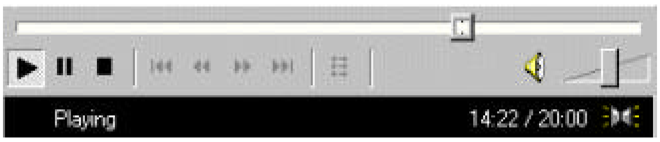


Figure 2.3 - Player control panel: Microsoft Media Player

is going further and further away from the starting point, the user becomes aware that if the wanted scene is not found from the rest of the sequence, s/he has to go backward further to resume the search before the starting point and all this arises because the user can look at only one point in the video at a time. The same kind of problem happens when one browses an audio recording, blindly trying forward and backward with no indication or clue of the context at hand. Attempts at making audio data browsable was explored in SpeechSkimmer [Arons 97]. SpeechSkimmer is a small, handheld device that plays speech recordings in various controlling mechanisms such as time-compression, pause shortening, emphasis detection and non-speech audio feedback. The fundamental problem we are examining here is that these media (audio and video) are time-based, making it difficult to browse through here and there within the data. An additional problem in video browsing is its multimedia nature, containing rich visual and aural information mingled together. A rudimentary browsing facility is found in some films distributed in DVD format, allowing the viewer to directly start playback from one of the pre-selected scenes in the video. These manually created scene access schemes with graphically attractive interfaces still need further elaborated presentation and interaction design, but they can be regarded as a starting point for browsing interfaces in that it allows direct access to the various points in video by providing visual summaries of each segment. With the enormous possibilities of manipulating data with the computing power at hand today, what we are confronting is the problem of design ideas for new methods of browsing video content, rather than the problem of manipulating the data computationally.

VIDEO CONTENT BROWSING IN TERMS OF THE LEVEL OF DETAIL

Probably the best way of thinking about video browsing ideas is in terms of various granularity (or level of detail) of the browsable content when the video data has become something browsable. Thus "something browsable" is often called *video abstraction* because the object the user is to browse is something abstracted from the original video sequence, thus something less in amount, less time-consuming and requiring less effort to look at - which is the whole point of browsing.

Describing video abstractions in terms of their level of detail is very useful in the video browsing interface design process, as it is also in the corresponding heuristics and guidelines of user interface design in general. As [Shneiderman 98, p523] amply emphasises, the idea of "overview first, zoom and filter, then details on demand" is regarded as important for any field where information is displayed, and this has been implicitly used with the experiences of designers and of common sense. In the conventional bibliographic information search facilities of the mid '90s and of today, a search result often shows a single line of highly condensed representations (usually the title and a date of a document as an overview), and clicking one of them displays the abstract at the bottom of the screen (zoom), then further user initiation would open up a new screen showing the full text of the document (detail on demand). In video browsing interfaces, the Apple video magnifier [Mills *et* *al.* 92] used hierarchically arranged frames taken from the video content, allowing a very coarse overview on the first row of the screen, then a level more details on the second row, then more details on the next row, as the user requests by pointing with a mouse. [Tonomura 97, p197] also explains the levels of indication granularity for multimedia representations, starting from an overview of a simple poster, then a composite poster, to a viewer and magnifier where certain part of the content can be viewed in detail.

The kinds of abstractions for video in its various levels of detail is very well examined in [Christel *et al.* 97], using as examples the Informedia Digital Video Library System's various video abstractions:

- title - a video document's name in text format
- poster frame - a single frame taken from a video document's content that can represent the content
- filmstrip - a set of frames taken from a video document's content
- skim - an assembly of significant bits of video sequences taken from the original video document

Various user-system interaction methods for each abstraction, and combinations of / transitions between abstractions are illustrated. The idea of different levels of detail will be usefully exploited in this thesis in the context of analysing the keyframe-based browsing interfaces later on (see Chapter 4).

OTHER VIDEO CONTENT BROWSING IDEAS

How a sequential, time-based video sequence can be made into something browsable is a very open question and new ideas for achieving this are needed. Among the limited examples of creative ideas for video content browsing, Video Streamer [Elliott & Davenport 94] is a set of adjacent frames of a video stacked on top of another, showing the edges of the frames under the top one. Being able to see the continuity of the frames without seeing most of the contents in frames makes a good way of condensing the visual contents of video. This idea can be useful in short clip editing and browsing, and has been applied in an experimental SCR video browser [Arman *et al.* 94]. A comic book style presentation [Boreczky *et al.* 00] is also an interesting idea, where automatically extracted representative frames are arranged and resized on a single page in a style similar to a page of a comic book. Other ideas on content browsing include various timeline representations [Aigrain *et al.* 95] showing keyframes on the timelines in 3-dimensional way to reduce the displayed space, automatic generation of a single or set of synthesised images containing video sequence information [Teodosio & Bender 93] [Kreyss *et al.* 97] and other graphic representations of related scenes. Different video abstractions have different interaction mechanisms, and part of the difficulty in coming up with browsing ideas lies in the interplay between video content representation and the kind of interaction required for that representation - this will be further investigated in this thesis in the context of video presentation analysis in Chapter 4 and interaction analysis in Chapter 5.

AIDS IN VIDEO CONTENT BROWSING

Browsing video content through some kind of an abstraction form can and should be aided by supporting information and other interaction styles. The ways of indicating the current time of the browsing can be also varied from a straightforward timeline bar currently used in almost all video player software (such as [RealPlayer] [QuickTime]), to indicating small quantifiable objects (as in [Arman *et al.* 94]) in the video abstraction, to more realistically visualising a time length as the depth of each browsing unit (as in [Aigrain *et al.* 95]). In browsing video content, users who have already seen the video before could benefit better by having some form of time information available along with the content abstraction itself (this notion will be used in linking interface elements to a particular user type later in Chapter 6).

Other aids to video content browsing include smooth transition between different video abstractions when available, synchronising the different abstractions when they are browsed or played, the facility to be able to maintain the current point of browsing when different abstractions of the same video is browsed (notably exemplified in 'synchronisation of different views' such as done in some browsers [Simpson-Young & Yap 96] [Li *et al.* 00] [Mc Donald *et al.* 01]), and all these are in great need for further research at the moment. More browsing aids as well as content abstraction itself are yet to be invented by creative and innovative designers of video browser interfaces.

2.3 Keyframe-Based Video Content Browsing

Video is composed of a sequence of still images, or *frames*, that are continuously changing at the speed of 25 frames per second, thus invoking an illusion of movement to a human eye. Much useful analysis can be done by using the data contained in individual frames - by applying image content analysis techniques and comparing the data between adjacent frames in video, we can tell when a camera shot suddenly changes, or whether the scene has got lots of motion or has very little motion and is stationary. Currently most of the automatic video indexing systems developed are based on this frame-by-frame image comparison. Due to the currently dominant use of frame analysis in video indexing, naturally the most common approach to content browsing interfaces in digital video systems (as seen from the Table 2.1) is displaying selective frames on the browsing screen: a reasonable and understandable browsing interface that allows the user to peep into the video content without having to play actual sequence.

The frames in video that can represent the video well are referred to as "keyframes", and can be used to turn the temporal content into static, glancible images. The effectiveness and usefulness of browsing such keyframes are evident in many places - a movie poster showing some snapshots of action scenes, a video tape cover showing interesting images from the video, web-based video databases showing attractive still images taken from the movie, and so on. Once the video sequence in its temporal nature is turned into a set of still images of keyframes, it becomes a matter of presenting them in different ways, and studies done in image perception and interpretation become useful and applicable. Subjective interpretation and possible ways of interpreting the same image in different ways have been investigated quite in depth [Shatford 86] [Krause 88] [Shatford 94] [Turner 94] [Enser 95], triggered by the need for consistent manual indexing of image repositories in libraries, archives and museums. When many such institutions realised the possibility of applying computer technology to achieve their main aims (preservation and access) and started their "image digitisation" projects to store their collections in digital systems, defining their indexing policies for subject access raised many serious questions. The use of controlled vocabularies such as ICONCLASS [Grund 93] and TELCLASS [Evans 87] came to the fore and a new thesaurus called the Art and Architecture Thesaurus [Molholt & Petersen 93] was developed to be able to index an image properly from its physical format to a subject reference more consistently and systematically. Also, computer-based image analysis has been developed in this context to provide automatic image indexing based on low-level attributes such as colour, texture and shapes in an image. From the difficulty in image indexing for computer systems came the idea of thumbnail-sized image browsing interfaces [Besser 90] which makes use of the exceptionally efficient human visual recognition system [Enser 95, p148] rather than attempting to precisely fine-index images, and since then the use of displaying a small set of pictures as a query result has become a *de facto* standard feature in image retrieval interfaces. Indeed, nowadays it is difficult to find any image retrieval system that does not provide thumbnail browsing interface - CD-ROM picture collections, museum kiosks, Internet art galleries and image search engines all have this thumbnail browsing feature, allowing the user to quickly browse through a large number of pictures so efficiently. The efficiency of displaying keyframes from video is in the same line as these image browsing interfaces, but there is one crucial difference: keyframes are more than a set of pictures, because there is a clear idea of time progression among the keyframe images - a set of keyframes is a *temporally ordered* set of images. Thus, there is one more thing to be considered in keyframe browsing when compared to the usual picture browsing interfaces. This is illustrated in Figure 2.4 below.

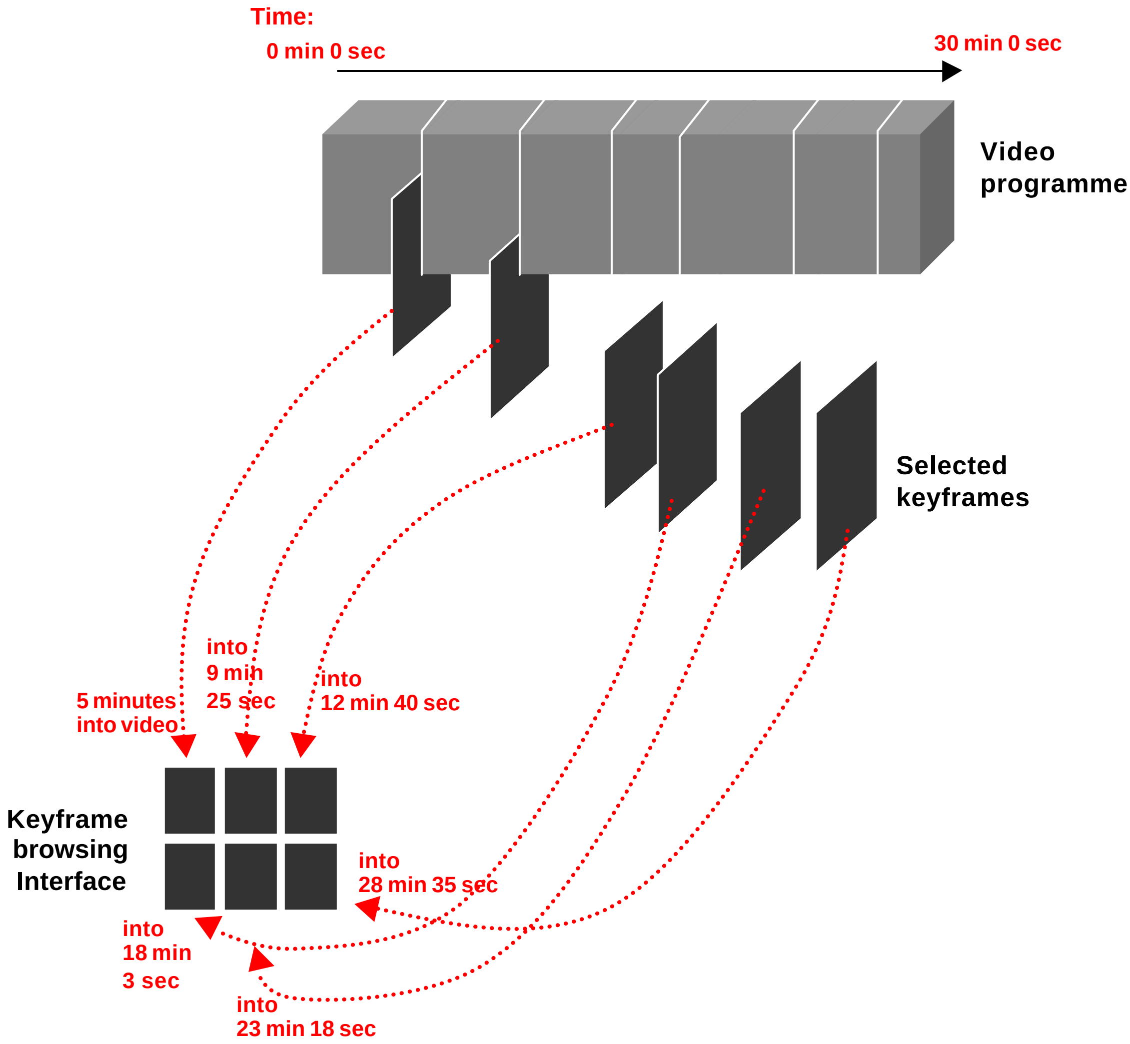


Figure 2.4 - Keyframes selected from video to be displayed on a browsing interface

In displaying keyframes, obviously the important decision has to be made on which frames in the video and how many of them should be selected to represent the video content. The *keyframe selection* process can be done manually by a human, or automatically by a system. In automatically selecting keyframes, there are various ways but they can be summarised as follows:

- Subsampling - the simplest way would be to select frames at a regular interval, for example taking one frame every 1 minute. This will result in a set of keyframes evenly distributed throughout the video (see Figure 2.5). However, the selected keyframes might not show meaningful or pertinent visual content because of many reasons. For example, the selected keyframe might show an image when the camera was unfocused, or before a main character walks into the centre of the frame, or may show the same images across many keyframes due to a long camera shot, and so on.

- Automatic video segmentation - by segmenting video sequence into meaningful chunks and selecting keyframes from each segment, the resultant set of keyframe can be more visually meaningful. A segmentation method dominant at the moment is called *shot boundary detection*, where individual camera shots taken in the video are detected thus chunking the video into a number of camera shots. This can be done by analysing individual frame images and comparing adjacent frames, frame-by-frame: during one camera shot the adjacent frame-by-frame differences are relatively small, because they usually contain continuous, uninterrupted camera shots. When one camera shot ends and the next shot starts, there is a sudden large change between adjacent frames, and this can be detected quite straightforwardly. Once camera shot boundaries are detected, keyframes are selected based on this segmentation. The easiest way of selecting keyframes from shots is to take the first frame or the last frame or the middle frame from each of the detected camera shot, as is done in most currently available systems. Then the resultant keyframe set would be a set of frames each of which is taken from a camera shot, and thus the size of keyframe set corresponds to the number of camera shots in the video (see Figure 2.6).

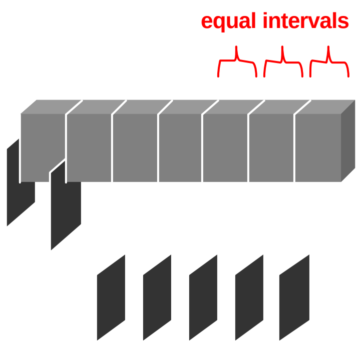


Figure 2.5 - Subsampling at equal intervals

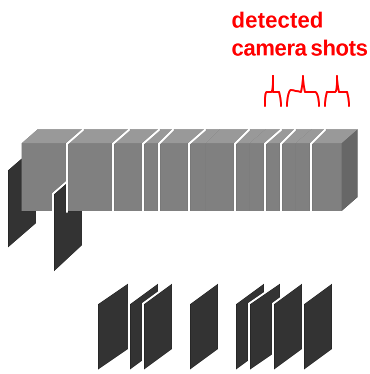


Figure 2.6 - Selecting the first frame of each camera shot

Most systems currently take one keyframe per camera shot, but sometimes there is a need for more than one keyframe per shot [Taniguchi *et al.* 95]. Automatic selection of the "right" keyframe is an area worth much consideration. Currently there are ideas and experiments on "intelligent" keyframe selection - meaning any method other than selecting arbitrarily such as first, middle, or last ones, based on some heuristic rules such as selecting a keyframe after a camera motion stops in a shot [Wactler *et al.* 96] [Dimitrova *et al.* 97] [Smith 96] [Zhang *et al.* 95] [Wolf 96] [Aoki *et al.* 96] [Uchihashi *et al.* 99].

In whichever way the keyframes are selected, the result is a set of keyframes extracted from the original video content, and it then becomes a design issue on how to provide a user interface that allows the user to browse through this set of keyframes.

Keyframe-based browsing can be categorised in different ways - a detailed categorisation will be introduced in the analysis part of this thesis in Chapter 4 - but to briefly illustrate the point, some common browsing interfaces are listed here:

- Storyboard - showing a set of miniaturised keyframes spatially on the screen in chronological order, allowing a quick browsing (as most of currently available experimental and commercial systems do, such as [Taniguchi *et al.* 95] [Virage]). This is also termed "keyframe list" or "filmstrip" in some literature;
- Slide show - flipping a set of keyframes one by one on the screen (as studied and developed in [Komlodi & Marchionini 98] [Smith 96] [DVB-VCR 98]);
- Hierarchically arranged browser - showing different levels of detail hierarchically, the user browsing keyframes in a drilling-down manner, especially suitable when the content itself is a structured video such as news or magazine programmes (as done in [Mills *et al.* 92] [Zhang *et al.* 95]);

Having many different ways of presenting a set of keyframes in large and small numbers, spatially and otherwise, this thesis attempted to clarify and classify possible design of keyframe-based browsing interfaces by analysing their elements and possible design options thus constructing a design space. This way, the keyframe browsing interfaces available in current digital video systems can be understood in a larger perspective within the design space. A comprehensive list of systems and their interfaces using keyframes will appear as examples of particular features, and are introduced and located in the constructed design space in Chapter 4.

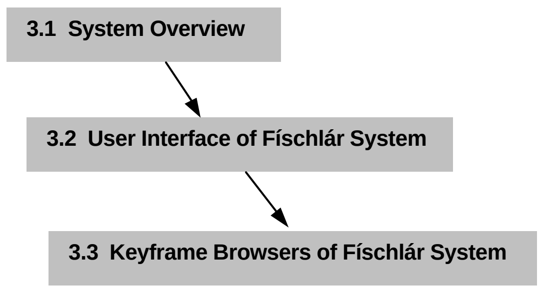
2.4 Summary

One of the research characteristics in user interface design issues for digital video retrieval systems is that there are not abundant example systems or projects to review at the moment. Video content browsing is quite a new idea and needs far more investigation. Among the small number of available systems, both experimental and commercial, the most common user interface for allowing video content browsing is by displaying on the screen a list of keyframes extracted from the video content. Currently several different variations of keyframe-based browsing interface have been explored, including a chronological display of keyframes ("storyboard") as most systems do, flipping through one by one ("slide show") and hierarchically arranged display. This thesis concentrates on different ways of designing keyframe-based browsing interfaces, and it will develop a methodology that allows us to develop many different ways of providing the user with keyframe browsing.

CHAPTER 3

The Físchlár Digital Video System and Its Keyframe-Based Browsing Interfaces

User interface design studies require experimentation with genuine working systems so that the actual look-and-feel of interfaces can be tested and real users' comments on using the interfaces can be obtained. The Físchlár system has been used in this research for testing out the developed video browsing interfaces against different video contents and different test users. This chapter introduces the system focusing on its overall user interface, and explains how it has been used in this research. Individual browsing interfaces of the system will be introduced in Chapter 4, along with the design framework shaped to develop these browsing interfaces.



3.1 System Overview

An experimental digital video system has been used throughout this thesis as a user interface testbed. *Físchlár*, a web-based digital video system that records and analyses TV broadcast programmes, has been developed in the course of last three and a half years at the Centre for Digital Video Processing in Dublin City University. The fully-automated system records broadcast TV programmes on users' requests, and once recorded, it applies its video indexing technique called shot boundary detection (explained in Section 2.3), segmenting the video into individual camera shots then extracting significant keyframes from each of the camera shots. The user can then browse through the video content using several distinctive keyframe browsing interfaces, and play the recorded programme by streamed playback from a high-capacity video server. All these features of recording, indexing, browsing and playback have been integrated into a single, coherent system, running 24 hours a day on a web server.

Figure 3.1 shows how the system runs internally. The system receives online TV schedules from PTV [Smyth & Cotter 00], a personalised television listings service on the web, and updates and displays today's and tomorrow's TV broadcast schedules for 8 terrestrial TV channels (RTE1, Network2, TV3, TG4, BBC1, BBC2, UTV and Channel 4). Clicking on a TV programme logs the recording command on the web server, which then updates the recording schedule on an SQL database that stores all the meta data for TV programmes. When the time for broadcasting comes, a TV tuner card inserted in a PC starts encoding the requested TV signal, in MPEG-1 format. The encoded file is stored on an 800 Gigabyte RAID level 5 disk and subjected to shot boundary detection. The shot boundary detection is the system's core video indexing component, and the Centre has been very much concentrating on developing different algorithms for shot boundary detection including colour histogram comparison, edge detection and Macroblock type comparisons in the MPEG-1 format. These are the algorithms that can segment video into individual camera shots by detecting the boundaries of camera shots, as explained in Section 2.3. The algorithms have been implemented and their effectiveness has been compared [O'Toole *et al.* 99] [Smeaton *et al.* 99] [Browne *et al.* 00]. The current version of the Físchlár system uses a colour histogram based algorithm to segment the video, which has 90% precision in its segmentation performance. The shot boundary detection's output data is stored in the video meta data storage archive, and from this keyframe extraction is done. The system's keyframe selection is based on each frame's colour analysis: all the frames within a camera shot are compared in terms of colour and from this an average colour distribution for the shot is generated as a colour histogram. One frame in the shot that is closest to this averaged colour histogram for the shot is selected and extracted as the keyframe for the shot. This method, though less elaborate than some other systems, selects keyframes reasonably well representing keyframes and results as a more pertinent set of keyframes compared to selecting the first, middle or last frame of each shot. Extracted keyframes are miniaturised into small, thumbnail-size JPG format files and stored on the web server along with their time alignment information.

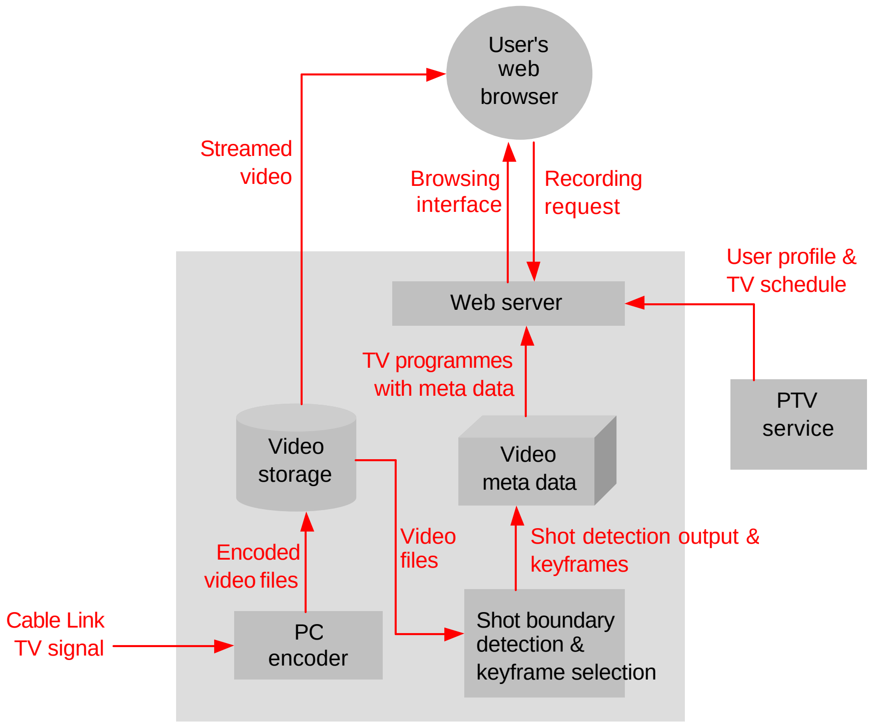


Figure 3.1 - Físchlár's internal mechanism

The system uses JSP (JavaServer Pages) to manage its web-based user interface, by dynamically writing out HTML pages to the user's web browser on the fly. Stored video programme files are listed on the browsing screen, and on the user's selection of a programme, JSP renders a keyframe browsing interface using HTML, JavaScript and JPG-format keyframes. Several different browsing interfaces are provided to the users as options. Figure 3.2 illustrates this in terms of the system's video data processing point of view.

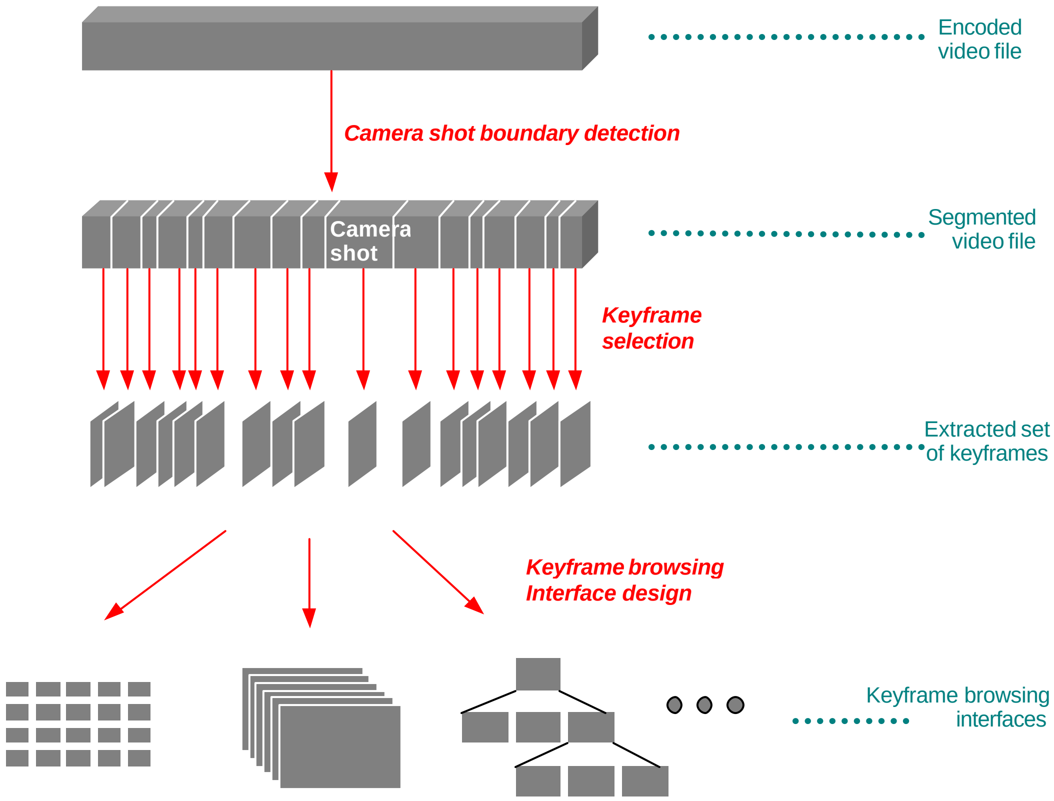


Figure 3.2 - Video indexing and provision of browsing interfaces in the system

In short, the Físchlár system takes in video sequences and extracts a set of keyframes to be presented to the user. The strong point of the system is that it is fully automatic and keeps on generating and storing sets of keyframes from newly recorded TV programmes everyday, providing an abundant supply and a variety of test video material for browsing interface design.

The system has been available since July 1999 within the School of Computer Applications accessed through a local area network, casually used by postgraduate students and staff. Continuous development in different areas of technology have been plugged into the system since then, and the system has served well as a testbed for newly developed techniques. Other than hardware improvement concerns on recording more than one TV channel at a time (as is done in the AT&TV system [Mills *et al.* 00], one of the very few other working systems that record multiple TV channels in parallel), various technical improvements have been tested and planned. Currently being developed is a technique to identify meaningful groups of shots into scenes thus further condensing down the size of the keyframe set (as called "keyframe filtering" in [Dimitrova *et al.* 97]), and also providing genre categorisation of each TV programme with personalised recommendations in collaboration with PTV, a personalised TV listings service [Smyth & Cotter 00] which uses a case-based reasoning mechanism to automatically recommend TV programmes that other users of the similar interest have indicated they liked. A desktop application software interface for Físchlár's underlying system is under development, to allow even better organised TV programmes and to provide a highly interactive playback-browsing experience [Mc Donald *et al.* 01]. Also underway is a technique to generate short playback summaries similar to a movie trailer, by combining bits of sequences taken from the video (as in [Christel *et al.* 99] [Lienhart *et al.* 97]). As these new techniques for content manipulation are developed, the system will integrate these for testing and experimentation against real TV programmes and with real test users. This further development at work is trying to overcome the currently perceived technological barrier of the many systems' shortcomings such as lack of semantic representation and our ignoring of audio information, as addressed earlier in [Dimitrova 95] and elsewhere. The system has become a working demonstrator around which all the project's ideas and new techniques are tested and refined. In system engineering side, to make the system more easily extensible and portable to other devices and platforms, we are planning to integrate XML technology.

A process is underway to deploy a streamlined version of the Físchlár system to the University's campus residences, to be able to test-run the system to a large number of users. This will then establish a real user base in a real situation where users will be using the video browsing system in their own way, from which the usage can be monitored and the system iteratively refined reflecting the observed usage patterns (more of this is described in terms of system evaluation in Chapter 7). The technical details of the Físchlár system are described in [Lee *et al.* 00a] [O'Connor *et al.* 01].

3.2 The User Interface to the Físchlár System

The user interface to the Físchlár system has been designed by the author starting in June 1999, and has seen continual changes and refinement based on discussion with project members and with peers through conference and workshop presentations, users' feedback and informal demonstrations, interviews and discussions, while new functionalities with technical developments have been plugged into the system. Whereas the content browsing interface of the system has been designed and added through the analytic design methods developed in the following chapters, the overall interface integration has been the result of applying general design guidelines, heuristics, graphic design experience and constant refinement through user feedback.

The Físchlár system was initially developed partly for the demonstration of our shot boundary detection mechanism, but also a testbed for experiments on browsing interfaces and our long-term plan was for system deployment to a group of users for the purpose of monitoring usage and feedback. Thus, an elaborate and careful user interface design and improvement of the system was always felt to be very important throughout the system's development. By using a complex frame structure of HTML pages and JavaScript, the Físchlár system tries to mimic a desktop software application. As part of this effort, there were many implementation limitations coming from the web-based hypertext paradigm where clicking a link of a page displays another page, whereas in a full desktop application software environment all sorts of more dynamic interaction can be easily implemented.

Overall, the web-based system was designed to look a little bit like a home VCR machine (see the Figures 3.3 and 3.4 below), having a dark gray background colour and a chrome panel. It was hoped that this VCR metaphor would provide first time users with this initial impression of the system as associated with a VCR. In fact, from the normal users' point of view the functionality of the Físchlár system is identical to a VCR - recording TV programmes and playing them. From this metaphor, any other additional functionality and features that can be found in Físchlár will be viewed as a kind of auxiliary or value-added, even the facility to browse through video content. Whether the content browsing feature - probably the most important feature from the project's point of view - is considered useful is for the users to decide. The deployment plan is such that the system provides our users with the conventional VCR features that they have already known for free, and allows them to use the system in their everyday life, in return for providing their opinions and feedback about the new features intermingled with the conventional features. This would be a good way of finding out whether a newly developed, technically-originated feature that has not been known to people would actually be felt to be useful and used by users in practice. The concerns on the evaluation of the system's usage as a web-based TV recording/playback service will be further explained in Chapter 7, after individual browsing interface evaluation has been conducted.

RECORDING INTERFACE

For the recording of TV programmes, the online TV schedule is provided for today and tomorrow with the title, time and description of each programme (see Figure 3.3). A user simply clicks on the programme s/he wants to record, and a message confirms that the programme clicked will be recorded. Because the system can currently record only one channel at a time, some of the programmes on one channel becomes overlapped with other recording schedules of other channels, thus making some programmes unrecordable. Those overlapping programmes and also the ones past their broadcast time are rendered in a gray colour and are unclickable, preventing a user error where clicking an unrecordable programme inadvertently. A simple colour scheme is used to distinguish between recordable (bright yellow), unrecordable (gray) and scheduled-for-recording programmes (red). On the right side of each TV programme, there are icons of thumbs for the user's preference indication. This five-scale feedback is used for automatic recommendation of future programmes, using a mechanism provided by PTV.

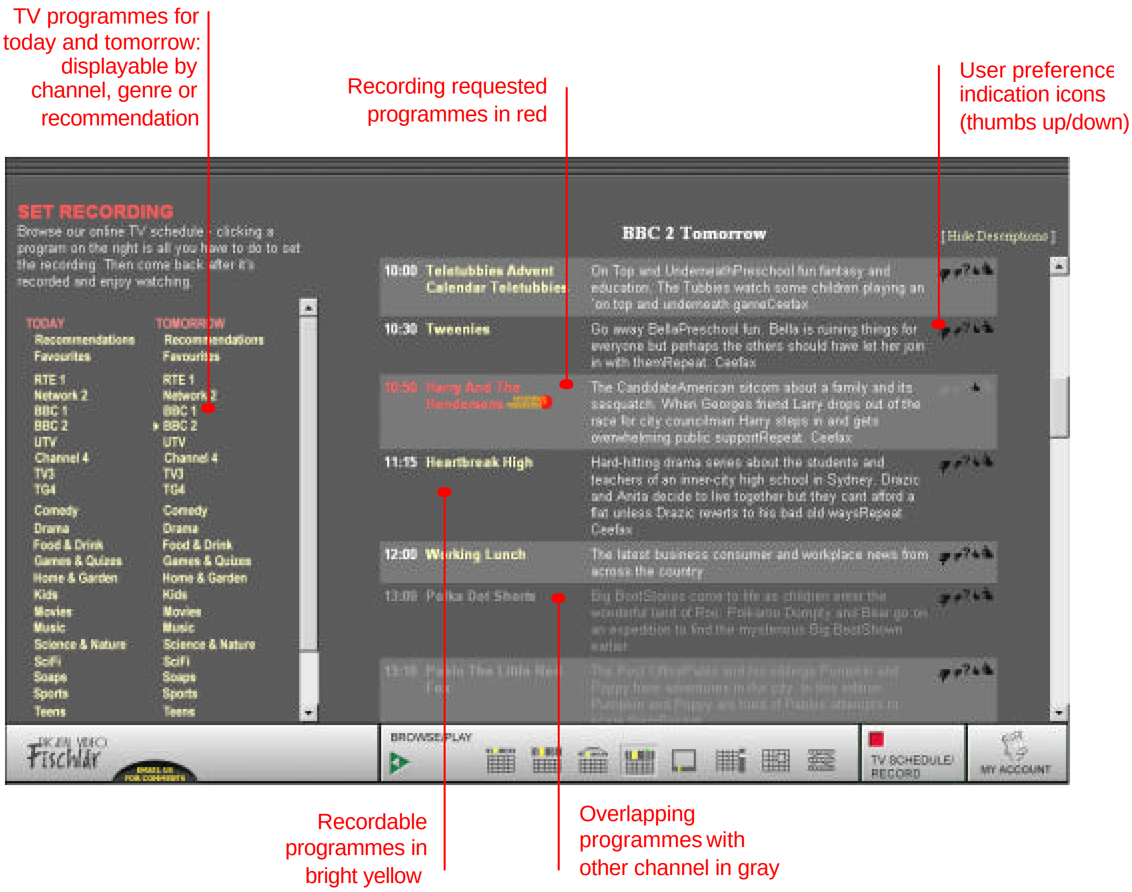


Figure 3.3 - Físchlár interface: recording interface

Having a large screen to display the TV schedule and allowing clicking on a programme to request its recording, is surely a far simpler and easier way of setting recording when compared to a conventional VCR system at home where usually the date, starting time, ending time and channel have to be set one by one. It would not be surprising if a survey was conducted to reveal that many home VCR users do not make use of the recording setting feature because it is difficult to do. VIDEO Plus+ code (also known as "ShowView" in Europe, "VCR Plus+" in America and "G-Code" in Asia) was launched in 1990 by Gemstar (http://www.gemstar.co.uk) to rectify exactly this problem of difficult recording setting by allowing the user to enter a simple unique sequence of numbers for a TV programme. Recent VCRs have tended to use larger display panels or even the TV screen itself where the user's setting status can be all displayed at once, making the interaction far easier as the user does not have to follow a single-track sequence of actions and remember what has been pressed before. Large computer monitor interaction with a mouse in the Físchlár system means that these problems usually disappear, enhancing the usability and encouraging so effectively the use of the recording feature - clicking a programme title is all that is required to set the recording.

BROWSING / PLAYBACK INTERFACE

The Browsing/Playback component of the system is an important feature of the Físchlár interface, allowing the user to browse the automatically extracted set of keyframes taken from the video contents, as well as playback from any point of interest from the browsing interface. The interface is illustrated in Figure 3.4 below:

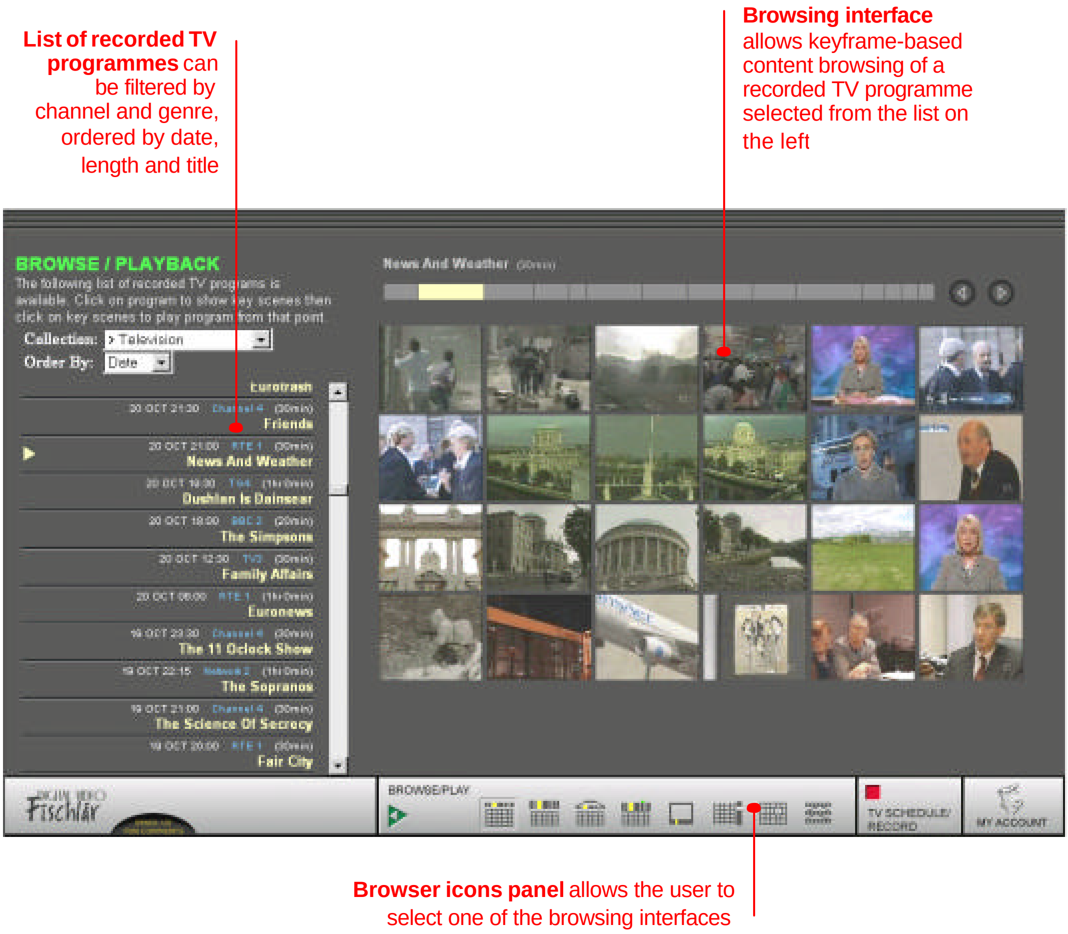


Figure 3.4 - Físchlár interface: browsing/playback interface

A rather simple browsing list of recorded programmes is on the left, with two drop-down option boxes for filtering and sorting the list. The right side of the screen is the main content browsing area, where the keyframes of the clicked programme from the list on the left are browsed with several different browsing interfaces. Keyframes extracted from the video content are used for browsing, and different styles and browsing aids have been used for different browsing interfaces. Clicking any of the keyframes at any browsing moment will pop up a small, separate window on top of the screen, starting playback of the video content from the clicked point in the video by streaming the sequence from a video server.

3.3 Keyframe Browsing Interfaces of the Físchlár System

The panel at the bottom of the screen (see Figure 3.5) shows a set of icons representing each of the keyframe browsing interfaces designed and implemented as a main part of this thesis. Currently there are eight different browsers on the panel, some of them having very distinctive styles and features, some of them similar to each other. The user can select any one of these browsers to browse the contents of any TV programmes recorded. These eight browsers have been designed from the design methodology developed in this thesis, and will be dealt with in depth in the subsequent chapters (6 browsers will be introduced in Chapter 4 and 2 more browsers in Chapter 5).

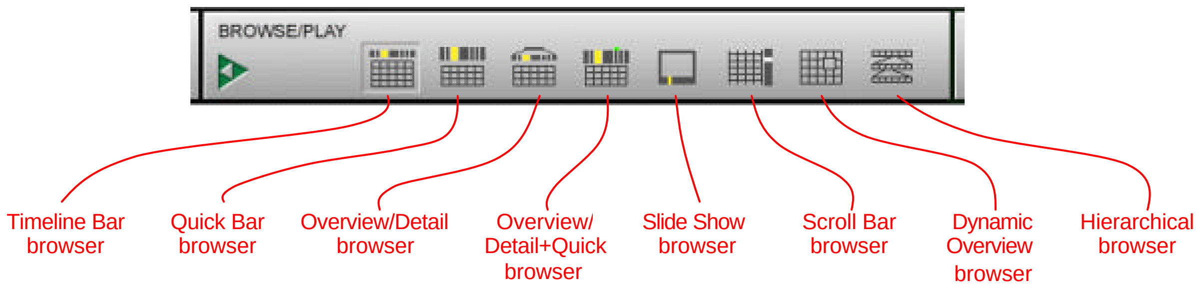


Figure 3.5 - Browser icons panel

For illustration purpose, one of the browsers in the panel, the *Timeline Bar* browser, is briefly described here. On selection of the Timeline Bar icon on the panel, the currently selected TV programme's keyframes are presented with the Timeline Bar browser on the main screen. The browser uses all keyframes extracted from the selected video programme, each of which is taken from a different camera shot. As the number of keyframes is usually very large - for example, a typical 30 minute soap opera on TV results in around 300 - 400 keyframes - even small, miniaturised keyframes cannot be all displayed on a single screen. Having a large number of images to display is one of the main characteristics of keyframe browsers and this fact will be used in the analysis from the next chapter. To be able to display many keyframes, the Timeline Bar browser presents chunks of 24 keyframes on

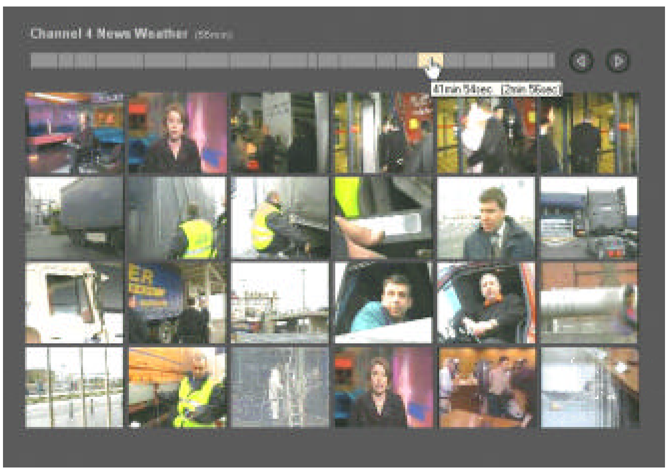


Figure 3.6 - Timeline Bar browser

a single screen, organised by the timeline at the top of the screen (see Figure 3.6). The timeline represents the whole length of the video, and chunked into many units each containing 24 keyframes to be displayed when clicked on. When a mouse cursor is over any of the units on the timeline, a ToolTip box pops up indicating the exact timestamp of that part of the video. The timeline allows the user to directly access the video content, showing keyframes from any part of the video. Thus a task of finding out a particular scene in mind or seeing roughly what the video is about is well supported by this browsing interface.

Having the fully working Físchlár system - an "automatic keyframe generator" - the question for the browsing interface becomes *what would be a good way of presenting a set of keyframes to the users?* Surely there can be many different ways of designing a browsing interface (including the Timeline Bar browser in Figure 3.6) for presenting a set of keyframes, and as is the case with any interface design, it tends to become an area of art and intuition. Displaying keyframes and using different browsing aids in various ways would be possible, and as clarified in Chapter 1, the study reported in this thesis took an analysis of design space particularly for keyframe-based content browsing interfaces, allowing us to think and reason more systematically about different elements and features of keyframe browsing interfaces. Finally, although in this thesis only the keyframe-based browsing interfaces are dealth with, the "missing information" such as sound, motion and subtitle will have to be considered which could well support users' interaction with the system.

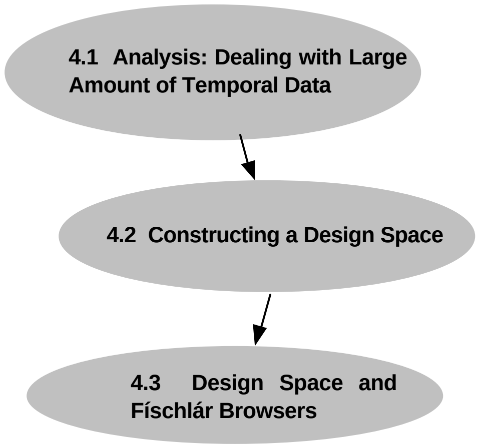
3.4 Summary

The Físchlár system is a web-based, fully working digital video recording, indexing and browsing system, developed to test new ideas in the area of video indexing. Currently the system includes an indexing module which automatically extracts a set of keyframes from a recorded video content, based on camera shots in the video content. The system's user interface allows the user to easily record TV programmes from an online TV schedule, and once recorded and analysed, allows browsing/playback. Concentrating on the browsing side of the interface based on the sets of keyframes extracted from the video content, this thesis develops a systematic method to design various alternative browsing interfaces, of which currently eight designs have been implemented and run on top of the system. The system is going through constant refinement and currently underway is the deployment of the system to the University's campus residences, to obtain real usage information and feedback for further improvement.

CHAPTER 4

A Design Framework for Keyframe-Based Video Browsers

This chapter shapes a design space for presenting a set of keyframes extracted from video content, by analysing the elemental features and possible options, then constructing a structured list of these possibilities. The basis of this analysis is noting a characteristic of the keyframe set to be presented - a temporally ordered, large amount of images. Actual design options listed are determined largely by considering available systems, though limited in number. The constructed design space will be visualised in a 3-dimensional diagram to help understanding, and some existing interfaces will be located in this space for illustration. Then six of the browsing interfaces for the Físchlár system designed from this design space will be introduced.



4.1 Analysis: Dealing with a Large Amount of Temporal Data

A set of keyframes extracted from video can be displayed using many different ways and styles. There have been numerous experimental and commercial developments in the various applications for which digital video can be used and each of these developed systems will have different visualisation and user interfaces. However, the development of a system having a single interface for video browsing would be too much of a simplification considering the many possible aspects associated with the ways in which video can be browsed as outlined in the previous chapter, and this would be likely to lead to poor usage and frustrated users. In practice, a single all-encompassing video browsing interface would be impossible and several variations on a browser theme are needed, each of which has to tackle a well-defined specification which can be obtained from many sources such as potential exploitation of the video data itself and prospective future users' requirements. Acknowledging the various factors that could be considered, the natural starting point for the work reported in this thesis became the exploitation of the video data itself, namely analysing the characteristics of video data that has to be handled. By way of analysis, different underlying *dimensions* of the video data itself and different *values* that each dimension can take have been identified. In other words, the video data we want to provide an interface to can be modelled in a multiple-dimensional space where each dimension contains a certain number of values. "Designing an interface" in this context becomes a matter of making a choice of different values for each dimension, different sets of choices resulting in different interfaces.

The data to be managed consists of sets of large numbers of keyframes in temporal order. Dealing with a large number of keyframes, the first dimension we identified is "Layeredness" - a matter on whether to display all available keyframes directly to users or to allow a way for more selectively presenting them. The temporal nature of the data leads us to think of two more dimensions - "Provision of temporal orientation," or how to make time information from the video visible to the user, and keyframe presentation style which we call "Spatial vs. Temporal presentation". Each of these dimensions is now described.

4.1.1 Layeredness

Most of the currently available digital video systems based on shot boundary detection need to display a very large number of keyframes. The Físchlár system takes one keyframe from each of the detected camera shots as many other systems do, and there are even systems which take more than one keyframe per shot [EUROMEDIA] [Taniguchi *et al.* 95].

TV programmes usually contain a large number of camera shots - for example, a 30 minute soap opera contains about 300 to 400 camera shots. This means that even when we take one keyframe per shot, this makes a set of 300 to 400 keyframes to be browsed from the interface. A two hour film can easily generate well over 1,000 keyframes for the user to browse.

Browsing a very large number of keyframes might not be a very efficient method, for the whole purpose of some users' browsing might be to have a concise view of the whole video content without having to spend too much time getting this overall view. Thus further filtering out and reducing the number of keyframes has to be considered in designing a video browser.

The method of reducing the number of keyframes ("keyframe filtering") may vary from simply selecting an arbitrary keyframe by grouping all keyframes within a regular interval, or there may be more complex ways of content-based and semantic-level keyframe grouping.

Whatever the method of reducing the number of selected keyframes may be, this implies a layered presentation where several different granularity levels can be chosen as an option, by the user.

At one end of the layeredness of keyframe-based presentations there will be a single keyframe that represents the whole video clip, as is frequently done for displaying search results for a video clip in experimental systems such as Internet CNN NEWSROOM [Compton & Bosco 95] and VISION [Li *et al.* 96]. One cannot expect this single keyframe to be able to convey the content of the whole video, but still it works as a kind of icon for the video and locates clips within the whole video. At the other end of the layeredness scale there will be the most detailed, full listing of keyframes with each representing a camera shot in the video, as most of current semi-automatic video cataloguing systems do (such as Virage VideoLogger [Virage], Excalibur Screening Room [Excalibur] and MediaSite Publisher [MediaSite]). Whereas this full keyframes listing shows the most detailed view of the video content, it loses some of its purpose for browsing because it takes so long to look at all the keyframes.

Between these two extreme layers, we can think of intermediate layers with different levels of detail, i.e. different numbers of keyframes to be used as the user's index into the whole video programme. It will not be possible to say which level of granularity is best for every situation as one user in a situation will have different needs from another user in the same or a different situation - a busy user might want to look at what the entire video content is about and have this condensed to just one screenful of small keyframes, while a user looking for a particular scene might want to browse in as detailed a way as possible and may thus be willing to spend more time on the task. It might also be the case that different types of user groups prefer browsing at different levels of granularity, leading us to consider a browsing interface with the level of granularity customisable for different user groups. In general a user might want to have an overview look at the content and if s/he finds it interesting then zoom in to see a more detailed view. This idea of "overview first, zoom and filter, then details on demand" [Shneiderman 98, p523] has been amply emphasised in the field of information visualisation, and the idea of layeredness can be used to conform to this design heuristic.

The important elements to consider in providing layered keyframe presentations are the number of different layers available in an interface and the navigational link between these different layers while the user browses the keyframes. There would be many different ways of specifying these possible values. For example, one might specify a set of values in this dimension as:

- Single layer
- Double layers
- Triple layers
- More than triple layers
- Number of layers customisable by user

Or others might specify a different set of values:

- Full-detail keyframe set (100% of original set)
- Medium size keyframe set (filtered keyframe set to 50% of the original keyframe set)
- Concise keyframe set (filtered keyframe set to 5% of the original keyframe set)

By looking at the current experimental and commercial systems available, the initial set of options in this work has been decided to be as follows. The distinction has been made on whether there is more than a single layer provided or not, and if it did, whether there is an explicit navigational link provided between these layers. Having this kind of explicit link means that in the interface the user can browse from one point of the layer to the same point in the different layer, thus maintaining the current browsing point in the video while jumping between different layers. This decision is largely based on the review of current systems available and to make a more sensible and concise representation later when a full design space is constructed:

- Single layer
- Multiple layer without navigational link
- Multiple layer with navigational link

This can be visualised as Figure 4.1 below, which will later become one of the axes in the final design space:

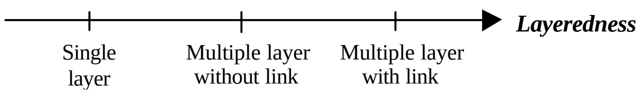


Figure 4.1 - Layeredness and some values

4.1.2 The Provision of Temporal Orientation

With the video medium having a temporal nature as one of its fundamental characteristics, it is important to consider providing (or not providing) a cue for the sense of time in user browsing. A set of keyframes usually provides no proper time sense or temporal orientation, other than a rough progression from one keyframe to the next. Looking at the keyframes selected based on camera shots can even distort and mislead the time orientation, as the number of shots and thus keyframes from one time segment can be very different from those in another segment of the video but of the same length, depending upon the frequency of camera shot cuts in the video content. A familiar time orientation used in many interfaces is known as the 'timeline bar' (either as an indicator or controller) and it is usually found in most video player software applications such as Microsoft's Media Player or RealNetworks's RealPlayer.

In keyframe browsing, it is easy for the user to lose the sense of time and at what point s/he is currently looking at in the video, and this work considered a designer's decision on ways of indicating the time sense as an important dimension in browsing. Several existing examples of this implementation are found such as simply time-stamping the numeric time in each keyframe (MediaArchive [EUROMEDIA], Virage VideoLogger [Virage], Screening Room [Excalibur]), displaying quantifiable objects to visualise shot length ([Arman *et al.* 94]) or a timeline bar indicating the current position of user's keyframe browsing (MediaSite Publisher [MediaSite]).

Again, identified in this dimension is a set of values, that is, a set of possible design options:

- The indication of *absolute time* in the currently viewed keyframe or set of keyframes (exactly how much into the video the current keyframe is)
- *Relative time* - an indication of the current browsing point in relation to the whole length of the video
- *No indication* of time (surely as a deliberate decision not to)

The above ideas can be again visualised as Figure 4.2 below, to be used later for constructing a design space:

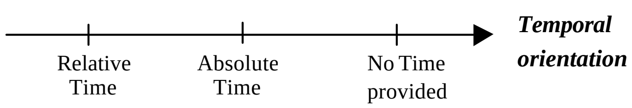


Figure 4.2 - Temporal orientation and some values

4.1.3 Spatial vs. Temporal Presentation

The conventional keyframe browsing idea would be to miniaturise each keyframe into a small thumbnail size image and scatter them *spatially* on the screen - in much the same way as the digital image retrieval user interfaces used in art gallery booths or CD-ROM picture collections where the user's query result is displayed as a set of thumbnail images so that s/he can quickly browse and identify the images s/he wants. But it is also possible to present keyframes on a user's screen one by one, the system automatically flipping through or the user manually flipping the keyframes. Considering the fact that the original video is time-based and sequential, it seems to make sense to present keyframes *temporally*, and this might even provide the feeling of watching the video.

Current digital video systems' user interfaces based on keyframes mostly feature showing thumbnail size keyframes spatially such as MediaArchive [EUROMEDIA], FRANK [Simpson-Young & Yap 96], MediaSite Publisher [MediaSite], Excalibur Screening Room [Excalibur], VICAR [VICAR], Virage VideoLogger [Virage] AT&TV [Mills *et al.* 00], TV Ram [Taniguchi *et al.* 95] and infrequently found are the systems presenting keyframes temporally such as DVB-VCR [DVB-VCR 98], WebSEEk [Smith 96] and the movie browser tool [Marchionini *et al.* 98].

An interesting video browsing study in comparing spatial presentation with temporal presentation in terms of different user tasks has been done recently [Tse *et al.* 99], reporting that spatial presentation was better for locating and identifying a particular object in the video, whereas temporal presentation was better for getting the gist of the video. Surely it is reasonable to think that a particular keyframe presentation method is more suitable for a particular kind of task than others, and it might also be the case with particular users' personal preferences.

Again, this dimension can be visualised as Figure 4.3 below:

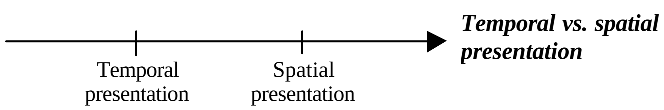


Figure 4.3 - Temporal vs. spatial presentation and its two values

4.2 Constructing a Design Space

So far three dimensions of keyframe-based browsing interface design have been established. These are the elements that have to be considered and decided when designing such an interface, and can be used as a decision-making tool.

Assembling this analysis of dimensions into one leads to a 3-dimensional space where each axis represents each dimension, as drawn below (Figure 4.4). Pointing to a location in this space determines three values taken from each of the dimensions and that makes one specific browser interface. In this way, "designing a browser" becomes *simply a matter of pointing one location in the design space* - this is the idea of design analysis which turns an interface design process into a concrete, simple and systematic decision-making process. Constructed is a visible set of possibilities of design space rather than a specific design, thus providing a designer all sorts of possible design ideas and a way of clearly comparing different design possibilities.

In this space we can locate positions of several existing video browsing interfaces, for example the SWIM hierarchical browser [Zhang *et al.* 95], DVB-VCR [DVB-VCR 98] and AT&TV [Mills *et al.* 00] (see Figure 4.4 below).

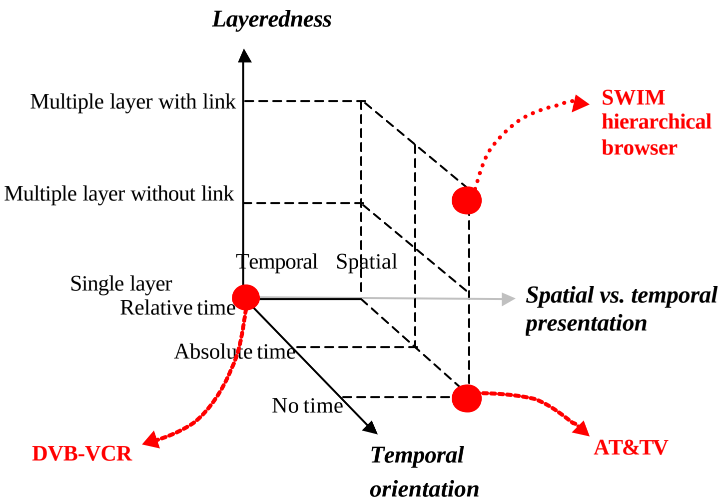


Figure 4.4 - Design space and three example interfaces' positions

The power of the constructed design space is that it shows different browsing interfaces more globally in their relative positions in terms of three dimensions, as well as bringing up new interfaces.

All three example interfaces located in this space above are in fact choices among alternative ways of designing interfaces.

This particular design space constructed here is based on the analysis of keyframe sets from video, and thus analysis approached from the data to be presented. When a particular point in the space is fixed (i.e. when a particular browser is specified), the important elements of concern are determined in terms of the data presentation, but still there are other decisions to be made - mainly on the exact interaction style and widgets to be used. This will be discussed and enumerated in detail in Chapter 5.

As the next section will illustrate, six distinctive browsing interfaces were designed based on the design space and implemented for the Físchlár system, and these do provide good coverage of the design space. Two more interfaces have been added later on (introduced in Chapter 5) that have the same locations on the design space with some other interfaces, to demonstrate further interaction options which will be described in chapter 5.

4.3 Design Space and Six Físchlár Video Browsers

The Físchlár system is currently running with eight different keyframe browsing interfaces, of which six are distinctive from each other and two are the same in terms of the design space. The users are free to use any of these interfaces to browse and play TV programmes they, or others, have recorded. This section will address the six distinctive interfaces' individual specifications in the dimension space, and the next section will describe each of the six interfaces. As the following figures will show, the objective is to cover as well as possible the previously defined design space in order to propose alternative browsing possibilities for the Físchlár system. The six distinctive Físchlár video browsing interfaces are listed in Table 4.1 below and are attached to a specific colour which will be used consistently in the following figures throughout this thesis.

Table 4.1 - Físchlár video browsers and assigned colours

(turquoise)	Scroll Bar browser (SB)	(light orange)	Overview/Detail browser (OD)
(light green)	Slide Show browser (SS)	(pink)	Dynamic Overview browser (DO)
(dark green)	Timeline Bar browser (TB)	(red)	Hierarchical browser (H)

The video browsers we designed are based on locations within the design space and are visualised here in Figure 4.5 below. This combined figure shows all essential dimension value choices for the six Físchlár browsing interfaces at one glance:

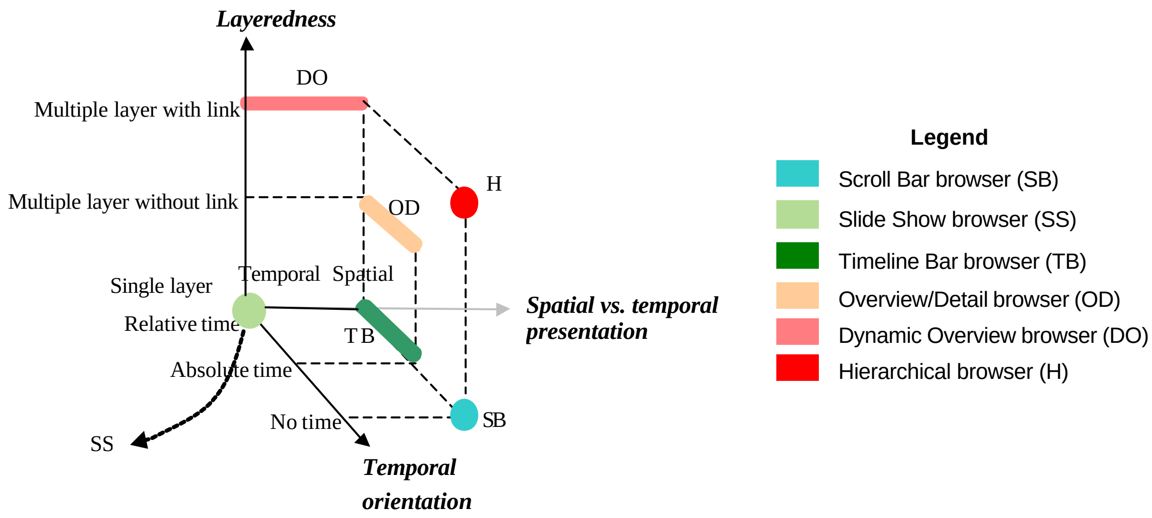


Figure 4.5 - Design space and current six Físchlár interfaces

Following this positioning of our six browsing interfaces in the dimension space presented, we now go into details of each interface.

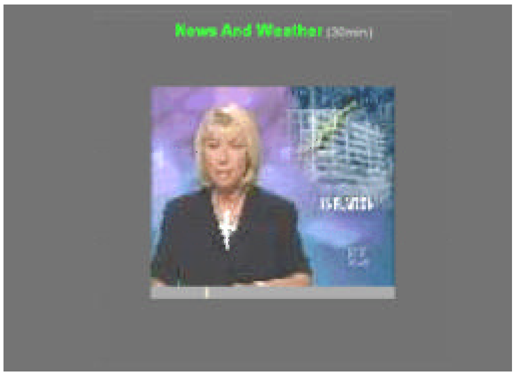
Presented below are screen shots, short descriptions and some selective and indicative comments from our users. Beginning with the Timeline Bar front end, interaction and dialogue with our users brought up new ideas and critiques, which then led to further ideas and then into additional front end interfaces. Added interfaces were again tested with users and, over time, some obvious usability problems were identified and removed as we then started to stabilise the feature set of each user interface.

The browsing interfaces presented below are a snapshot of the on-going refinement and discussion with our users within the Lab so far and will be further refined, although we note that these six user interfaces have become fairly stable at the time of writing.

Scroll Bar browser (Single layer - No time - Spatial presentation)

The keyframes are chronologically arranged in their full detail (one keyframe per camera shot), all put side by side on a long single page. The user scrolls up and down the keyframe list using the scroll bar to the right, to browse the keyframes. This browser is the most commonly used style in most digital video systems, sometimes called "storyboard" or "filmstrip." The locations of keyframes do not correspond to exact regularly-spaced time, because there can be more keyframes selected in one region of the video than other regions. The scroll bar block is only a rough indication of whereabouts the user is currently browsing in the video.

Slide Show browser (Single layer - Relative time - Temporal presentation)



The browsing interface automatically flips through each keyframe one by one on the screen, as in a slide show. While flipping the keyframes, a small timeline under the keyframe indicates the current point of keyframe in relation to the whole length. Temporally presenting a set of keyframes is quite a different method than the way most digital video systems present keyframes spatially.

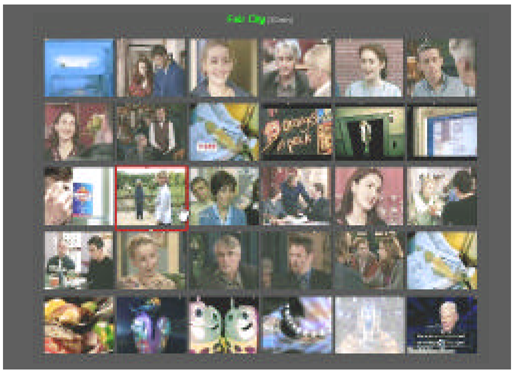
Timeline Bar browser (Single layer - Absolute & Relative time - Spatial presentation)

As introduced in Chapter 2, this browser presents keyframes with a segmented timeline bar at the top. Clicking any of the segmented units on the timeline will display a 24 keyframe page in that segment in time. Locating a mouse pointer on the timeline pops up a ToolTip box indicating the absolute time of that time region covered by the 24 keyframes currently on screen. While clicking any of the timeline bars directly brings the browser to the keyframes in that time, the left/right arrow buttons to the right of the timeline bar can be also used to sequentially browse from one unit to the next. In this way the user can concentrate on looking at the keyframes without having to reposition the mouse pointer to different parts of the timeline.

Overview/Detail browser (Multiple layer without link - Absolute & Relative time - Spatial presentation)

The double-layered timeline bar at the top is used to navigate 2 distinct sets of keyframes and different points within the video. Clicking on the short bar at the top will show a small set of keyframes selected from throughout the video content, providing the user with an overview of the entire video content. The lower bar is the timeline as in the Timeline Bar browser, with its segmented timeline showing detailed keyframes in that region of timeline.

Dynamic Overview browser (Multiple layer with link - Relative time - Spatial & Temporal presentation)



The screen displays spatially a small number of keyframes selected from throughout the video content, providing an overview. When a user brings the mouse cursor to any of these keyframes, the pointed one will start flipping through a number of adjacent and following keyframes in that segment, temporally presenting more detailed keyframes. While flipping through these keyframes, a small timeline appears below the keyframe indicating the current point of browsing.

Hierarchical browser (Multiple layer with link - No time - Spatial presentation)

The screen shows 6 keyframes selected roughly from throughout the video content at the top row of the screen. When a user brings the mouse cursor over any of these keyframes, a second set of 6 keyframes appear below showing a more detailed view of that pointed region. Bringing the mouse cursor over any of these keyframes shows another set of 6 keyframes below it. The user can move the mouse cursor quickly on the keyframes, while the browser reorganises and presents the hierarchically-arranged keyframes in a highly interactive way. This idea has been developed and implemented earlier in other systems [Mills *et al.* 92] [Zhang *et al.* 95].

FURTHER IMPROVEMENTS AND NEW BROWSER ADDITIONS

The design space we developed earlier has been useful for us in helping to identify and develop new browsing interfaces not covered by existing browsers and revealed to us as "gaps" in the design space. For example, within the space we can think of a new interface by dragging the "Scroll Bar Browser" inward along the *Temporal Orientation* axis to provide a greater amount of time information - this browser interface would be a Scroll Bar Browser variation with stronger time orientation for the user.

Also, we want to drag all the bottom user interfaces up to the top on the *Layeredness* axis by implementing additional layers to Scroll Bar Browser, Slide Show Browser, Timeline Bar Browser and Overview/Detail Browser. Figure 4.6 below shows some possible interesting browsers in the dimension space.

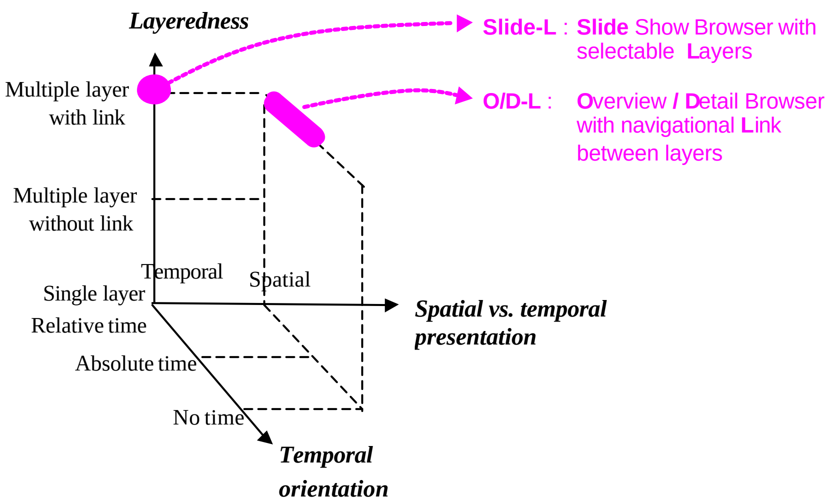


Figure 4.6 - Design space and some possible interfaces

4.4 Summary

In this chapter, the analysis of keyframe browsing interface design was conducted. The result was the identification of important dimensions of browsing interfaces and their possible values:

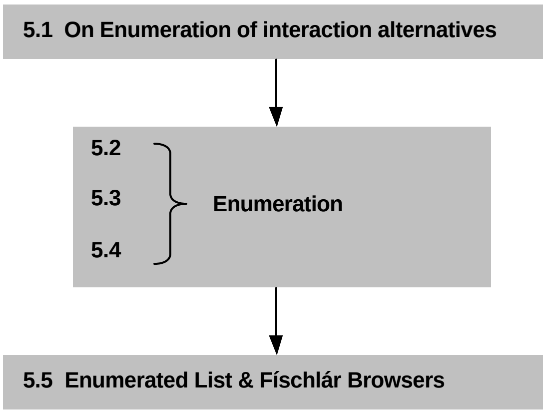
1. **Layeredness** dimension - single layer, multiple layer without navigational link, multiple layer with navigational link
2. **Temporal orientation** dimension - Absolute time, Relative time, No time
3. **Spatial vs. Temporal presentation** - Spatial, Temporal

After we had completed the development of this design space, these identified elements were then combined into a single 3-dimensional diagram, a design space, where each axis represents each dimension identified along with its values. This design space allowed us to come up with new interfaces by combining different set of values from each dimension, making the design process a simple, concrete decision-making process. From the design space, six distinctive browsers have been designed and implemented for the Físchlár system.

CHAPTER 5

Interacting with Keyframe-Based Video Browsers

This chapter enumerates the possible *interaction* alternatives for keyframe-based browsing interfaces to digital video systems. This process is based on the previous chapter's design space, by considering possible interaction styles and options for each of the dimension values. This completes the analysis of our structured listing of design alternatives both for the data presentation mode and the actual interaction at widget-level. Two new Físchlár browsers are designed and implemented to demonstrate the widget-level differences. Following that, a decision check list is drawn up to characterise our total of eight Físchlár browsers at the widget-level detail.



5.1 Enumeration in Interaction Alternatives

In Chapter 4 we constructed a design space where the most important elements of keyframe-based browsing interfaces could be specified. This construction of the design space came from analysing the characteristics of the data that is to be presented to the user in a video browsing system, namely a large number of temporally ordered images. Consequently, what the design space shows is the possible ways of data (i.e. keyframes) presentation in a browsing interface.

For a concrete user interface to be specified, however, *how exactly the user-computer interaction is to occur* has to be addressed, too. Even after the three dimensions' values are determined (for example, single layer - absolute time - spatial presentation), there are lots of further large and small design decisions to be made in terms of the interaction mechanisms such as the actual widgets to be used on the screen. Thus, there can be many different implementations coming from the same location within our design space. These 'other kinds' of design considerations are different from the data presentation variations dealt with in the previous chapter in the sense that it is more about actual interaction-level decisions, and still significantly affects the users' browsing behaviour and satisfaction, thus it must be considered carefully.

This chapter will elaborate in further detail the different interaction styles that can be selected for each of the design space dimension values and also another interaction element that is dimension-independent. The detailed widget-level options elaborated here are mainly based on a desktop computer with a GUI interface (such as Windows NT) environment where a large screen space is available and a mouse and possibly a keyboard is used for user interaction. In other words, this chapter starts restricting the possible design space by allowing us to think about specific widgets and styles mostly suitable for a modern GUI interface.

Having completed this elaboration means that we can pinpoint any possible keyframe-based browsers, and thus designing a browser becomes in effect a matter of being a two-stage selection process:

(i) select dimension values from the design space to specify the data presentation mode (from Chapter 4), then

(ii) select interaction elements for each dimension value to reify the design (from this chapter).

Throughout this chapter we present a detailed account of possible interaction elements, listed by their dimension values when applicable, and at the end of this we have an interaction style which is independent of any of the dimensions. At the end of this chapter, a tailored list of dimensions/interactions is used to characterise the six Físchlár browsers introduced in the previous chapter, and an additional two new browsers are presented to emphasise the interaction decisions introduced in this chapter.

5.2 Interaction for Layeredness

5.2.1 Single Layer

A single layered browser uses a set of keyframes extracted from video content, usually by taking one single keyframe from each shot. This is illustrated in Figure 5.1 below.

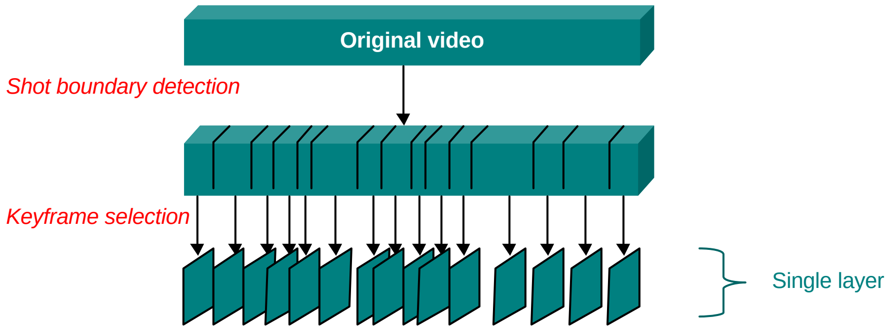


Figure 5.1 - A set of keyframes for single layer

Using this set of sequential, ordered keyframes, a single layer display is concerned mainly with how to present a large or a small set of such keyframes on the screen. The actual design can be in different forms, the main problem being that the screen size might not be large enough to display all keyframes at once. Roughly three configurations for actual design can be suggested:

1. showing keyframes one by one
2. showing many miniaturised keyframes on one screen and allowing continuous scrolling
3. showing many miniaturised keyframes on one screen and allowing pagenation to flip through them

The following are possible design strategies for these three configurations.

ONE BY ONE

Having a set of sequential keyframes, we can make a simple keyframe browser which has a square area in the middle to display a keyframe and a controller with which the user can flip through one keyframe to the next (Figure 5.2).

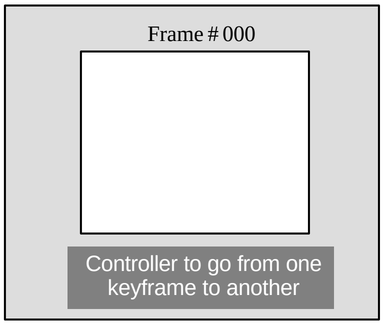


Figure 5.2 - One-by-one keyframe browser panel,
where variable controllers can be used

While it would become temporal presentation if the interface automatically changes a keyframe to the next in the sequence, when a user controls the changes, various control mechanisms can be applied, including the following possibilities:

- Arrow buttons to go to the next and previous keyframes. Some indicator that shows current keyframe number or time would be useful with this controller.

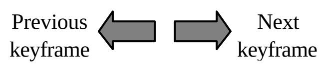


Figure 5.3 - Keyframe flipping buttons

- A keyframe bar which shows the position of the current keyframe within the whole keyframe set and allows dragging the current point, which will flip through various keyframes.

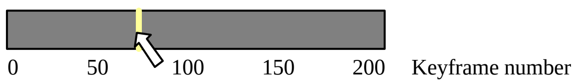


Figure 5.4 - Keyframe bar

- A timeline bar which is similar to the previous keyframe bar but which represents the proper time scale of the video and the current time point, rather than keyframe quantity.

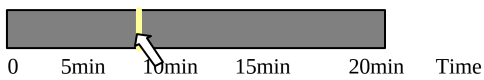


Figure 5.5 - Timeline bar

CONTINUOUS DISPLAY (WITH KEYFRAMES MINIATURISED)

By *miniaturising* keyframes into small thumbnail size images, it becomes possible to present many keyframes on a single screen - the basic idea of spatial presentation. The browsing task usually does not require high quality images with all the details in each image, and thus a set of thumbnail-size keyframes can be an efficient way of user browsing. What is required for this are some image quality studies for a specific screen setting, to find out an optimal keyframe size if we are to miniaturise these keyframes. If the number of keyframes is more than a screen can display at a time, some mechanism to show more keyframes has to be introduced, of which the simplest method would be to use a vertical or horizontal scroll bar (see Figure 5.6).

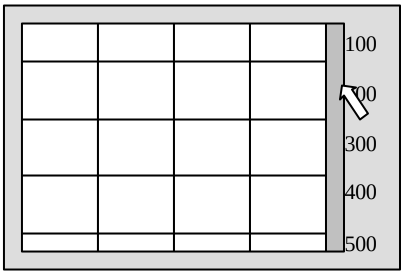


Figure 5.6 - Continuous display of miniaturised keyframes

The scroll bar's basic controlling property allows row-by-row scrolling, screen-by-screen scrolling, as well as immediate random access to any part of the scroll bar - considering its simplicity, a scroll bar is a very useful and pertinent controller in keyframe browsing. The standard scroll bar can be enhanced by displaying the exact range of the displayed point in relation to the whole length of the video (see Figure 5.7).

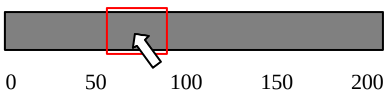


Figure 5.7 - Keyframe bar showing the range of displayed keyframes

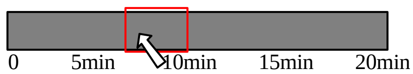


Figure 5.8 - Timeline bar showing the range of displayed keyframes

Modification of this scroll bar can be achieved by accommodating the time information in the timeline bar, which shows the current time point of keyframes displayed, rather than the quantity of keyframes displayed (Figure 5.8)

PAGENATED DISPLAY (WITH KEYFRAMES MINIATURISED)

We can segment the long keyframe list into page-sized units each of which can be displayed on a single screen, and we can provide a page flipping controller (see Figure 5.9).

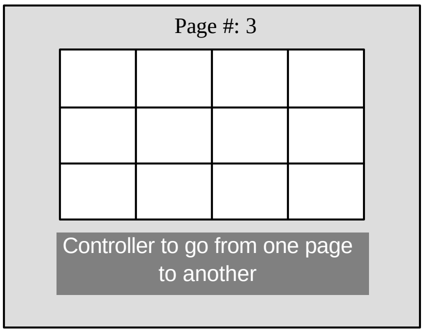


Figure 5.9 - Pagenated display

The main part of the screen is the panel where a pageful of miniaturised keyframes is displayed. At the bottom there is controller that allows page flipping - this controller is the same idea as used in the one-by-one keyframe browser above, and so either simply clicking the buttons (Figure 5.3), dragging the keyframe bar (Figure 5.7), or dragging the timeline bar (Figure 5.8).

5.2.2 Multiple Layer without the Navigational Link

The characteristic of the multiple layer interface is that it can display different quantities of keyframes based on the user's choice (Figure 5.10). Thus the data to be displayed consists of multiple sets of keyframes, intending to be presented and browsed separately.

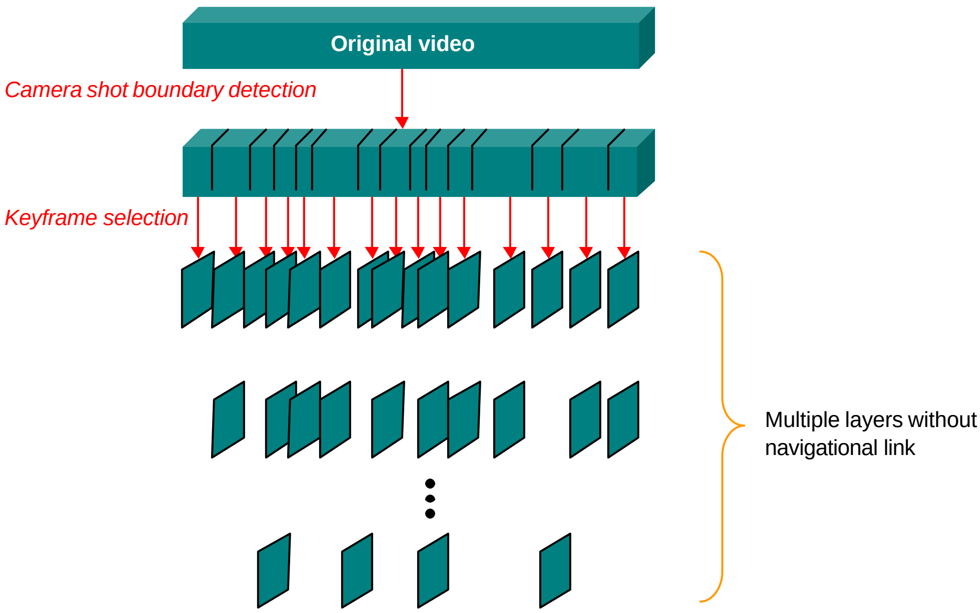


Figure 5.10 - Multiple sets of keyframe display without navigational link

Having multiple separate sets of keyframes for different granularity of content, there are two browsing methods such an interface has to provide:

- How to navigate within a layer (this is same as single layer interface)
- How to move from one layer to another (this is the main concern of this interface)

Designing for a multiple layer interface without navigational links among the layers disregards the user's possible direct navigation between the different layers. Generally this makes the design relatively simple, as each layer is a separate browsing space - each layer is almost like an independent interface. The idea behind this design is that once a user selects a layer, s/he will be *mainly browsing within that layer only*. Rather than emphasising the navigation between different layers in one browsing session, the importance is in the initial selection of the right granularity for the browsing task at hand, and from then on efficiently browsing within that selected layer. If the user changes the layer during a browsing session though, the current point of user browsing in the video is lost and beginning of the video in the newly selected layer is displayed (thus there is no link between layer navigation).

Initial selection of a layer for browsing can be simply hidden or set as a default, and thus this more complicated feature for navigating between different layers and making this understandable can be avoided. There can be different ways of multiple layer arrangement, but explicit or implicit layer selection for the *default layer* is the main idea.

We can make the most condensed layer (i.e. the smallest set of keyframes) as the default layer, with provision of further explicit layer selection (Figure 5.11). This allows initial viewing as an *overview* by showing the shortest summary of the video content.

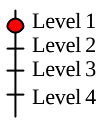


Figure 5.11 - Explicit layer selection,
with top-level as default

It would be desirable if a user can pre-set their default layer, in much the same way as the various options one can set in many software packages, or probably even better, it would be desirable if the browser could remember which layer the user has been at the last time s/he used the browser and the system should start the browser from that layer. Such a user-customisable interface is considered always preferable and much emphasised in all interface design literature today. In the other direction, it would be possible to let the system automatically set the default layer for the user - e.g. the automatic setting depends on the total size of the keyframe set, or perhaps an automatic personalised setting by observing the user's usage), which could become a starting point for an adaptive interface.

Another point that has to be made here is the notion of *atomic* and *composite* browser - multiple layers allow us to have more than one browsing interface. If a browser has multiple layers and in each layer uses presentation modes that are different from each other (i.e. two interfaces designed that occupy different locations within the design space), we cannot truly say that this is a single interface - it is two browsers in one, a composite browser which will make simple comparisons with other browsers more difficult.[1] Such a composite browser will probably have a synergetic effect for having different atomic browsers in different layers, in effect making it an interface with multiple browsers within it and each of them selectable by the user.

5.2.3 Multiple Layer with the Navigational Link

Browsing interfaces for multiple layers with navigational links provide the user with multiple *dependent* sets of keyframes (Figure 5.12). It has to provide the user with a way to navigate within a layer as well as from one layer to another in a more integrated manner than previous multiple layers without link interfaces have done .

[1] Informal discussion with Mc Donald, K. (October 2000)

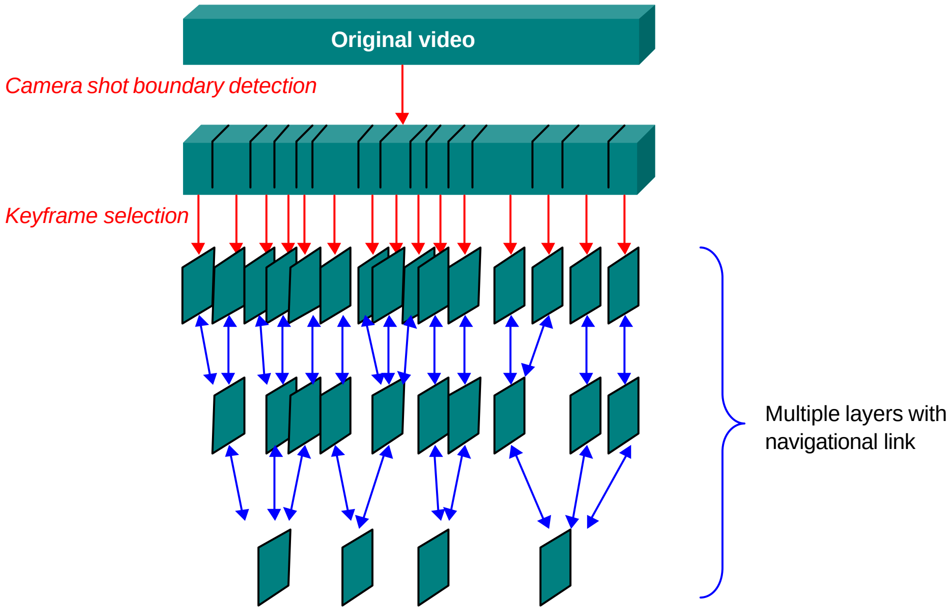


Figure 5.12 - Keyframes for multiple layer display with navigational links between keyframes

Browsing keyframes within a layer could be the same as browsing with a single layer, but it is more likely that it should be designed differently in order to integrate between-layer browsing seamlessly into within-layer browsing. Unlike multiple layer without links, navigating among different layers is as essential part of the browsing as navigating within a layer. By jumping up and down the hierarchical arrangement of keyframe sets, the user can browse the large amount of keyframes, *without losing the current point of browsing* within the video while simultaneously changing the layer.

The generally important thing in this design is that there has to be very careful thought and consideration given to the topic of proper visualisation of layer relationships and of the user's layer jumping actions, so that the user has a good idea where within the layers of keyframes s/he is in, whether going up or down the layer abstractions. Difficulty in design arises because screen space is limited and a number of keyframes from different layers have to be displayed together; and sudden changes of layers in what appears to the user as identical formats among the different layers can easily disorient the user. Our commonly agreed 'playback-action' command which is a left-mouse button click on a keyframe can easily be confused when the operation of changing the layer has to be done with a similar action on a keyframe such as a right-mouse button click. Good use of animation and subtle layout emphasis for keyframes and other screen objects can be useful to reduce these problems, and seems to require skillful interface crafting. Some possible designs alternatives are:

* Show the top-level layer on the screen, and on user request show the lower-level layer on top of it (Figure 5.13), as referred in [Boreczky *et al.* 00] as "drilling down." Although it would be ideal if the popped-up layer does not hide the background layer too much, the main idea is to visualise the fact that the popped-up layer is taken from one particular part or sub-section of the higher-level layer. Navigating among layers makes more sense and becomes more meaningful when a person can easily see where s/he is coming from and where s/he is going to and can also compare keyframes from both layers together, i.e., *showing details but preserving the context.* User actions for requesting lower-level keyframes could be by either clicking on the right mouse button (as the left mouse button is often set as a playback trigger), or a MouseOver action for quicker and more dynamic feedback. Also, it would be preferable to show some form of animation that indicates the upper keyframes popping up, so that the feeling of continuation can be kept.

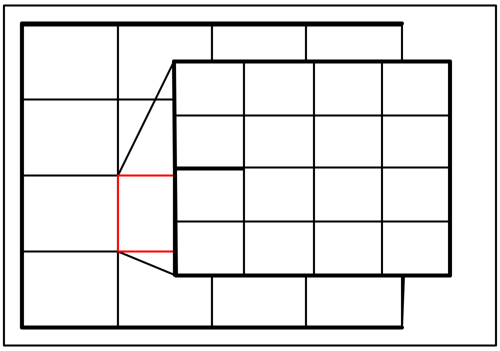


Figure 5.13 - Link to the keyframes on a lower-level layer

* Depending on how many keyframes there are on a layer, we can think of displaying them without imposing one layer on top of another (Figure 5.14), as done in [Mills *et al.* 92]. The user action for requesting lower-level keyframes could be by right mouse clicking or MouseOver on a keyframe, and going up to a higher level might need a separate button or a scroll bar, or simply bringing the mouse-cursor to the top of the screen (going down the layer needs specifying one of the keyframes, whereas going up need not). Ideally, while moving up and down the layers, the keyframes' expansion and collapse should show animation rather than suddenly replacing into other keyframes, thus providing a good idea of the user's direction of navigation. The animation may be so quick that it's not possible to see the keyframe contents - it's enough to show where it's moving to).

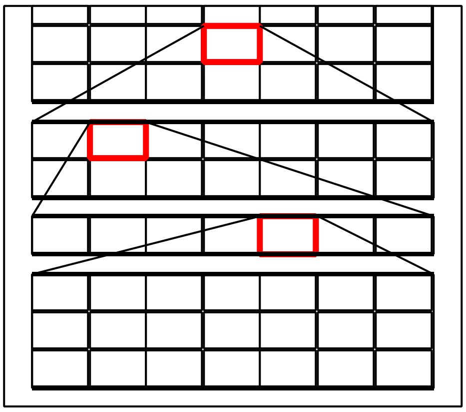


Figure 5.14 - Variable size layers and linking

- A mixture of spatial and temporal visualisation: one interesting way of multiple layer display is to display a set of top-level keyframes, then for each keyframe on that level a user-requested slot shows lower-level keyframes in the same slot temporally.

5.3 Interaction in Provision of Temporal Orientation

Providing time information on the interface can be done in many different ways, including timestamping each keyframe with its current time (absolute time) and visualising the current point in relation to the whole length of the video (relative time).

5.3.1 No Time Information

Video browsers do not necessarily require to show time information. This point was made in Chapter 4 in the discussion of temporal orientation that the provision of time information is something that has to be considered. The designer might deliberately omit time information from the browser. For example, when the nature of the browsing is more by a particular subject or topics such as in a news programme, the emphasis of the browsing is not about time and the user has not much concern about the whereabouts within the video that the current keyframe is drawn from, and in this case time information will not be of much importance to this user and might even clutter the screen and hinder more efficient interaction.

5.3.2 Absolute Time

Supporting user browsing by providing absolute time information (i.e., indicating that the "current keyframe is taken from 11min 26sec into the video") can be designed in different ways, but basically it is done by showing the numeric form of time on the browsing screen somewhere. The simplest way to do this is to timestamp each keyframe, as is often found in many commercial video editing systems (see Figure 5.15). Again, it would be better to allow the user to turn the time information on and off as an option.

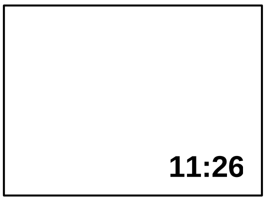


Figure 5.15 -
Timestamping a keyframe

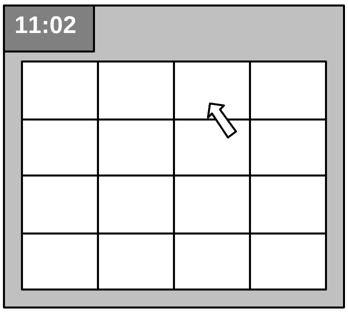


Figure 5.16 - A separate, dedicated location showing time as mouse is over a keyframe

The actual timing information can be displayed as in Fgures 5.15 or it can be displayed inside the global window (but outside a keyframe lest it distracts from the keyframe content) on a fixed position (Figure 5.16). As a user moves the mouse cursor over each keyframe, the time information changes reflecting the time for that keyframe pointed to.

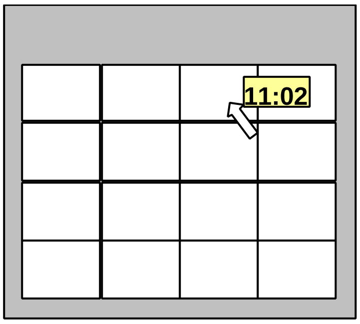


Figure 5.17 - ToolTip box popping up to show time information as mouse is over a keyframe

Another option for including absolute temporal information is to display without occupying a fixed position on the screen but by using a 'floating' ToolTip box when the mouse cursor is on a keyframe (Figure 5.17). This box appears when the mouse cursor is on a keyframe and the user stops any movement for about 0.5 seconds. This can be suitable for the browsers where time orientation is not a prime concern. As tooltips are floating over the screen, they do not clutter keyframe content, while still basically providing time information. This is not much different from the above options, except that the time information is less immediately displayed at the benefit of saving screen space.

5.3.3 Relative Time within the Video

This mode of interaction is about visualising the current viewing position in the video in relation to the whole length of the video. To indicate the current point relative to the whole, a timeline is mostly used in some form or another, which looks similar to those timeline bars used in video player software (Figure 5.18).

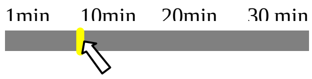


Figure 5.18 - Timeline bar showing the current point of browsing in video

Note that the timeline bar can be also used as controller as well as an indicator of current time. This is a good example of tightly-coupled input and output [Ahlberg & Shneiderman 94],

This mode can also be augmented by indicating where in the video the keyframes have been extracted from as well as where in the video the current browsing point is (see Figure 5.19). Using this same timeline bar for multiple layer browsing would also be a good way of visualising the current granularity of the layer. Figure 5.20 shows an example of such a timeline bar which shows the location of extracted keyframes (and roughly the amount of keyframes) whenever the layer changes, white lines indicating the keyframe positions and yellow line indicating the current browsing point of the user.

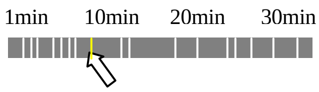


Figure 5.19 - Timeline bar showing points where keyframes are selected from and the current point

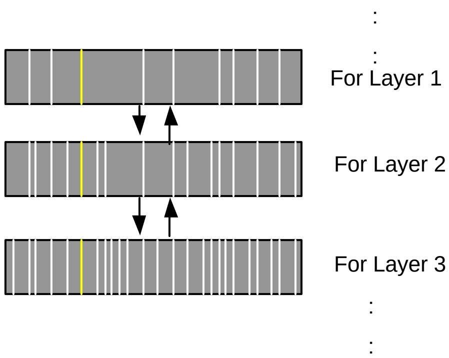


Figure 5.20 - Timeline bar with keyframe extraction point as a multilayer visualisation

5.4 Interaction in Spatial vs. Temporal Presentation

5.4.1 Spatial Presentation

Most of the above design alternatives have been in the category of *spatial presentation* in the sense that the keyframes are miniaturised so that many of them can be displayed on a single screen. The process of browsing in itself can benefit from the situation when many items of concern are spatially presented in one screen so the user can scan many items next to each other.

An obvious consideration in the spatial presentation of keyframes is the trade-off between the size of each keyframe and the number of keyframes that can be displayed at one screen (see Figure 5.21 and 5.22). Consideration of this matter should consider the purpose of the system or purpose of the user's browsing - for example, someone looking for a movie character crying will need a higher keyframe quality and thus a bigger keyframe size than someone trying to simply get a rough gist of the whole video content. Also, the size and quality of a miniaturised keyframes should be examined in the context of a specific hardware and software configuration (the screen size, resolution and colour depth, brightness and contrast, and so on). Again, it would be a useful feature if the user could customise the size of keyframes.

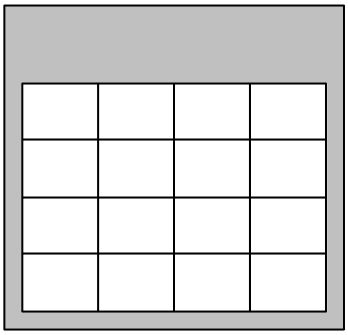


Figure 5.21 - Sixteen keyframes on a single screen

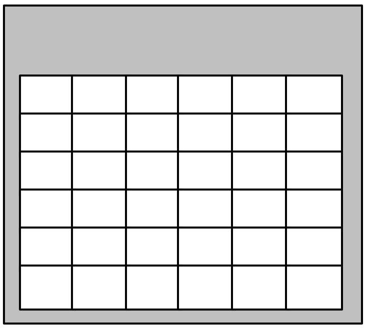


Figure 5.22 - Thirty-six keyframes on a single screen

5.4.2 Temporal Presentation

Considering the fact that the keyframes have a temporal ordering, we can present keyframes one by one as in a slide show style (for detailed analysis, see Chapter 4). Saving screen space has been noted as providing an advantage over most spatial presentation alternatives [Komlodi & Marchionini 98]. The basic screen layout for this design would be a square or rectangular area where keyframes can be displayed and automatically flipped to the next (see Figure 5.23).

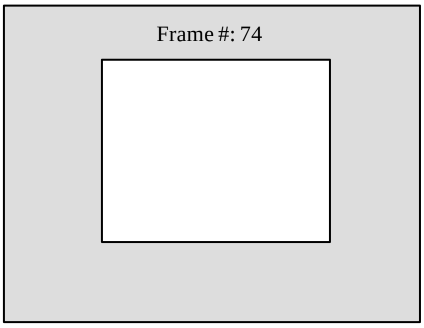


Figure 5.23 - Temporal presentation panel

Different kinds of controllers and/or indicators (including the elements mentioned above, including time indicators such as timestamping or a timeline bar, and various layer navigators) can go with this interface, in order to aid user viewing. The simplest but still a quite necessary one is a bar indicating the current position of the keyframe in the video. The bar may indicate the keyframe number (Figure 5.24) or the absolute time (Figure 5.25).

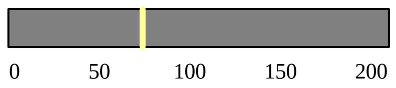


Figure 5.24 - Temporal presentation with a bar based on keyframe number

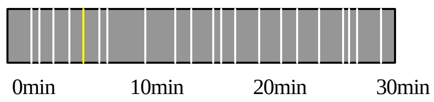


Figure 5.25 - Temporal presentation with a bar based on time

While keyframes are temporally presented to the user, it would be a bonus if the user could pause/resume the flipping for more careful examination of keyframes. Going further, it would be beneficial if s/he *can control the pace of the speed of keyframe flipping,* as different individuals will have different capabilities and styles in perceiving and observing images as they are presented. A controller that can modify the flipping speed could be like the one shown in Figure 5.26 below. This way, we can imagine a user setting the speed as fast when the keyframes are not interesting to him/her, and reducing the speed when interesting ones are presented.

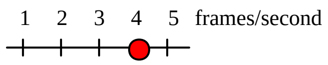


Figure 5.26 - Keyframe flipping pace controller

When we say that the "design space is huge" we mean that there can be many different ways of finalising an interface, and we do not only mean the various positions one can locate within the design space by deciding values for each dimension, but we also mean the various ways to reify that design decision to an actual, visible interface by considering the interaction levels of screens and widgets. This section listed and suggested some of the possible ways of these specific alternatives for each of the dimension values, based on currently perceivable widget styles.

5.5 General Interaction Style: Static vs. Dynamic

The interaction alternatives we have considered so far are dimension-dependent alternatives, i.e. options that are only applicable when the dimension level option has been decided. For example, the particular interaction alternative we call the 'ToolTip box' becomes our concern only when the 'Absolute time' option is chosen from the Temporal orientation dimension. On the other hand, there are interaction alternatives not dependent on any dimension values. Here the one identified is the interaction style in terms of its degree of interactiveness.

It is possible to imagine two similar interfaces (for example two interfaces designed at the same location in our dimension space) having different degrees of interactiveness. A prime example is an interface's mouse event: whether the interface recognises the user's mouse click on an object or a mouse pointer positioned over an object as his/her definite action makes a huge different in the overall interaction session. The timeline bars and other controllers introduced so far can be programmed in two different ways in this respect.

To bring down the continuous degree of interactiveness into discrete choices, the notion of *Static* and *Dynamic* interaction styles have been introduced at this stage:

A static interaction style is where the user has to initiate a definite action such as a mouse clicking in order to trigger a system reaction, and without such user-initiated actions there will be no visible reaction from the system. A dynamic interaction style is where the user's action is sensitively and continuously fed back as system reaction.

The degree of interactiveness in a user interface will not generally be a binary value such as Static or Dynamic but will actually be continuous, and might benefit from further elaboration in the future. Here the notion was mainly introduced to allow us to experiment with the same interface (i.e. interfaces occupying the same location in our design space) with different style widgets that respond differently on users' actions. The focus of this study is to experiment and clarify ideas rather than trying to capture as reallistically as possible the human-computer interaction attributes. As with some of the other characteristics that will be considered in this thesis such as user preferences and tasks, a more detailed and precise element-capturing process will require more input from other disciplines such as psychology, thus openning up the scale of this study too much.

5.6 Eight Físchlár Browsers Viewed from the Design Alternative List

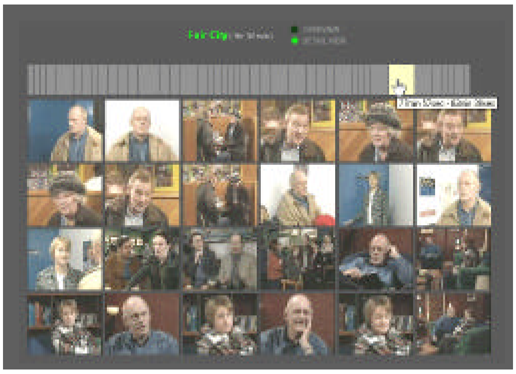
In this thesis, Chapter 4 presented an analysis of the significant dimensions of keyframe-based browsing interfaces and identifyed possible values for each dimension, thus constructing a finite design space where we can determine a distinctive interface. This chapter so far has further identified and detailed a list of possible interaction alternatives for each of the design options in our dimension space as well as one additional independent alternative on the degree of interactiveness. Having identified and itemised all these elements, one good way of thinking about these design alternatives is to draw a check list of all the alternatives, and then we can characterise any keyframe-based browsing interface in terms of this itemised design check list.

Before drawing up the list, we introduce two more Físchlár browsing interfaces in addition to the six presented in Chapter 4. The purpose of these two new browser interfaces is to experiment with different interfaces which have identical locations in the design space, but have different interaction styles, and this will allow us to conduct experiments on the interaction style independently of browser interface.

Quick Bar browser (Single layer - Absolute & Relative time - Spatial presentation)

This is essentially the same as the Timeline Bar browser, with keyframes spatially presented in a single layer, its timeline indicating relative time, while pointing the mouse over the bar will pop up a ToolTip box indicating the absolute time. This interface has the same location in the design space as the Timeline Bar browser and the only difference is the behaviour of the timeline: as the user moves the mouse cursor over the timeline bar, the keyframe page immediately changes reflecting the corresponding time, rather than having mouse clicking to trigger this change as in the Timeline Bar browser. This makes this new browser interface very reactive to the user's mouse movement commands.

Overview/Detail + Quick browser (Multiple layer without link - Absolute & Relative time - Spatial presentation)



This browser interface is the same as the Overview/Detail browser, with keyframes spatially presented in a double layer without a navigational link, and both relative and absolute time are provided. The difference with this new browser is that the detailed view behaves the same as the Quick Bar browser described above.

As can be seen, these two new browsers are the Dynamic equivalent of the Timeline Bar browser and the Overview/Detail browser. Having exactly the same location in the design space, the main differences introduced with these new browsers is that they are Dynamic in their interaction. These two additional interfaces have been designed and implemented on the Físchlár system, along with the other six interfaces. Because Físchlár is a network-based system, making an interface Dynamic caused us problems with loading speed as a large quantity of keyframes must be pre-loaded into a web browser in order to be able to show up as quick as the user moves the mouse cursor over a timeline bar. An instant sweeping of the cursor over the bar should display all the keyframes very quickly without having to wait for network latency. Implementing this kind of interaction required loading all the keyframes to the local computer's memory when the browser interface is launched, which could take considerable time depending on the amount of keyframes for the video being viewed. Once all the keyframes are loaded, however, the immediate reaction is well demonstrated. In spite of this loading speed problem, these two new browsers have been used with other six browsers and from a practical usage point of view there were some problems caused by the time taken to load them with all keyframe images, but from the experimental point of view they are still valid and useful in testing the analysis. Since the two browsers have been implemented, all the following discussions and evaluations were conducted with the total of eight Físchlár browsers.

Table 5.1 below is one way of summarising the eight Físchlár browsers in terms of this design alternative list.

Table 5.1 - Físchlár browsers and their design alternatives taken

	Layeredness					Temporal orientation						Spatial vs. Temporal			Presentation style	
			Multiple layer with link				Absolute time			Relative time			Temporal			
	Single layer	Multiple layer without link	Layers overlap	Layers don't overlap	Mixed spatial and temporal	No time	Timestamp in keyframe	Dedicated location	Tooltip box	Timeline bar	Timeline bar with selection indication	Spatial	Constant flipping speeds	Flipping speed controllable	Static	Dynamic
Scroll Bar	✓					✓						✓			✓*	
Slide Show	✓									✓			✓		✓	
Timeline Bar	✓								✓	✓		✓			✓	
Quick Bar	✓								✓	✓		✓				✓
Overview/Detail		✓							✓	✓		✓			✓	
O/D + Quick		✓							✓	✓		✓				✓**
Dynamic					✓					✓		✓	✓			✓
Hierarchical				✓		✓						✓				✓

* Scroll Bar can be both highly interactive as well as static - dragging the square block is a highly interactive experience, whereas clicking on the arrows is static experience. But here it is ticked as static, because if both is possible it is usually good for users with static preferences which will be further explained later in Chapter 6.
** O/D + Quick is strickly speaking half static and half dynamic - the overview screen can be considered completely static as normal O/D screen. But because the strengh of this browser is its Detail view and distinction between this browser and O/D is in its Detail view, we put more weight to the Detail view characteristic, which is Dynamic.

The table shows a clear and concise view of which sets of alternatives each browsing interface has taken and incorporated. A new interface can be designed by simply adding another row in the table and choosing some of the design alternatives.

This chapter completed the consideration of all the possible design options in terms of keyframe presentation methods and interaction alternatives. The process of constructing the detailed checklist and underlying design space was mostly based on the review of available and thinkable systems and general design heuristics. How valid and sound this analysis result is has yet to be answered.

Thinking about this analysis result's validity and soundness is, however, a very subtle and fine-level consideration. The validity of the constructed design space should be understood in terms of whether it truly represents all keyframe-based browsing interface possibilities and covers all possible features that are important in user-browser interaction and whether, as a result, it can be usefully applied in the real world. It would be difficult enough to check to see how truly it represents all possibilities and picks up important attributes of video browsing. The design space was constructed out of the need to be able to consider more comprehensive and fuller possibilities in designing browsing interfaces in the first place, and these fuller possibilities have not been realised anywhere in literature before. However, it is possible to roughly gauge its validity by applying it to conceivable application areas and see how well the design results are achieved by using the design space. Usefulness partly implies validity - if it is constructed properly, it should be able to be applied and used well. Partly the usefulness has already been demonstrated with the design of the various Físchlár browsing interfaces, which were designed based on the constructed design space and alternative interaction enumeration. It was already mentioned that designing a new, different browsing interface is a matter of locating a position in the design space or ticking the feature list in the above checklist table. Being able to see and compare the browsing interfaces from a clearly itemised feature list as in Table 5.1 above is in itself an important benefit and usefulness of this design analysis result. In Chapter 6, more focused applications of the design space will be dealt with to illustrate the usefulness of the design space. Particular target environments will be set and the design space used to base the selection of specific interfaces suitable for those environments.

Another important way to check the validity of our design space is to brainstorm new designs or new elements in browsing interfaces in a completely intuitive way, and then see whether they can be pointed out and addressed from the view point of the design space. In Chapter 7, a user-oriented evaluation is conducted using the 8 Físchlár browsers, to brainstorm and collect ideas and aspects of the browsing interface from people. This will allow us to see how well real users' ideas (which are not affected by our design space structure and framework) can be understood in terms of the design space structure and framework. Would there be points or features voiced from real users that our design space is unable to address or specify? Would there be much more emphasis in real users' ideas on a particular attribute of the design space that might benefit further elaborated analysis? Surely users' ideas cannot be perfect and address all important points, but it will be important to try and relate real users' ideas to the theoretically constructed design space. These will be used as a way of checking the validity of the constructed design space so far. Thus, the following two chapters (Chapter 6 and 7) are intended to illustrate the usefulness and validity of the constructed design space.

In summary, the thesis that we present here in this dissertation is that the design space which we have constructed earlier, is reasonably complete and comprehensive and will be validated by the construction of as many as eight different browser interfaces, and their qualitative evaluation by a set of real users. We will return to the question of whether our thesis is validated in the conclusing chapter.

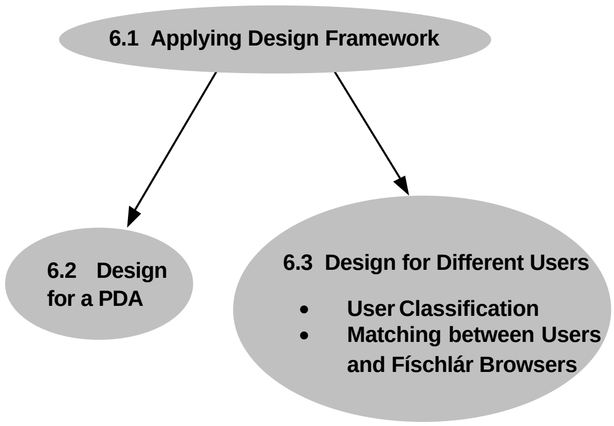
5.7 Summary

In analysing our design space so that we can specify a particular interface from this space, there has to be a detailed widget-level alternative list from which a specific design can emerge. Continued from the previous chapter's work on the basic design space that identifies distinctive modes for keyframe presentation, this chapter further elaborated this by listing a set of possible interaction styles at the concrete widget-level, thus forming a detailed check list of design alternatives for keyframe-based browsing interfaces. By putting in this list any keyframe-based browsers we wish to, it is easy to see what characterises that browser has compared to another. The power of the analytic design methodology is this very benefit of being able to compare any available browsers under the same check list, and the creation of a new style browser becomes a matter of choices from this set of design alternatives.

CHAPTER 6

Context, Users, and Video Browsers

This chapter is about the choice of a specific design among various possible design alternatives so far covered in the thesis. Of all the possible designs, selecting a desirable one for a particular target environment and context is done through reasoning about a set of constraints to be applied to the design space. In this chapter, two different kinds of such constraints have been considered: First is a platform concern where a video browsing interface for a small, handheld PDA is considered; the second is the user concern where different user preferences work as constraining factors in selecting suitable video browsers from the design space.



6.1 Context of Video Browser Usage

The previous two chapters have seen us construct a design space and a detailed design alternative list for digital video browsers whereby many different keyframe-based video browsing interfaces can, in theory, be specified. The power of analytic design is this very quality that allows us to devise with numerous different interface designs from within the one overall structure. At the end of Chapter 5 it was demonstrated that any existing browsing interface for digital video based on keyframes can be characterised simply by checking the list of analysed elements of the browsing interface as the dimensions or elements of this design space cover the major aspects that differentiate browser interfaces from each other.

Our design space and our detailed alternative list, however, does not say which of these numerous alternative combinations have to be chosen for an actual interface implementation and its eventual usage. From the developer's point of view, it is a great advantage to be able to see and consider many different possible designs without requiring some kind of new perspective or special creativity, but having multiple possibilities in itself is not enough as the designer faces the hard question of having to choose a particular design from the numerous possible design alternatives.

Naturally following the construction of the design space and the widget-level interaction alternative list, one is now able *to choose one specific design* that is most suitable for a specific target context where a browsing interface is to be used. This concern covers a wide area, involving the target users' personal, social and cultural context as well as the particular machine and platform where the interface will be running. The best user interface is the one that is most focused and suitable for a specific user in a specific environment: at the back of all the user-oriented development, usability engineering and participatory design processes that have been so much emphasised and practised in recent years [Nielsen 93] [Smith 97] [Dumas & Redish 99], is the idea of the design iteratively getting nearer and nearer to the needs of the specific target users and their environment. Unless we focus on a specific situation where the target system is to be used, it is impossible to confidently say anything about a system's usefulness, its usability and its success. That is, we cannot judge or evaluate an implemented system's usability without bringing in the usage context. This is all very reasonable and in a sense obvious where we reason about systems which have been developed and implemented and where these form the starting point for the evaluation we wish to carry out. In the case presented in this thesis we do not start with a system and its interface but we commence the process of building an interface for a user with a design space and from that we develop an actual instantiation which turns into an actual interface. This makes our case a little different to the classic user-oriented development and so, from the perspective of the design space, having a specific target context can be viewed as the process of *applying a set of constraints to the constructed design space*, thus reducing the possible combinations of alternatives and eventually settling down with one particular design, suitable for a specific environment. This is illustrated in the Figure 6.1 below.

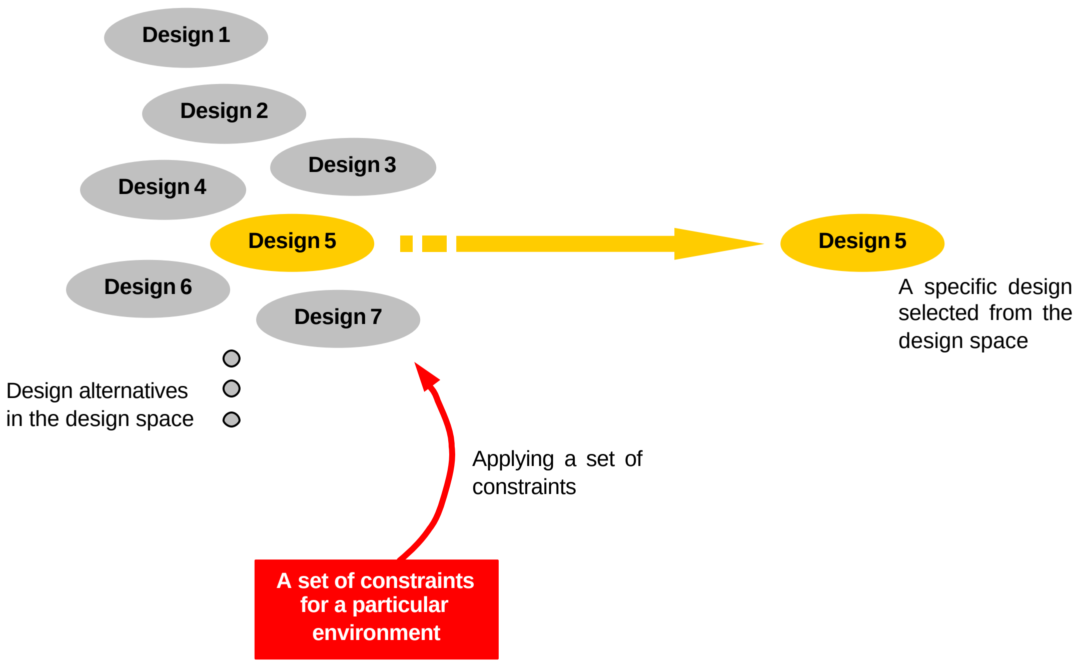


Figure 6.1 - Applying constraints to a large design space to select one specific design

Moving further along this line of reasoning, even with a specific target environment there are different users who may want to achieve the same goal but in different ways. Thus a single, focused user group for which a system is to be designed, can diverge still further. People have different aptitudes, attitudes and preferences when using the same system to complete the same task in the same environment. This means that when the system developer works on refining the system and attempts to optimise system features for a very specific environment, there comes a point where the designer starts favouring some of the target users' wishes at the cost of ignoring others. At this point, any attempt to consider one user's wish can result in a complaint from another user. In this sense, even in most of the user-centred system development projects that are currently considered desirable, individual users' differences are largely ignored by attempting to fit in everybody into a single interface with a single set of features. *It is not possible to build a single user interface that can satisfy the needs of everybody*. How specific can we target when developing a specific user interface? We can consider providing several versions of an interface that are most suitable for individual users or groups of individuals having similar characteristics. This would involve identifying individual users' characteristics and matching between a specific interface design and a specific user, as illustrated in Figure 6.2 below.

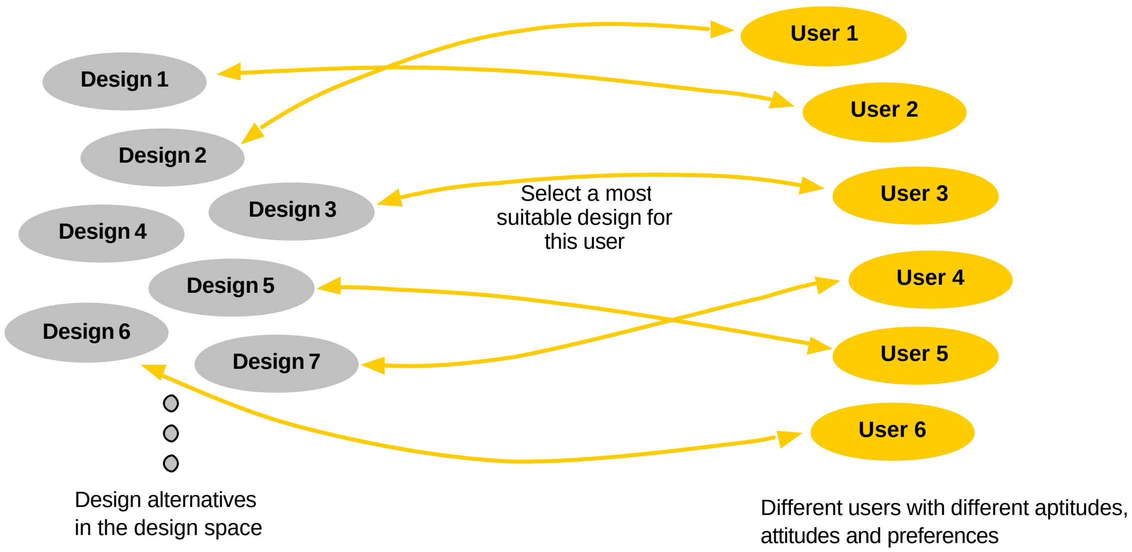


Figure 6.2 - Matching between a user and a specific design in design space

In this chapter, the usage contexts for users browsing digital video via sets of keyframes are considered in depth, they are then applied to the design space and interaction alternatives are constructed as outlined in the previous chapters. Two different cases have been investigated. In the first case, a simple process of selecting a specific design from the design space is considered by setting a usage context for video browsing in a mobile environment on a PDA (Personal Digital Assistant). In doing this, we consider a set of constraints to be applied in a PDA interface and we select suitable design elements from our design "inventory", thus coming up with a few specific design examples. This idea is outlined in Figure 6.1 above. In the second case, we classify users by identifying several important parameters that affect video browsing behaviour significantly, and we identify a set of 8 different *user types* which are then matched against the 8 Físchlár video browsers presented earlier in terms of which suits best. This idea is outlined in Figure 6.2 above.

6.2 Device Concerns as Design Constraints - Video Browser for a PDA

6.2.1 Platform and Device Diversification

The current Físchlár system is mainly targeted for the normal web browser users with a desktop PC. PCs sitting on an office desk or more recently in a domestic environment have been the dominant form of computer usage for quite some time. However, recently emerging and growing in popularity are the mobile devices such as the PDA and sophisticated mobile phones equipped with various functionality including contact organising, word processing, spreadsheet, email and web browsing. The need for data sharing among different devices has become more and more important. As mobile devices are becoming more and more powerful, we see the emergence of the idea that the actual device one is using can be just one of many different devices to access the same underlying system or database - PC, PDA, mobile phone, information kiosk, or even wrist watches are now converging to have the same level of computing power and connectivity, all serving similar functionality (email, calculation, web browsing, and so on) and all suitable for different environments (in the office, in a bus, on a street, in bed, and so on). The video browsing application is also such a functionality that could greatly benefit from device diversification moving out from just watching broadcast TV programmes and video on a desktop PC, into many other everyday situations such as waiting for a bus on the street, in an airport, in an airplane, and so on. More powerful but smaller hardware and higher bandwidth network are starting to penetrate into our everyday lives, everywhere and at any time.

6.2.2 Mobile Technology for Video and their User Interfaces

The mobile communications market is bringing together GPS systems for navigation, palmtop computing and wireless voice/data access using GSM, GPRS, EDGE and UMTS. Current GSM technology allows 9600 BPS bandwidth, enough for SMS-messaging and this current wireless mobile internet access is not meant to be full surfing but to allow pinpoint access to critical amounts of online data whilst "on the move." GPRS which is about to emerge from the laboratories, will scale that bandwidth up to 115 KBPS and will have the advantage of "always on." EDGE is another technology which will scale this up further. The big impact will happen with the arrival of the "third generation" phones using UMTS technology sometime in 2001/2002. This is supposed to allow bandwidth of up to 2 MBPS but even if this is not achieved there would be sufficient bandwidth to allow streaming of low quality video material. Currently there is no single integrated box which combines GPS navigation, wireless voice/data communications and palmtop computing, but this will change shortly.

With the growing technical feasibility in mobile communications, the user interfaces for mobile devices have to be considered. As many conventional software systems for desktop PCs are ported to mobile devices such as PDAs and mobile phones, the user interface concern about the software on these devices are becoming an important issue. This is primarily because these mobile devices have physical input/output elements that are different (and usually more limited) to those of desktop PCs. The CHI (Computer-Human Interaction) community is feeling responsibility for the development of a new interaction paradigm [Marcus *et al.* 98] which is not the current direct manipulation metaphor mainly suitable for desktop PCs. All major three operating systems for the currently available PDAs (EPOC, Palm OS, and Windows CE) use the direct manipulation style with the main emphasis on the visual cue. In this style of interface, data and administrative elements are visually represented on the screen and dragging, pointing or selecting using a pointing device allows user actions with instant and continuous visual feedback. Continuous visual feedback is one of the great advantages for the desktop PC environment where the user usually concentrates on the display at all times - continuous visual feedback is one of the main factors that have made current desktop interfaces so usable. On a desktop PC environment, the user keeps on looking at the screen while information is displayed and modified. Based on this environment, experienced designers and researchers have been coming up with design guidelines that help design better interfaces in all different levels and significance: what a layout should be like, what control styles should be there, how animation can be used in a subtle way to get user's attention, what colour combinations should be used, and so on. All these guidelines are the tools on the base interface platform of desktop PC with the direct manipulation paradigm, with the user looking at the screen at all times and hands ready for keyboard input or mouse movement. However, these accumulated guidelines for desktop environments we have developed so far could be unsuitable for mobile devices - in the case of mobile and distractive environment with a small screen, selecting one of the multiple items on the screen which requires concentration and constant visual attention could be a negative factor in interaction [Kiristoffersen & Ljungberg 99].

Thus what we need to develop is a different interaction suitable for small-screen devices in a mobile environment, probably with much more emphasis on aural feedback (often mentioned in the literature as potentially important, see [Hindus *et al.* 95]), more tactile input (probably with real buttons that can be pressed by feel rather than "virtual buttons" on the screen [Myers *et al.* 00]), and with less emphasis on visual feedback. An interaction style where less various choices are available at one time and instead allowing a very limited number of widgets more frequently and repeatedly might be a good direction for mobile devices.

6.2.3 Design for a PDA

ABOUT THE REVO

The Revo is a handheld mobile computing device with a keyboard (see Figure 6.3) produced by Psion and it costs about US$400. Most of its competitors on the market such as the HP Jordana, the NEC Mobile Pro, the IBM Workpad or the Compaq C series run Windows CE and have a colour screen but are

Figure 6.3 - Psion's Revo

more than twice the price of the Revo. Revo is typical of the current generation of palmtop mobiles in that, like a PDA, it has the standard issue applications such as an address book, calendar, to-do list and memo pad but in addition it has a word processor and a spreadsheet application. It is designed to replace the laptop for simple tasks such as reading email or writing anywhere.

The Revo has a 32-bit RISC-based ARM 710 CPU with 8MB memory, a speaker, an RS232 docking connector and an IrDA-compliant connector allowing short range wireless connections and it runs an operating system called EPOC. The word processor and spreadsheet are compatible with Windows applications. A user navigates using either the small built-in keyboard or the touch sensitive screen using either a stylus or a finger. Revo has a web browser and via the IrDA port can be connected to the net. Because the screen is only 480 by 160 pixels and 16 shades of grey rather than colour, it is best suited to viewing web pages which have been designed and formatted for handheld mobile devices. Many of Revo's software and its general interaction mechanisms have a considerate design rationale, with sound design principles behind them (see [Healey 00]).

DESIGN OF REVO VIDEO BROWSERS FOR THE FÍSCHLÁR SYSTEM

The advantages of a truly mobile interactive interface to the Físchlár system would be enormous for the user as it would allow easier control of recording as well as lookup of the archive of recorded programmes, and it would enable not only alerting but mobile browsing and playback of content. The downside of this would be that we would be trading desktop screen quality for mobile interactivity, suggesting many limitations: the Revo's limited Memory size, its overall slower response time and image display rate, and its small gray-scale screen - these are some of the factors that limit possible design alternatives for browser design on a Revo. Also, the web browser provided on the Revo has been used in our work for the video browsing interface, which again limits the possible interaction styles. Informed by the design framework developed in previous chapters, some suitable browser interfaces for Revo were designed. Of various design possibilities that can be derived from the framework, Revo's particular circumstances and environment dictate which dimension options and which instances should be favoured, of which the following exemplary designs emerged.

The first design example is in Figure 6.4 below. The small and grey-scale screen makes a spatial presentation of keyframes unsuitable, though possible. Temporally presenting keyframes allows displaying each keyframe as large as the screen allows for better recognition. The remaining space on the right is used for displaying the list of indexed TV programmes (much the same as in the desktop version of the Físchlár system introduced in Chapter 3), from which the user can select to launch, the keyframes extracted from that programme being displayed on the left.

Figure 6.4 - Revo video browser design: Single layer / Relative time / Temporal

The timeline bar at the bottom of the keyframe indicates the current point of browsing. Tapping on the arrows at the either end of the timeline bar displays the previous/next keyframe. Tapping any of the timeline bars immediately jumps to the keyframe corresponding to that time. Each segment of the timeline bar might contain more than one keyframe as the number of keyframes can be more than the number of segments shown on the timeline bar. In its current design, using a finger or stylus to tap on the arrows or a timeline bar segment is the only way to flip through keyframes, but ideally, automatically "slide showing" the keyframes would reduce user interaction, while also allowing tapping to directly access any point in the video. Also, ideally, the timeline bar could be overlapped *inside* the keyframe, in a semi-transparent style, thus allowing the actual keyframe size to be slightly larger while both keyframe contents and the timeline bar are visible at the same time - this would require more precise experimentation as to recognisability in semi-transparency screens, as is reported in some other studies [Kamba *et al.* 96] [Harrison & Vicente 96]. Note that most controller widgets such as the scroll bar and timeline bar are located at the borders of the screen, preventing the user's interaction from getting in the way while viewing the screen.

In the second example, multiple layers of keyframes are provided so that the user can use different keyframe sets for browsing (Figure 6.5).

Figure 6.5 - Revo video browser design:
Multiple layer with link /Absolute & Relative time / Temporal

Keyframes are again presented temporally, thus being able to make it as large as the screen allows. A centred keyframe allows the remaining space on the left and right for widget buttons that can be pressed by the user's thumbs while holding the Revo firmly with both hands. This interaction style is similar to that of using some popular video games' controller pads. The pair of buttons on the left controls the flipping of keyframes, forward and backward, with relative time provided in the form of the timeline bar, and absolute time in the form of the time-stamp above it. The pair of buttons on the right controls jumping between different layers of keyframes, indicated by the vertical layer indicator, the current layer name and the layer size (i.e. number of keyframes) indicated above the timeline bar. Bringing the layer up results in a smaller keyframe set to be flipped through, while bringing the layer setting down displays a larger numbers of keyframes to be flipped through. Changing the layer does not affect the current point of the timeline bar on the left, meaning that jumping between different layers still maintains the current point of the user's browsing. Using these two pairs of buttons with thumbs, one can imagine a user moving up and down the layer with the right thumb while flipping through keyframes with the left thumb, in a highly interactive way. Ideally two pairs of keys (such as 1, 2 and 0, P) on Revo's keyboard could be assigned to do the same functions as the buttons on the screen in order to enhance tactile feedback and the feel of pressing real buttons. This way the virtual buttons on the screen can be removed and the remaining space can be used for displaying other information such as available programme listings. The Revo provides a sound setting where a small tick sound can be heard on every keyboard or screen touch input for feedback, which can be useful in the browser interaction. Many limitations faced here due to the Revo's current technical capability are expected to become less of an implementation barrier, as PDAs are fast improving in their hardware capabilities.

Attempts to port desktop capabilities to a mobile device such as a PDA are challenging both in terms of technology and usability. Described in this section is the work on porting Físchlár's video browsing capability to Revo PDA's small screen by redesigning its browsing user-interface to suit the Revo's particular screen specification and interaction methods. The design space was used to initially come up with various design possibilities, we then selected appropriate ones and focused on maximising ease of use within the Revo's constraints. These example interfaces for the PDA and the way they were developed illustrates the overall approach of possible design space construction and selection of appropriate values and points in this design space. Without any such initial ground in which to start designing, such work inevitably becomes arbitrary, with the tendency of simply squeezing in data and widgets from the original desktop version design. Having no concrete alternative design options to consider, the resultant design would become more limited, its success largely dependent on the designer's ingenuity and craftsmanship.

6.3 User Concerns as Design Constraints

The previous section demonstrated the application of our keyframe browser design space to a particular environment - a small, touch sensitive screen PDA to be used in a mobile environment. The constraints for the environment (allowing less attention, less fine interaction, less spatial presentation, and so on) worked in the sense of limiting the possible design options from the large number of possible designs. The statements of constraints applied to the design space was rather rudimentary, coming from general ideas and guidelines on PDA interface design (for example, "auditory feedback is desirable"), but seemed to result in something useful enough to produce outcome designs.

In this section, a different set of constraints will be applied in a much more elaborate and systematic way, namely *the users*: different users have different preferences and background knowledge thus a suitable browsing interface for one user would be different from another user, even when the same specific target usage and physical environment is specified. In this section the users' background knowledge (mainly domain knowledge) and preferences work as limits on the design space.

This section will follow the following steps:

(i) *User attributes* pertinent to video browsing are identified and as a result a simple user classification is done, resulting in 8 different user types;

(ii) The identified user attributes are individually matched to desirable design options in the design space and interaction alternatives, then

(iii) Each of the user types is matched to the most suitable of the Físchlár browsers.

By following these steps, each user type carries a set of constraints and from this a "good" browsing interface for that user type can be determined from the whole of the design space. This matching result is used in the Físchlár video browser evaluation in Chapter 7, by identifying a test user's user type and assigning the most appropriate of the browsers to him/her for use. This is a way of relating the rather artificial browser/user/matching process in this chapter to the real world.

6.3.1 User Attributes

When using an interface, different users have different levels of domain knowledge, general computer skills, aptitude, motor skills, and various interaction style preferences. Probably one of the most difficult aspects of user interface design is this diversity of user characteristics. Different users will surely have different opinions about the same interface and can achieve their best usage with different interfaces, depending on their individual differences. So if a system is to be used by the general public, considering individual users' diverse characteristics becomes fundamental to developing an interface (or interfaces) that are acceptable to users.

Whether trying to identify the commonest ground for all possible future users and designing a single interface that could roughly cater for all (as can be seen in museum kiosk systems), or trying to categorise several user groups and provide alternative interfaces for each of these, prime importance is to consider the diversity of user characteristics and to try to reflect this in the final interface. It should be noted, however, that an attempt to classify or categorise such diverse and non-discrete aspects as human preferences will surely distort the real state of affairs [Potosnak *et al.* 86]. However, attempting to do this still does account for users and does help us to view the design process from a more user-oriented point of view.

This section identifies three important user characteristics or "user attributes" that affect usage, particularly in relation to keyframe video browsing.

USER'S KNOWLEDGE OF THE VIDEO CONTENT

Domain knowledge is one of the obvious factors that influence a user's keyframe browsing, and constitutes as one of the three dimensions in the "user cube" [Nielsen 93, p43] [Smith 97, p38] which categorises user characteristics along with system knowledge and general computer knowledge. This factor is in effect whether the user has seen the video (thus already has knowledge of the content before starting browsing) or not. The actual parameter values will be either of the following, though finer distinctions may also be possible (having a rough knowledge of the video without having seen it, for example, knowing that it is a typical action film with important action scenes towards the end of the video would make a difference in browsing behaviour).

- have seen it before
- haven't seen it before

A simple example is a user who knows roughly how much into the video a certain incident happens - already knowing that an explosion scene s/he is looking for is near the end of the video, s/he will want to quickly go towards the end of the time-ordered keyframe set and then more slowly look for a keyframe that contains an explosion image. Knowing the approximate time offset in relation to the whole video is often a very important clue for the user to browse, and this somehow naturally determines the shape of a browsing strategy and calls for a particular interface style that will efficiently support this strategy. A user who does not know where an explosion happens (and most likely will not know what the explosion will look like) will prefer random browsing without much concern of the time, and will be better off with an interface which allows quick but controllable keyframe browsing.

PREFERENCE ON BROWSING STRATEGY: LINEAR VS. STRUCTURED BROWSING

Video data has sequential order which characterises it from other types of data. A strong advantage of keyframe browsing in an interactive user interface is that it can provide different ways of accessing keyframes, rather than imposing a singular linear browsing (this can be compared to the hypertext version of a book, which allows different ways of reading, rather than imposing conventional reading from page 1 to the end).

A more conventional way of presenting a keyframe set would be linearly in time order: showing earlier frames first and later ones later. When more meaningful sub-story segmentation is possible, we can think of presenting keyframes in a more structured, almost hypertextual way by their identified sub-story units. One important advantage of hypermedia is that it can break the conventional linear way of user viewing and allows easier random access that is sensitive to the user's current state.

Whether linear presentation or structured presentation is preferred will again depend on different users and their tasks at a given time: a user wishing to know what the video is like will probably like to use an interface where keyframes are linearly presented from beginning to the end, whereas a user looking for a particular point in the video, and who knows the context of that point in the video, will most likely to want to use an interface where a concise overview of the whole set of keyframes is presented and can go into further detail for a particular part of it. However, it is important to realise that this is not to say that for a certain task, a certain user is always better off using a particular way of viewing. Depending upon the user's current state (e.g. tired or energetic, feeling lazy, stressed, hurried, calm, and all sorts of other factors at a particular moment) s/he might prefer a different viewing for the same task - it very much depends upon that particular point and thousands of other subtle factors and will be unable to identify them fully. Probably the most important thing is to realise the fact that an enormous amount of factors will affect a user's video browsing behaviour and we can identify only some of them.

PREFERENCE ON INTERACTION STYLE: STATIC VS. DYNAMIC INTERACTION STYLE

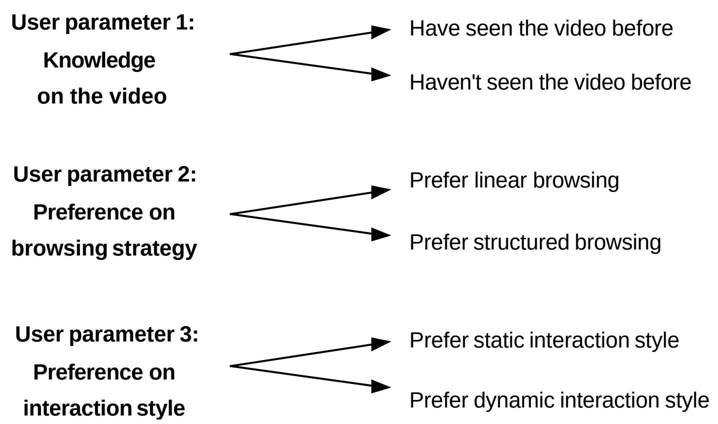
The third attribute contributing to characterising a user is his/her preference to a system-user interaction style, namely static interaction style or a dynamic interaction style. We consider one type of person who prefers an interaction style in which the screen is relatively stable and static, wherein s/he faces a static screenful of information and administrative elements and needs to purposefully act on something to get a reaction from the system. Probably more often found among users with only basic computer experience, this type of person is more satisfied when a minor user action does not cause a significant system reaction and instead a definite and determined user action (such as pointing to an object then clicking) results in definite reaction from the system.

On the other hand another type of person might prefer an interaction style in which the screen is more dynamic and sensitive to user's action, with direct manipulation (characterised by its immediate, incremental and continuous feedback) and a usually high rate of screen changes to user action. In general but not always, people with higher proficiency in the use of computers will be likely to prefer this type of interaction. In the meantime, a highly dynamic and changeable screen does not always mean that it provides a dynamic interaction style - a TV screen is highly changeable with many pictures showing in a short time, but it is one-way only and completely non-interactive. In this sense, for the Static/Dynamic interaction style we define here, it roughly corresponds to "low/high level of interactivity", where the dynamism of a screen is caused by the user's action, rather than by the system itself.

This preference affects a user's use of an interface and is closely related to how an interface is programmed to react to user action. Thus it is more linked to interaction alternatives, for example, whether a ToolTip box pops up on the screen, whether one has to click a timeline bar or just bring the mouse-over it, etc.

CONSTRUCTING A USER CLASSIFICATION

To summarise the three user attributes (or parameters) that we wish to cater for:



There will be many more parameters than these 3 which affect the usage of video browsing, but these are the three identified attributes we consider the most important in the context of video browsing with currently available graphical user interfaces.

To consider the three user parameters in terms of an integrated "user," now we set 8 *User Types* by all possible parameter combinations. A user will fall into one of these 8 types. This is shown as Table 6.1 below:

Table 6.1 - Eight User types by combining user parameters

	Video knowledge (Have seen / Haven't seen)	Preferred browsing strategy (Linear/ Structured)	Preferred interaction style (Static / Dynamic)
User Type 1 (U1)	Have seen	Linear	Static
User Type 2 (U2)	Have seen	Linear	Dynamic
User Type 3 (U3)	Have seen	Structured	Static
User Type 4 (U4)	Have seen	Structured	Dynamic
User Type 5 (U5)	Haven't seen	Linear	Static
User Type 6 (U6)	Haven't seen	Linear	Dynamic
User Type 7 (U7)	Haven't seen	Structured	Static
User Type 8 (U8)	Haven't seen	Structured	Dynamic

It should be acknowledged that this user classification in the table above is rigid and very simplified. There would be many cases this classification of users cannot address. For example, the user might have not seen a particular episode of a TV programme, but still know the overall structure of content (popular quiz programmes often starts with introduction to participants, follows same sequence of different types of quiz; news starts with today's summary, followed by a top story, ending with weather forecast, and so on). In this case, what User Type does the person fall into? What is attempted here is to demonstrate the main point of how analytic design method can address different users and can be usefully exploited, thus this simple user categorasation was decided acceptable here. If we are to further develop this idea in the future, surely more elaboration and more complex classification that can cover finer level situations will be required.

After using a particular interface, a user, having one of the combinations in the above table, should feel that the system has a "good interface" - or indeed is a "good system" - when the interface s/he used at that moment matches his/her attributes. These 8 User Types will be used for further development of our framework.

6.3.2 Matching between Users and Físchlár Browsers

This section is the last step in our process - attempting to link between the identified design alternatives and the identified user parameters. The process will then be applied to 8 Físchlár interfaces, to determine the best Físchlár interface for each of the User Types. This whole process eventually turns into the form of suggestions on which browser to use for a user when s/he comes to the Físchlár system, and further suggestions on alternative browsers. This is the stage where the user side and the interface side are related and are linked to each other. This process is not in any sense precise and accurate, as [Stary 00, p84] says:

> *The process of mapping complex, contingent human behaviours of information processing to rule-bound events and properties of accurate interfaces is an extremely challenging task.*

USER PARAMETERS AS SPECIFIC CONSTRAINTS ON DESIGN ALTERNATIVES

Using the full list in the previous two chapters, i.e. keyframe browsing interface dimensions and their values, possible interaction alternatives, and user parameters, we now try to match them together. What must be provided (Y) and what must not be provided (N) in an interface for each user parameter are marked in Table 6.2 below. Features that do not matter to provide or not are left blank. A short explanation for the reasons for marking is also provided.

Table 6.2 - Design alternatives and user parameters

	Layeredness					**Temporal orientation**						**Spatial vs. Temporal**			**Presentation style**	
			Multiple layer with link				Absolute time			Relative time			Temporal			
	Single layer	Multiple layer without link	Layers overlap	Layers don't overlap	Mixed spatial and temporal	No time	Timestamp in keyframe	Dedicated location	ToolTip box	Timeline bar	Timeline bar with selection indication	Spatial	Constant flipping speeds	Flipping speed controllable	Static	Dynamic
Have seen Video	N					N	Y									
Haven't Seen video	Y		N	N	N											
Linear Browsing	Y		N	N	N					Y						
Structured Browsing	N		Y									Y				
Static Interaction					N			N	N				N	N	Y	N
Dynamic Interaction						N									N	Y

Y (Yes) : Important to provide this feature
N (No) : Must not be used
blank : Does not matter whether provided or not

It was determined that a user who has seen the video should not be provided with a single layer interface - s/he will more likely prefer multiple layer browsers, knowing roughly or clearly what the video content is like and where scenes occur. For a user who has not seen the video, however, content-based navigation between different layers is not desirable, as it makes less sense for them to know where some story occurs in relation to other parts. Such users will benefit more from either a simple single layer or multiple layers without a navigational link. S/he who has seen the video will also prefer browsing with some form of time information provided, because s/he will likely know some scenes in the video in terms of their relative time progression (for example, "this scene happens half way through the video", "I know the scene I want to see is near the end of the video"), whereas for s/he who has not seen the video this is less likely to be an advantage in browsing.

A user who prefers a linear browsing strategy will like to use the single layer or the multiple layer without a navigational link, but the browsers that use jumping between layers with an explicit navigation mechanism will not be liked. The timeline bar is one of the browsing aids that s/he will like because it represents a linear progression of video content. On the other hand, s/he who prefers a structured browsing strategy should not be provided with a single layer browser but any type of multiple layer browsers that provide a navigational link, as this linked browsing is the strategy such a user is likely to want to use. S/he will also like to use a browser where keyframes are spatially presented because structured browsing can benefit from spatial arrangements of keyframes, although temporally presented browsers should not necessarily be perceived negatively either.

For the user who prefers a static interaction style, any form of temporal presentation should be avoided - whether keyframes are automatically flipped through or manually by the user, this makes the browser too dynamic for him/her. Any fast moving or floating objects on the screen also should be avoided, such as time information constantly changing on a screen or a ToolTip box popping up as s/he moves the mouse cursor on the screen. This user will prefer the browser which provides a static interaction style, rather than a dynamic (static and dynamic interaction styles as a browser's attribute were explained in Chapter 5). S/he who prefers a dynamic interaction style should be provided with some form of time information, because the dynamic interaction could easily lead to time disorientation. S/he also should be provided with the browser with a dynamic interaction style rather than static.

USER TYPES AS CONSTRAINTS ON DESIGN ALTERNATIVES

We now construct a table that matches each of the design alternatives to the 8 User Types in Table 6.3 below. This table will be used to measure how well each Físchlár browser supports each User type later on.

Table 6.3 - User Types to design alternatives

<table>
<tr><th rowspan="3"></th><th colspan="5">Layeredness</th><th colspan="6">Temporal orientation</th><th colspan="3">Spatial vs. Temporal</th><th colspan="2">Presentation style</th></tr>
<tr><th rowspan="2">Single layer</th><th rowspan="2">Multiple layer without link</th><th colspan="3">Multiple layer with link</th><th rowspan="2">No time</th><th colspan="3">Absolute time</th><th colspan="2">Relative time</th><th rowspan="2">Spatial</th><th colspan="2">Temporal</th><th rowspan="2">Static</th><th rowspan="2">Dynamic</th></tr>
<tr><th>Layers overlap</th><th>Layers don't overlap</th><th>Mixed spatial and temporal</th><th>Timestamp in keyframe</th><th>Dedicated location</th><th>ToolTip box</th><th>Timeline bar</th><th>Timeline bar with selection indication</th><th>Constant flipping speeds</th><th>Flipping speed controllable</th></tr>
<tr><td>User Type 1</td><td>N</td><td>Y</td><td>N</td><td>N</td><td>N</td><td>N</td><td></td><td>N</td><td>N</td><td>Y</td><td></td><td></td><td>N</td><td>N</td><td>Y</td><td>N</td></tr>
<tr><td>User Type 2</td><td>N</td><td>Y</td><td>N</td><td>N</td><td>N</td><td>N</td><td></td><td></td><td></td><td>Y</td><td></td><td></td><td></td><td></td><td>N</td><td>Y</td></tr>
<tr><td>User Type 3</td><td>N</td><td></td><td colspan="2">Y</td><td>N</td><td>N</td><td>Y</td><td>N</td><td>N</td><td colspan="2">Y</td><td>Y</td><td>N</td><td>N</td><td>Y</td><td>N</td></tr>
<tr><td>User Type 4</td><td>N</td><td></td><td colspan="3">Y</td><td>N</td><td colspan="5">Y</td><td>Y</td><td></td><td></td><td>N</td><td>Y</td></tr>
<tr><td>User Type 5</td><td colspan="2">Y</td><td>N</td><td>N</td><td>N</td><td></td><td></td><td>N</td><td>N</td><td>Y</td><td></td><td></td><td>N</td><td>N</td><td>Y</td><td>N</td></tr>
<tr><td>User Type 6</td><td colspan="2">Y</td><td>N</td><td>N</td><td>N</td><td>N</td><td></td><td></td><td></td><td>Y</td><td></td><td></td><td></td><td></td><td>N</td><td>Y</td></tr>
<tr><td>User Type 7</td><td>N</td><td>Y</td><td></td><td></td><td>N</td><td></td><td></td><td>N</td><td>N</td><td></td><td></td><td>Y</td><td>N</td><td>N</td><td>Y</td><td>N</td></tr>
<tr><td>User Type 8</td><td>N</td><td>Y</td><td></td><td></td><td></td><td>N</td><td></td><td></td><td></td><td></td><td></td><td>Y</td><td></td><td></td><td>N</td><td>Y</td></tr>
</table>

Y (Yes) : Important to provide this feature
N (No) : Must not be used
blank : Does not matter whether provided or not

This table is the specification of browsing interface requirements for each of the 8 User Types that we have identified. For example, if we are to design a browsing interface for User Type 2 (who has seen the video, has preferences for Linear browsing and Dynamic interaction), such a browser should provide multiple layers without a navigational link, should provide a timeline bar and a dynamic interaction style, as specified in the User Type 2 row in the table. This table makes a good example of user constraints to be applied to the design space, illustrating the fact that different types of users have different interface requirements, and how they can be concretely specified.

MATCHING BETWEEN 8 USER TYPES AND 8 FÍSCHLÁR BROWSERS

The Físchlár system's 8 browsing interfaces currently have different design alternatives and support different user attributes. Using Table 6.3 above, we can now relate users to the 8 Físchlár's browsers. In Table 5.1 in Chapter 5, we characterised the 8 Físchlár browsers in terms of the detailed design alternatives. Each browser's characteristics in this table can be compared to each User Type requirements list in Table 6.1. For example, to see how well the **Timeline Bar** browser supports User Type 2, we compare the browser specification and the User Type 2 specification. We compare User Type 2 in Table 6.1 and the **Timeline Bar** in Table 5.1 (see Figure 6.6 below).

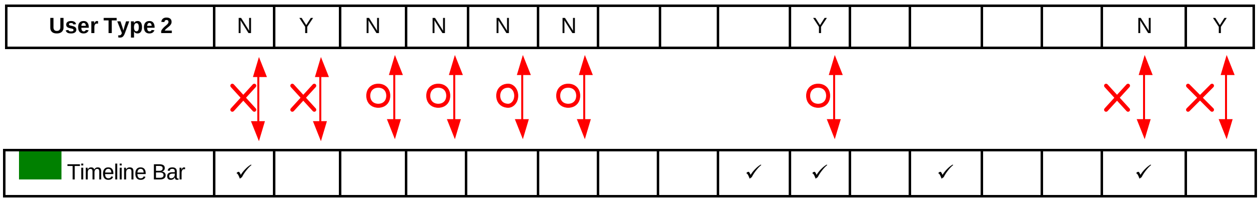


Figure 6.6 - Comparing User Type specification and browser specification

Of the 9 requirements to satisfy User Type 2 (the number of cells with either Y or N), the **Timeline Bar** browser satisfies 5, thus the **Timeline Bar** browser is 55.6% (= 5/9 x 100) suitable for a user in User Type 2. In this way, we can measure how suitable each of the 8 Físchlár browsers are for each User Type. Table 6.4 below shows the full calculation of this. Highly suitable cases are bolded.

Table 6.4 - How well the current 8 Físchlár browsers support the 8 User Types

	Scroll Bar	Slide Show	Timeline Bar	Quick Bar	Overview/ Detail	O/D + Quick	Dynamic Overview	Hierarchical
User Type 1	9/13 (69%)	10/13 (77%)	10/13 (77%)	8/13 (62%)	**12/13 (92%)**	10/13 (77%)	8/13 (62%)	7/13 (54%)
User Type 2	3/9 (33%)	5/9 (56%)	5/9 (56%)	7/9 (78%)	7/9 (78%)	**9/9 (100%)**	7/9 (78%)	5/9 (56%)
User Type 3	8/12 (67%)	8/12 (67%)	9/12 (75%)	7/12 (58%)	**10/12 (83%)**	8/12 (67%)	7/12 (58%)	8/12 (67%)
User Type 4	1/7 (14%)	2/7 (29%)	3/7 (43%)	5/7 (71%)	4/7 (57%)	6/7 (86%)	**7/7 (100%)**	5/7 (71%)
User Type 5	**10/11 (91%)**	**10/11 (91%)**	**10/11 (91%)**	8/11 (73%)	**10/11 (91%)**	8/11 (73%)	6/11 (55%)	6/11 (55%)
User Type 6	4/8 (50%)	6/8 (75%)	6/8 (75%)	**8/8 (100%)**	6/8 (75%)	**8/8 (100%)**	6/8 (75%)	4/8 (50%)
User Type 7	8/10 (80%)	6/10 (60%)	7/10 (70%)	5/10 (50%)	**9/10 (90%)**	7/10 (70%)	5/10 (50%)	7/10 (70%)
User Type 8	1/6 (17%)	1/6 (17%)	2/6 (33%)	4/6 (67%)	4/6 (67%)	**6/6 (100%)**	5/6 (83%)	4/6 (67%)

As can be seen in Table 6.4 above, some browsers support certain User Types better than others, and some User Types are more readily satisfied by several browsers than others. For example, the **Dynamic Overview** browser is suitable for User Type 4 by 100%, but is not suitable for other User Types. User Type 5 can be satisfied with **Scroll Bar**, **Slide Show**, **Timeline Bar** and **Overview/Detail** browsers all by 91%, whereas User Type 8 can be satisfied 100% by **Overview/Detail+Quick** browser and all other browsers are quite low in suitability.

BEST AND WORST BROWSERS FOR THE 8 USER TYPES

Based on the previous table, presented in Table 6.5 below is the ranking of best browsers and worst browsers for the 8 User Types and some short remarks on each.

Table 6.5 - Best and worst browsers for the 8 User Types

User Type	Best browsers	Worst browsers	Remark
U1 Seen video Linear Static	Overview/Detail (92%)	Hierarchical (54%)	[Seen video] + [Linear preference] combination makes multiple layer without link about the right interface to use. Linear and static preference resulted in Hierarchical browser being the worst one.
U2 Seen video Linear Dynamic	O/D+Quick (100%)	Scroll Bar (33%)	As can be noticed in this table, interfaces with multiple layers without link (O/D or O/D+Quick) are generally good browsers for many user types. Most especially the Quick element in O/D+Quick browser made it better for the Dynamic user as in this case.
U3 Seen video Structured Static	Overview/Detail (83%)	Quick Bar (58%) Dynamic Overview (58%)	For this User Type, no single browser is more than 90% suitable. The best one is only 83% suitable (Overview/Detail). A static interaction preference resulted in both Quick Bar and Dynamic Overview being the worst browsers for this User Type.
U4 Seen video Structured Dynamic	Dynamic Overview (100%) O/D+Quick (86%)	Scroll Bar (14%) Slide Show (29%)	Being the most exploratory and active, this User Type suits most dynamic, highly interactive interfaces, thus Dynamic Overview, and the 2nd best one is O/D+Quick. This means simplistic, less interactive interfaces are not preferred - very low suitability for Scroll Bar and Timeline Bar.
U5 No knowledge Linear Static	Scroll Bar (91%) Slide Show (91%) Timeline Bar (91%) Overview/Detail (91%)	Hierarchical (55%) Dynamic Overview (55%)	Being the least exploratory and most conservative, this User Type suits static, less interactive interfaces but providing time information properly, such as Timeline Bar and Overview/Detail. Hierarchical browser is worst as it requires a high level of interaction and no time information is provided. Dynamic browser also unsuitable for its high level of interaction.
U6 No knowledge Linear Dynamic	Quick Bar (100%) O/D+Q (100%)	Scroll Bar (50%) Hierarchical (50%)	The Linear but Dynamic preference of this User Type made Quick Bar and O/D+Quick browsers' linear style but dynamic interfaces the 100% best interfaces.
U7 No knowledge Structured Static	Overview/Detail (90%)	Quick Bar (50%) Dynamic Overview (50%)	[Haven't seen video] + [Structured] or [Seen video] + [Linear] combinations usually make interfaces with multiple layers without link good ones to use. Along with a Static preference made Overview/Detail the best interface. The worst ones are dynamic ones.
U8 No knowledge Structured Dynamic	O/D+Q (100%)	Scroll Bar (17%) Slide Show (17%)	O/D+Quick browser is also frequently the top-ranked browser along with Overview/Detail browser. The Structured and Dynamic preference of this User Type makes Scroll Bar and Slide Show the worst browser to use.

Above table tells us what interface or interfaces to present to which User Type, and what interfaces to avoid. More importantly, it gives us an indication of whether an interface can be further improved to suite a User Type. For User Type 1 the best interface is **Overview/Detail** browser by 92% - the reason this does not reach 100% can be found by comparing User Type 1's specification and **Overview/Detail** browser specification **- Overview/Detail** browser's popping-up ToolTip box (note this is a dynamic element on the screen) for time indication makes the static preference of User Type 1 less suitable. By removing the ToolTip box feature from the browser or modifying it to static timestamping in each keyframe, we achieve a 100% suitable browser match (according to our own matching process) for User Type 1. In fact, this modification will make the browser for many other User types (U3, U5, U7) also perfectly or moderately suitable - the **Overview/Detail** browser is the most versatile browser. It is the most suitable for as many as 4 different User Types (1, 3, 5 and 7). On the other hand the **Overview/Detail + Quick** browser is 100% suitable for as many as three User Types (2, 6 and 8) corresponding to these User Types' dynamic preference style.

The **Hierarchical** browser's suitability to different User Types is seen to be generally low. The highest suitability is understandably for User Type 4 (having seen the video, having structured and dynamic preferences), although this is only at 71%. This is due to the fact that the **Hierarchical** browser does not contain any form of time information at all, whereas User Type 4 should benefit by having time information while also browsing in structured and dynamic way.

From the above ranking table, now we can suggest a particular Físchlár browser when a user comes to our system, by identifying his/her User Type as illustrated in Figure 6.7 below:

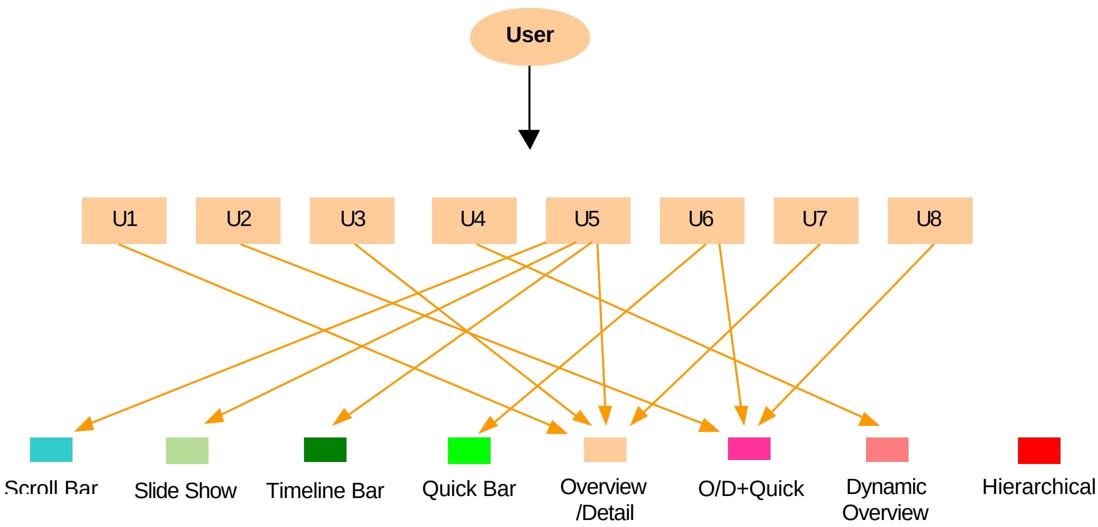


Figure 6.7 - Suggestions for a browser to start with

The context of this Figure is that when a person comes to the Físchlár system to browse video content, we can identify his/her User Type probably by asking three user parameters ("have you seen that video?", "what is your preference on browsing strategy?" and "what is your preferred interaction style?"), then according to the answer, we can suggest which browser to use. For some users we can suggest more than one browser (see User Type 5 and 6), the reason being that some browsers resulted in the same highest percentage of suitability for a User Type, though for different reasons according to their attributes.

It should be noted that the initial matching table of user attributes and browser attributes (Table 6.2), which resulted in the User Type-browser attribute table (Table 6.3) and the above browser ranking tables (Table 6.4 and 6.5), largely came from the theoretical and analytic work done in previous chapters, and needs to be verified in some way. The notion of "making a browser 100% perfect for certain User Types" is only in terms of the framework we have developed in this chapter and the parameters we have chosen to identify User Types. The initial way in which user attributes are matched to browser attributes could be argued without any evident conclusions (for example, is it always true to say that “s/he who has seen the video should not be provided with a single layer browser?”). This has merely been a starting point for matching between users and browsers, and once suitable browsers for each User Type were identified, more concentration will be given to refining these browsers for those users by conducting user testing rather than attempting to reverse-engineer the initial matching table.

BROWSER SIMILARITY IN TERMS OF USERS

Having viewed and analysed how close we believe our browsers are to our User Types, we can now view how the browsers are related to each other based on similarities or how close the 8 Físchlár browsers are, in terms of user preferences. For each User Type the best browsers were taken and linked to each other, thus linked ones mean there is a similarity between them. The differences in terms of user preferences are labelled on the arrows (see Figure 6.8 below):

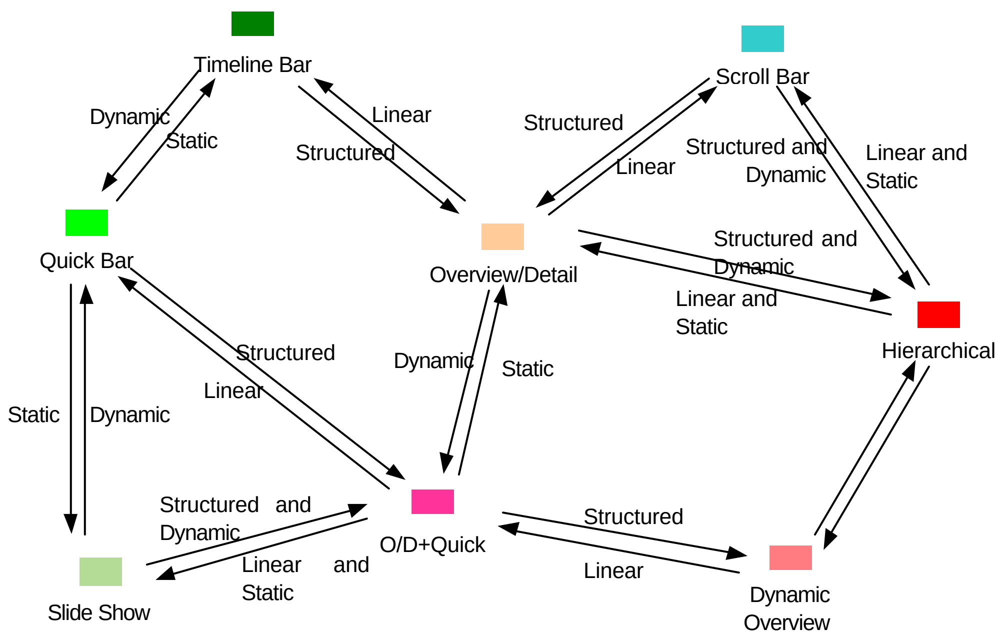


Figure 6.8 - Físchlár browser similarities in terms of user preferences

Note that the similarities and differences among the browsers were taken from the Table 6.4 - from the User Types' calculation rather than from browser characterisation of Table 5.1 in Chapter 5. In fact, several different views could be drawn using browser characterisation alone, but Figure 6.7

above was drawn from users' two preferences and thus is probably a better representation of browser similarity. The arrows indicating no labels (i.e., between **Hierarchical** and **Dynamic Overview**) means that there is no user preference-based differences - this point arises because this model is not expressive enough to distinguish this fact. What other unknown user attributes could there be to clearly distinguish between **Hierarchical** and **Dynamic Overview**? In terms of users, the two browsers are both suitable for structured browsing, and dynamic interaction, although User Type 4 is likely to favour **Dynamic Overview** more due to the time information it provides. For one thing, the **Hierarchical** browser provides 4 layers of linked browsing, whereas the **Dynamic Overview** provides 2 linked layers, yet this fact is not expressed because the design space only specifies multiple layers or not. Another point is that the **Hierarchical** browser is a completely spatially presented browser, with every keyframe's appearance and disappearance depending on the user's mouse cursor, whereas the **Dynamic Overview** combines temporal presentation with spatial presentation, with its dynamic aspect expressed by the fact that the user's mouse movement starts flipping through keyframes in detail view. One might consider **Dynamic Overview** "less" dynamic than the **Hierarchical** browser because the former interacts with the user less when automatic flipping through starts when in detail view. These considerations are interesting and might provide further new elements or attributes that need to be integrated either into the design space or as part of the user parameters. Part of the goals of the user-oriented evaluation we will present in Chapter 7 is to explore these relationships between browsers hoping to get more information on possible attributes not identified in constructing framework so far. This will be addressed in Chapter 7.

6.4 Summary: Físchlár Browsers and Users

The analysis and design space construction, user classification, matching between 8 Físchlár browsers and 8 User Types, and similarity between browsers can all be summarised as Figure 6.9 below, completing the user-as-constraints task we set ourselves in this Chapter:

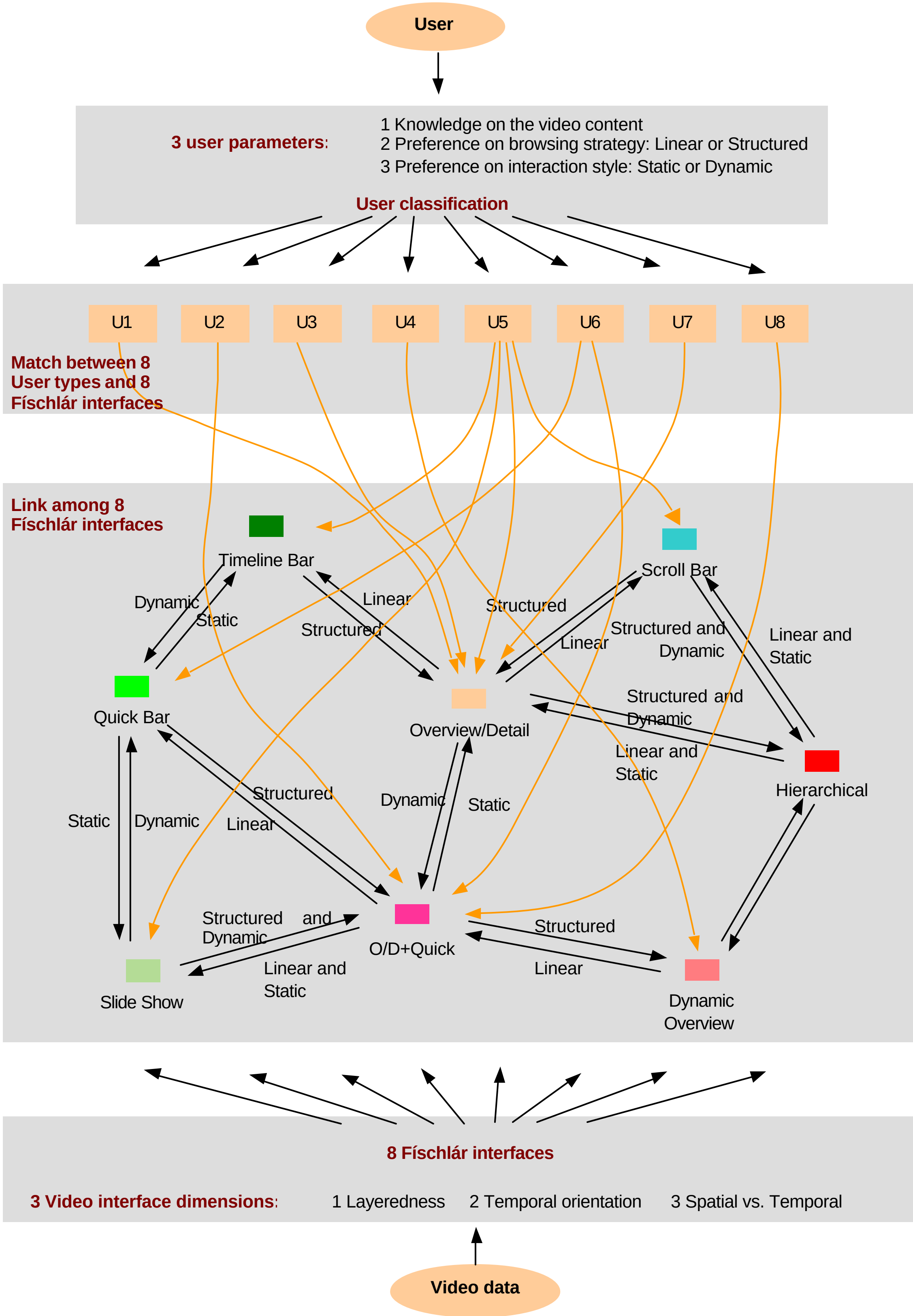


Figure 6.9 - Complete diagram of the user-browser framework

User concerns comes from the top of Figure 6.9, browser concerns comes from the bottom of the diagram, and they are matched in the middle part of the Figure. Completing this Figure for a given scenario involves a 4-level tuning processes:

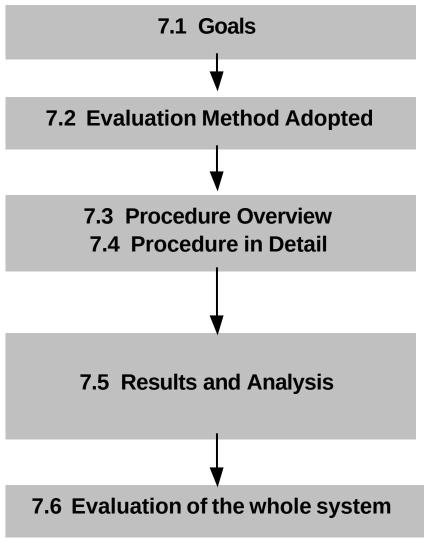
1. Identifying browser dimensions/values and interaction alternatives for each value, then design interfaces by deciding the point in the dimension space specifying a set of design alternatives;
2. Identifying user parameters then composing User Types;
3. Matching between designed interfaces to user types, then
4. Linking among designed interfaces by looking at similarities in terms of user parameters.

This way a user-interface design becomes a more clearly identifiable process in which the main actors (the user and the interface) are analysed, possibilities are considered, suitable constraints are specified and applied and as a result a specific interface is suggested to a specific user. However, this work is largely based on theoretical analysis and assumptions, lacking any kind of confirmation from the real world. In the next chapter, the above diagram will be used to plan and conduct an evaluation of the Físchlár browsers and the underlying framework so far constructed.

CHAPTER 7

The Evaluation of the Video Browsers

This chapter evaluates the eight Físchlár video browsers designed and analysed so far, by conducting a qualitative user testing. By identifying the usage problems, opinions and ideas from test users, the evaluation tries to improve the designed browsers individually and also importantly attempts to discover other elements or attributes of the design space and of the user model, which might have been overlooked or neglected while shaping the framework in previous chapters. The outcome of the evaluation provides further refinement of the browsers and insight into better design of browsing interfaces.



7.1 Evaluating the Browsing Interfaces

The design of the eight Físchlár keyframe browsing interfaces was based on the design space introduced in Chapter 4. However, it is true that common sense, consideration of other existing interfaces, possibilities and limitations at the programming level, past design experience and even some personal preferences have taken part in the actual implementation stage (the long listing of interaction styles and widget options in Chapter 5 has been done after most of the browsers were implemented) and rightly done so, because knowledge gained after working on the actual implementation enabled us to come up with the list of possible interaction widgets and styles - and the check list for each browser's features was also drawn up after the browsers were implemented.

The design space, interaction list and the user classification constructed in the previous chapters have been used as a framework for the browser evaluation in this chapter. Although much informal user testing has been conducted in the form of demonstrations, discussions, simple think-aloud and interaction observations whenever a browser was implemented and tested, there is a need to conduct a more regulated, recorded, objective test stage for individual browsers. Also, a more directed questioning on the comparison between different browsers would be very beneficial to the browser design study. This is especially true since there is so much variety of ideas in the video browsing interfaces as mentioned in Chapter 3, and it was considered particularly important to obtain people's fresh point of view and opinions on some of the browsers for brainstorming purposes.

Thus a plan was set for evaluation of the implemented browsing interfaces with test users. Two main goals of this evaluation were:

Goal 1: To find out usability problems with individual browsers

One of the greatest advantages of conducting a usability testing is to reveal small and large "glitches" in the design. By getting the test users to use the browsers and highlighting any problems they encountered, these design flaws and usability problems can be relatively easily found and from this, refinement carried out. As the Físchlár browsing interfaces were first implemented and tested, much informal user testing in the form of demonstrations and think-aloud has been tried and problems identified and removed. This showed us how many problems a newly-developed interface can have and how cheaply and easily many of these problems can be found out by doing a simple test run with users. This evaluation's first goal was to continue finding out further usability problems, if any, and to see if those problems would be related to larger problems relating to the original framework from which the browsers were developed.

Goal 2: To find out other important elements or attributes of browsing interfaces and/or user classification, which might have been overlooked or ignored while constructing them.

Test users' comments and opinions can provide a very rich qualitative ground to pull together ideas and clues to better design space analysis. The initial analysis of the design space - including the three dimensions and the values for each of the dimensions - was largely based on the premium idea of the characteristics of the keyframe set and other existing and thinkable browsing interfaces today. Surely there may well be some important items missed from the design space - for example another important feature of keyframe browsing interfaces could be incorporated into the design space as a 4th dimension with its possible values. What was hoped in this evaluation was that the comments from users after using some of the browsers will reveal important clues on what other important features have to be considered in designing a browser (i.e., new dimensions and new values), or modification of the currently analysed dimensions and values, and overall provide some further insight into the browsing interface design.

7.2 The Evaluation Method Adopted

To achieve the above goals, the method we decided on was to be mainly *user-oriented evaluation*: qualitative rather than quantitative, a micro- rather than macro-evaluation [Harper 96] where each case is carefully considered for extrapolation rather than averaging different cases for generalisation [Bawden 90, p92].

For the most part, it was considered that at this early stage of interface design exploration, any quantification of exact measure would not be of much use - if a specific end-user application with specific targets was developed using some of the Físchlár browsers, then a well-defined category of performance and interaction efficiency could be set and measuring them by exact measure would become more appropriate. Instead, the main purpose of the evaluation was, as mentioned earlier, to find out usability problems and hopefully come up with better ways of structuring the design space. There is a generally accepted "ready-made" categorisation of usability similar to [Nielsen 93, p26] and [Shneiderman 98, p15]:

- Learnability - how quickly the user can learn how to use the system
- Efficiency - how quickly the interface allows a task to be carried out
- Memorability (retention over time) - how easy it is to remember how to use different features
- Error rates - how easy it is to make errors and to recover from them
- Subjective satisfaction - how pleasant it is to use the system

If a specific context for a system's target exists, we can set priorities in the categorisations (for example, for video editing systems in a broadcast company, efficiency would be a more important category than subjective satisfaction) and specific quantitative measures could be determined for the categories (for example, time in seconds to complete a document reformatting task). When the interface under consideration is not for a specific application but for a general future use, instead what is pursued is to identify areas of focus and possible criteria. Well-defined categories are the object of research in qualitative studies, whereas they are the means of research in quantitative studies [McCracken 88, p17]. This kind of qualitative study to gain insight rather than measure exact performance is conducted more and more nowadays in the early stages of experimental multimedia system development projects. The Informedia Digital Video Library's initial deployment to the Open

University campus was evaluated with 29 participants, mainly through qualitative questionnaires and group discussion [van der Zwan *et al.* 99]. A small, handheld speech browsing system called SpeechSkimmer was developed and evaluation was conducted with 12 test users, mainly through interviews while asking them to use it [Arons 97]. The picture retrieval system EPIC's novel image querying interface was evaluated with only 8 users, with each user's case very much analysed in detail with questionnaires containing semantic scales [Jose *et al.* 98]. In the social science field where qualitative inquiry has been used for a long time, experimenters are not encouraged to conduct evaluation with more than 8 users, as this tends to lead the experimenters into thinking in a quantitative way and the amount of data gathered would be daunting, causing the characteristics of each user case to be overlooked. The rich user comments and discussions of individual user cases should help gain insight in shaping categories and frameworks to more clearly think about designing browsing interfaces. In the case of the Físchlár system, a quantitative evaluation using a much larger number of users would not have been a problem since at the time of writing, Físchlár has over 300 registered users and is in regular use by staff and students on campus and in the student residences at Dublin City University. In this case, Físchlár is used to access an archive of over 300 hours of digitised broadcast television which is updated continuously.

For the browser evaluation in this thesis, the main emphasis has been to obtain qualitative data from test users on the browsers and their comparison. While the design space analysis and user classification constructed in previous chapters served as an initial framework for the evaluation procedure as will be explained later, this framework was not meant to be of a definite nature, and indeed this evaluation's purpose was to modify this initial framework. The actual measures used are:

Satisfaction with each browser (i.e., how much the user liked the browser) in 7-point scale

This was used to see roughly whether the users like the browser suggested by the framework as the most suitable more than other alternative browsers.

Reason for satisfaction/dissatisfaction - open question

This was the main focus in the evaluation, to find out usability problems and obtain comparative remarks on different browsers. This question could provide rich qualitative data on ideas for browser improvement and other important elements to be integrated in a future design space.

7.3 Evaluation Procedure Overview

The evaluation procedure was planned based on the framework developed in the previous chapter. In presenting various browsers to test users, it was decided that only three browsers for each test user would be presented and their opinions asked: by showing only three, it was expected that a test user could more carefully assess each of them and compare among them better. What the evaluation is hoping to capture is the test users' detailed opinions and especially what makes a browser different from others and why, thus limiting the number of browsers for each person to 3 is likely to be more productive than showing all 8 browsers with small and large differences.

In selecting three browsers for each test user, the framework diagram of user model-browser matching drawn in previous chapter (see Figure 6.9) was used - the best browser for a person, and the neighbouring two browsers (i.e., the two browsers that are similar to the best browser, but different in terms of the 2 user preferences). This is illustrated in Figure 7.1 below. For example, if the test user is identified as User Type 3, then the three browsers to be presented will be:

- **Overview/Detail** browser (determined as the *"best" browser* for User Type 3)
- **Scroll Bar** browser (variation in terms of browsing strategy: *more Linear* than the best browser)
- **Overview/Detail + Quick** browser (variation in terms of interaction style: *more Dynamic* than the best browser)

This way, naturally getting some feedback in terms of comparative thinking would be likely to be achieved, and those user-perceived differences between similar browsing interfaces are exactly what this evaluation was hoping to reveal.

Surely there are other possible ways of designing the testing that would be useful for this evaluation. Exposing all 8 browsers to a user, though might not result in more focused comparative user comments, could result in other surprising insight. Showing simply one browser to different users and attempting to get as detailed comments as possible (between-subject comparison), or showing not the neighbouring similar browsers but very different 2 browsers would be another interesting way of getting comments. Although one particular strategy was taken in this evaluation, it will be benefitial to try a few different ways in the future.

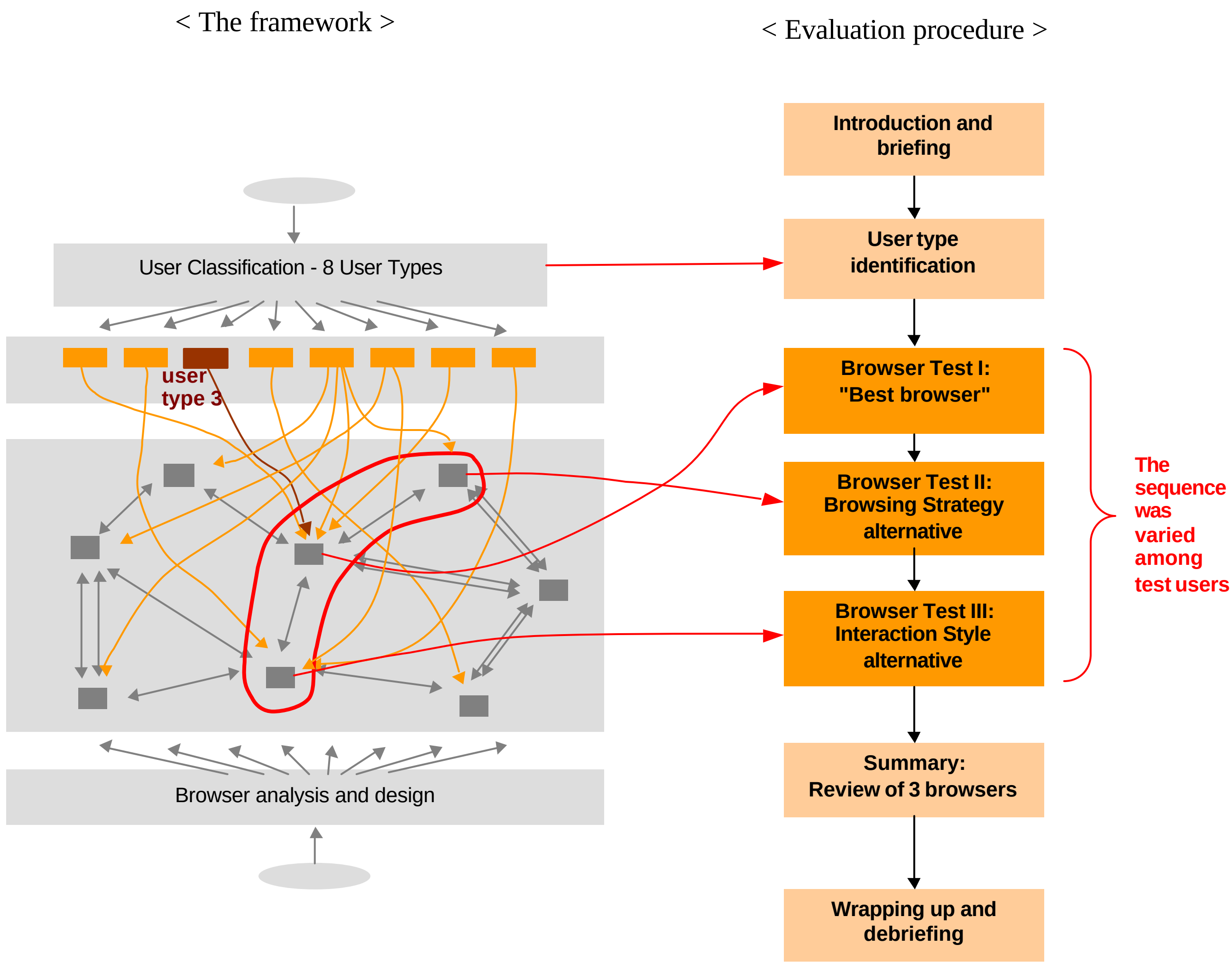


Figure 7.1 - Framework and evaluation procedure

A special web-based package was prepared so that the test user does the session online for about 30-40 minutes. This package is different from the Físchlár user interface described in Chapter 3, as shown in Figure 7.2 below.

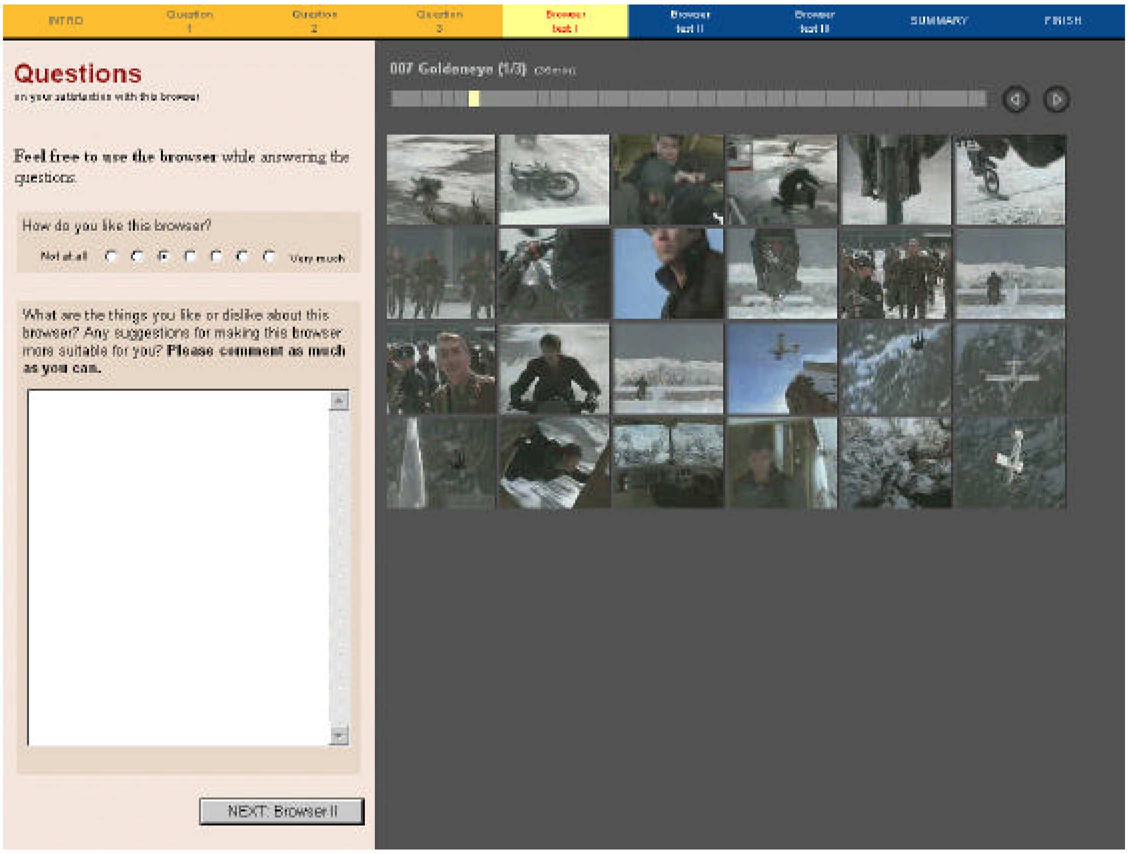


Figure 7.2 - A screen from the evaluation package

The detailed package interface design rationale and more of its screen looks can be found in Appendix A. The package starts by identifying the User Type of the current test user, and automatically determines which three browsers are to be presented for that user. Measures are automatically recorded as the user ranks and types in his/her opinions and comments, which were analysed after each user session was completed. The test video material was the film "007: GoldenEye" (MGM, 1995), divided into 3 equal-length parts for each browser testing. The film was chosen because much of its content had high brightness and action scenes that could be relatively easily recognised even when the size of each keyframe was small, and also because the video content contained various incidents and situations.

After brief introductory screens explaining the test procedure, the User Type identification stage presented several questions and example interfaces to determine the user's User Type, followed by 3 subsequent Físchlár video browsers each of which catered for that User type's preferences in different aspects (see Figure 7.1 above).

At each of the three browser presentations, a simple task was assigned to the user and questions asked. What the evaluation was expecting to capture from users' answers was their liking/disliking of each of the three browsers and their reasons. The higher the rating for the "best" browser over the other two, the more verified the framework would be, and also the more users' reasons for that rating are in terms of *"more Structured"* or *"less Dynamic"* - even if their actual phrasing might not be exactly that - the more verified the framework would be. Presenting all eight different browsers and asking for opinions was considered harmful as this could overload the test users and might turn out ending up getting less informative feedback overall, whereas with only three browsers to compare, it was expected we encourage more focused and comparative remarks. Finally, a summary of the three browsers was presented with a final set of questions and a debriefing screen thanking users for participating in the evaluation.

7.4 Procedure in Detail

Three browsers were presented to each test user, and which three of the 8 Físchlár browsers were to be presented was determined as follows. The order of the chosen 3 browsers decided below were evenly varied among the test users, thus minimising any effect caused by the sequence of the 3 browsers presented.

BROWSER I: BEST BROWSER FOR THE USER

In the previous chapter on User Type and browser matching, we have determined the best browsers for each User type, based on the constructed theoretical framework. This corresponds to the initial links established in the framework (arrows in orange colour). This works as an anchor browser with which the other 2 nearby browsers will be compared, and the reasons for the preference to be captured. Table 7.1 below shows the schedule for the first browser to present.

Table 7.1 - Schedule for first browser to present for each User Type

	Best browser
User Type 1	Overview/Detail
User Type 2	Overview/Detail+Q
User Type 3	Overview/Detail
User Type 4	Dynamic Overview
User Type 5	Scroll Bar
User Type 6	Quick Bar
User Type 7	Overview/Detail
User Type 8	Overview/Detail+Q

For example, a user identified as User Type 2 was presented with the **Overview/Detail+Quick** browser which our framework indicates as the best browser for him/her. With the browser determined, we asked the user to do a simple task of finding out a particular point in the video ("the scene where James Bond is falling off the cliff and catches the falling airplane and steers it back up the sky") and to playback from that point as a confirmation. So the data that was collected and used from this was:

1. Satisfaction (in a 7-point scale)
2. Reasons (any good/bad points) (as an open question)

The open question blank was made large, to encourage the user to type in as much comments as possible. At the end of these questions, the CONTINUE button led to the next browser to be tested.

BROWSER II: ALTERNATIVE BROWSER IN TERMS OF BROWSING STRATEGY

Our framework suggests alternative browsers once a user used the best browser. The alternative browsers are the browsers located nearby the best browser, with indications of different emphasis taken by that alternative one from the best one. Once the best browser testing was done with the first testing, we then presented one of the alternative browsers that changes the user's preference on browsing strategy either more supporting or lessening according to his/her preference. This can be identified from the link arrow's label among the browsers in the framework diagram. It should be noticed that there are quite a few alternatives to present for some User Types (for example, if a test user is determined as User Type 5, there are four different browsers that we can decide to present as the best browser). The two sets of choices taken in this evaluation are as Table 7.2 below. The second alternative schedule was made after the 16th test user, due to the fact that one of the browsers (**Scroll Bar**) had not been tested at all until the 16th testing, thus bringing the browser more likely to be used from then on. This was due to the fact that only User Type 5 was supposed to be presented with **Scroll Bar** browser, and no single user of the 20 users we used were identified as User Type 5.

Table 7.2 - Schedule for the second browser to present and alternative schedule

	Anchor Browser	**2nd browser to Present (test users 1-16)**	**2nd browser to Present (test users 17-20)**
User Type 1	Overview/Detail	Hierarchical	Hierarchical
User Type 2	Overview/Detail+Q	Dynamic Overview	Quick Bar
User Type 3	Overview/Detail	Timeline Bar	Timeline Bar
User Type 4	Dynamic Overview	Overview/Detail+Q	Overview/Detail+Q
User Type 5	Scroll Bar	Overview/Detail	Overview/Detail
User Type 6	Quick Bar	Overview/Detail+Q	Timeline Bar
User Type 7	Overview/Detail	Timeline Bar	Scroll Bar
User Type 8	Overview/Detail +Q	Slide Show	Overview/Detail

For example, the user identified as User Type 2 was thus initially presented with **Overview/Detail+Quick** browser, and was then presented with **Dynamic Overview** for more structured browsing (which will reduce the satisfaction). This way we expected it to be more likely that we would get the user's comments on their satisfaction / dissatisfaction on the second browser compared to the first, and their reasons why. A similar kind of task of finding out a particular point in the video was assigned ("the scene where a military tank runs over the exploding cars with fire"), and the same data was collected as in the previous test:

1. Satisfaction (in a 7-point scale)
2. Reasons (as an open question)

The reasons that the test users would give here were expected address in terms of Linear or Structured, as well as any other, yet-to-be-known ideas - although how they actually phrased this aspect could be different and should be carefully analysed. If the test users' reasons for their liking/disliking were about the linear/structure aspect, then it would confirm the soundness of one part of the framework. If the test users' reasons were not directly relevant to the linear/structure aspect, then these other reasons can be considered for new elements that might be included in an improved version of the framework. Because of this, it was important that the questions asked *did not lead* to certain answers (such as multiple choices or any other leading questions). It should not imply what aspect of the interface has changed from the previous interface seen by the user and what kind of answers are expected. Obtaining new insights into the framework was one part of this evaluation purpose. At the end of these questions, the CONTINUE button led to the next browser to be tested.

BROWSER III: ALTERNATIVE BROWSER IN TERMS OF INTERACTION STYLE

Like previous testing on preference on browsing strategy, we then presented the user with one alternative browser that is linked from the best (anchor) browser, in terms of interaction style. The alternative browser to be presented was one that allows either a more Static or a more Dynamic interaction style. This can be identified from arrow labels among the browsers in the framework diagram. There are a few alternative ways of this scheduling, and the decision was made as shown in Table 7.3 below. Again, the second schedule was made and used after the 16th user testing, because the **Scroll Bar** browser had not been used at all until the 16th user testing, thus to make it more likely to be used by users.

Table 7.3 - Schedule for the third browser to present and alternative schedule

	Best browser	**3rd browser to Present (test users 1-16)**	**3rd browser to Present (test users 17-20)**
User Type 1	Overview/Detail	Overview/Detail+Q	Scroll Bar
User Type 2	Overview/Detail+Q	Overview/Detail	Dynamic Overview
User Type 3	Overview/Detail	Overview/Detail+Q	Scroll Bar
User Type 4	Dynamic Overview	Hierarchical	Hierarchical
User Type 5	Scroll Bar	Hierarchical	Hierarchical
User Type 6	Quick Bar	Timeline Bar	Slide Show
User Type 7	Overview/Detail	Overview/Detail+Q	Timeline Bar
User Type 8	Overview/Detail+Q	Overview/Detail	Scroll Bar

For example, the user identified as User Type 2 who was initially presented with **Overview/Detail +Quick** browser, was presented at this stage with the **Overview/Detail** browser, reducing his/her satisfaction (as User Type 2 prefers Dynamic interaction style). Again a task to find out a particular point in the video was asked ("the scene where James Bond grabs the villain's foot at a high Radar, then lets go of him and he falls down"), with same data collection:

1. Satisfaction (in an 7-point scale)
2. Reasons (any good/bad points) (as an open question)

Again, the open question was important in that we expected the test users to give either Static/Dynamic aspects or any other reasons that might help us to gain more ideas on new attributes later to be integrated in the framework. At the end of these questions, the CONTINUE button led to the next stage of the evaluation process.

REVIEW OF THE 3 BROWSERS

After all three browser testings were done, a screen showing which browsers the user had used was summarised, for reminding purpose. With screen shots of the 3 browsers to help reminding, a final set of questions was asked:

1. Comparative satisfaction (by selecting the best browser of the three)
2. Reasons (as an open question)
3. Any other reasons for less liking the other two (as an open question)

DEBRIEFING

Finally, when the evaluation was complete, the screen showed short messages thanking the user again for participating in the evaluation and mentioning that we would let him/her know the outcome of this testing and any further browser development in the near future.

DEMOGRAPHY OF TEST USERS

Twenty test users participated in the evaluation, all of them already having good experience in using Windows-based desktop applications with a mouse and a keyboard, although there were some variations in the degree of proficiency. The basic demography is shown Table 7.4 below:

Table 7.4 - Demography of the test users

by department		by status		by sex	
Computing	16	Research student	9	Male	18
Electronics	3	Lecturer/researcher	7	Female	2
Biology	1	Undergraduate student	4		

7.5 Results and Analysis

This section analyses the evaluation results, by looking at the test users' feedback. The focus is to consider each test user's comments and come to conclusions from them, rather than trying to draw a statistical picture from the comments. Firstly the individual eight Físchlár browsers' concerns are discussed with test users' comments, by categorising the nature of the comments. Then a more general discussion and some conclusions are drawn in terms of browsing interface design. Finally, the ideas on strengthening the design framework and matching between browsers and users are discussed. All the user comments quoted in the following are exact words and phrases taken from the log data recorded from the evaluation package after the evaluation. As a result, spelling and grammatical mistakes from users may be found and sentences may not be complete. Square brackets with supplementary phrases are added where helpful for clarification purposes.

7.5.1 Regarding the Individual Browsers

Many of the test users' comments were the confusions and wishes based on the initial misunderstanding of the workings of the browsers. They are all valuable feedback for the designer (i.e., the author) to correct and devise an easier way to give the right impression on how to use them, but in long-term usage they might be of less significance in terms of usability. However, being able to extract general and specific opinions and comments on each of the 8 Físchlár browsers was informative and useful to gain insight into browser design and how people see and use interfaces overall.

Timeline Bar browser

TIMELINE IS EASY AND USEFUL

It was noted that test users commented on easy and straightforward use of the timeline.

> "Easy to navigate." (Test User 2)

> "likes: - easily browseable." (Test User 10)

> "I do like the time information in the link alt tag... it is useful to know how much of the video has been gone through..." (Test User 12)

IS FREQUENT CLICKING STYLE GOOD OR BAD?

The **Timeline Bar** browser uses the timeline where the user clicks a segment on the bar to see a set of keyframes from that time segment (see Figure 7.3). Thus to see all keyframes, the user has to keep on clicking until all the segments on the timeline are clicked. From the initial implementation of this browser, this point of requiring frequent clicking was often mentioned as tedious and frustrating. Even if the clicking action on a segment immediately displays the keyframes without any delay, the style of continuous clicking can get the user tired. Annoyance with the tedious clicking was found in many cases with browsers with this interaction style, especially in comparison to other browsers that allow a more dynamic response such as the **Quick Bar** browser. However, not everybody was reluctant to have a clicking timeline bar:

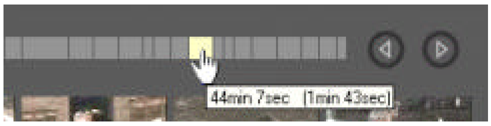


Figure 7.3 - **Timeline Bar** browser, triggered by clicking

> "I preferred the time bar where you click [rather than this one with **Overview/Detail+Quick** with mouse-over]" (Test User 11)

> "like:Does not swap between groups until I specify." (Test User 2)

More comparisons on clicking vs. mouse-over interaction styles were made when test users were presented with **Overview/Detail** and **Overview/Detail + Quick** browsers (see analysis of **Overview/Detail+Quick** browser below).

USEFULNESS OF THE ARROW BUTTONS

When the timeline is clickable as in this browser instead of by a mouse-over, having two arrow buttons beside the timeline seems to be useful. Test users who liked **Timeline Bar** browser liked to use the arrow buttons, and kept on clicking without moving the mouse while they looked at the keyframe area.

> "[Compared to **Overview/Detail + Quick** and **Dynamic Overview**] The search is easy with the arrows." (Test User 8)

Again, having a clickable timeline with arrow buttons is more frequently mentioned when **Overview/Detail** browser is presented (see analysis of **Overview/Detail** browser below).

Overview/Detail browser

USEFULNESS OF OVERVIEW

Overview in itself seems to be regarded very useful, providing a good idea what the video is about, as some of the comments say:

> "This is definitely the best of the three browsers and I like it a lot. I especially like the OVERVIEW and this was badly lacking in the second browser [**Timeline Bar**]" (Test User 19)

> "[After using **Overview/Detail+Quick** and **Overview/Detail** browsers] I missed the overview [in **Timeline Bar**]..." (Test User 12)

> "[**Overview/Detail** browser was better, because it shows overview.] (verbal discussion)"
> (Test User 17)

> "like:Overview provided a good view of the areas in the movie and very quickly." (Test User 2)

Having an overview feature is one particular case of a browser with multiple layers without a navigational link. The usefulness of the overview in the user interface has always been considered important and is mentioned often in interface design guidelines and heuristics. The test users' reaction and comments above seem to conform to this heuristic on providing an overview, and may signify the importance of providing the feature.

NO NAVIGATIONAL LINK BETWEEN THE LAYERS

Also mentioned and discussed quite intently in user comments and after-session discussion was the browser having *no navigational link between the two layers*. In the matching between the user parameters and browser alternative options in Chapter 6, it was initially determined that either a single layer or multiple layers without a navigational link would be preferred by people who have not seen the video and for people who prefer a linear browsing strategy. This means that, for example, a user who has not seen the video will be better served with a browser that has multiple layers but without direct linkage between point one in a layer to a point two in another layer. It is not clear whether a user who has not seen the video really prefers a browser without a navigational link, because user identification at the beginning of the evaluation might not be correct - indeed, the User Type identification stages (with three questions) were mainly meant to provide an initial anchor point in determining the 3 browsers to present, rather than to precisely capture the user's type (only 11 users out of 20 were properly predicted to their best browsers, thus their correct User Types). During informal testing and throughout the evaluation, it was noted that people want to be able to relate keyframes in the overview to the ones in the detail, checking if each keyframe in the overview corresponds to each segment in detail view. What was highlighted by the evaluation is that those users who have seen the video (whether they prefer structured or linear browsing) mentioned their wish to have some kind of link between the layers, though this was expressed in different phrases:

> "...but it might be better if clicking on a picture opened up the detailed view rather than [playing] a actual clip." (Test User 11)

> "It would be great to be able to select a number of adjacent segments... which can be expanded into the detail view." (Test User 12)

> "Clicking on an image when the browser is in the overview state should maybe bring up the section of the overview that that image is from, rather than the video player." (Test User 4)

> ..."the problem is that when clicking on image there was no indication of which group it belonged to, I had to search through the groups until the area came up... while the overview is a great idea, it is ruined by not expanding down in detail" (Test User 2)

> "Some way in the third interface [**Overview/Detail**] so that it isn't blind seek to the correct shot list. Might recognise the a few candidate locations in the overview but cannot browse around at the shot level." (Test User 1)

"[If clicking one picture in overview leads to corresponding detailed view, that would improve the browser] (verbal discussion)" (Test User 17)

The phrases used in the users' comments above - "open up detailed view", "expand", "bring up the section", "showing corresponding detail view" - all reflect the same expectation and wish to be able to navigate between the layers while maintaining the current point of browsing: navigational link between the layers.

GRAPHICAL REPRESENTATION OF THE DOUBLE-LAYERED TIMELINE

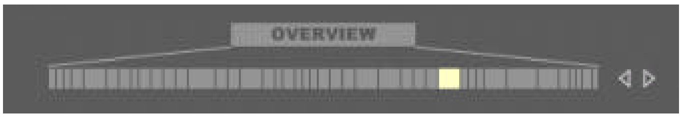


Figure 7.4 - **Overview/Detail** browser's layer visualisation

Having the visual representation of the overview and detail in the same area (see Figure 7.4) seems to help to better understand the double layered nature of the keyframe arrangement. Another double layered browser (**Overview/Detail+Quick**) does not have this graphical representation of the layers, and seems to be perceived as more ambiguous. Sometimes people expect the arrow buttons on the timeline bar to bring more timeline segments on left or right, instead of moving the current segment to the left or the right. Having the graphical overview and detail bound by the diagonal lines indicating the timeline bar is an expanded view of the overview, seems to reduce this misunderstanding, as some comments show:

"It is more obvious how to use Browser 1 [**Overview/Detail**] then the other browsers [**Overview/Detail+Quick** and **Hierarchical**]." (Test User 4)

"the option of navigating the overwiew and it being clear (graphically) where abouts in the program you are." (Test User 10)

"[In the **Timeline Bar** browser] the user doesn't [know] if the timeline bar is finished or not, so it can have the impression that is more than what is displayed. The second one [**Overview/Detail**] has an overview (even the word - suggests a complete thing)." (Test User 18)

But it is also true that at the initial implementation stage of the browser, this particular visualisation was often mentioned as not very appealing by several test users. The aesthetic appeal should be a part of a good user interface, and there should be a further prioritisation when a particular application is designed. For example, a video game for long-term teenage users will be better off emphasising a graphically appealing interface at the cost of initial slow learning, whereas a video browser for a walk-in kiosk will be better off having a graphical representation to help users understand how the interface works at the cost of a less appealing screen. However, these should not always be conflicting factors.

TEDIOUS CLICKING PROBLEM

Overview/Detail browser has the static interaction style where the user clicks on the timeline to see a set of keyframes, the same way as in **Timeline Bar**. As the comments on **Timeline Bar** above, there were comments on the tedious clicking, especially when compared to other browsers with the mouse-over interaction style, but also preference on this clicking when they used arrow buttons without having to move the mouse.

> "...first one [**Overview/Detail**] is too complex interface for everyday use (too many clicks)." (Test User 7)

> "[**Overview/Detail** is] less sensitive [than **Overview/Detail+Quick**] and the addition of the arrow buttons, allowed me to page down through the frames without having to spend time positioning the mouse each time..." (Test User 9)

NO TEMPORAL ORIENTATION IN THE OVERVIEW

Although the overview in the browser is mainly to provide simply a quick glance of the whole video content, having no time information for each overview keyframe was mentioned as a disadvantage:

> "When viewing an overview section there is no indication of where that section is in relation to time (ie. 5min - 10 min)." (Test User 4)

This would be a small matter and could easily be fixed by, for example, showing a ToolTip box when the mouse cursor is on a keyframe.

SMALL SIZE OF OVERVIEW KEYFRAMES

Some test users pointed out the size of each keyframe on the overview screen as too small, and it was noticed several times that users were moving nearer to the monitor when the overview keyframes were displayed.

> "Also, pictures are very small [in the overview], so make it larger when MouseOver." (Test User 6)

> "Screens are very small - needed to move up close to the monitor to see what the content of some of the keyframes [in the overview] was." (Test User 14).

The reason for making the keyframes in the overview smaller than the detail view was to display as many keyframes as possible on a single screen, but for some users the size was felt too small for inspection. The same problem was also pointed out in **Overview/Detail+Quick** browser below, and will be mentioned again.

Quick Bar browser

QUICKNESS A GREAT ADVANTAGE FOR SOME PEOPLE, BUT NOT ALWAYS

Throughout initial testing just after implementation and during this evaluation, this browser's immediate response in interaction was often pointed out as very positive, allowing a quick browsing without tedious clicking. Compared to **Timeline Bar**, using this browser the user can simply sweep the mouse over the timeline in a second to see the whole keyframe set. More comments in this respect were noted in **Overview/Detail+Quick**. It should be also noted that there *are* people who do not like this quick response. Some people said the movement was too sensitive and prone to inadvertent change (see analysis of **Overview/Detail+Quick** below), although the good points of such direct manipulation with immediate response on user action would be the fact that s/he can easily and quickly reverse the inadvertent change s/he made mistakenly. Suggested was some different widget behaviour that responds quickly but also allows stopping inadvertent mouse movement.

> "I prefer this quick mouse movement [to **Timeline Bar**], but I want to be able to fix the current position" (Test User 6)

For some people, **Timeline Bar** is too slow and tedious, but **Quick Bar** is too sensitive and easy to change. This seems to call for a new widget ideas for a better interaction mechanism. More concerns on this point were commented on in **Overview/Detail+Quick** browser below.

Overview/Detail + Quick browser

NO NAVIGATIONAL LINK BETWEEN THE LAYERS

As in **Overview/Detail** browser, having no navigational link between the layers has been pointed out:

> "Might be nice to expand from overview to detailed by clicking on a scene to expand" (Test User 13)

> "If the images in the overview mode were used to open the corresponding section in the the Detailed View, this would be very benefical for finding your chosen scene." (Test User 4)

USEFULNESS OF OVERVIEW

As in **Overview/Detail** browser above, the overview feature was pointed out as good:

> "The overview is a good idea and after using it, the other two browsers [**Overview/Detail** and **Timeline Bar**] seem so much slower in loading up the pictures." (Test User 11)

NEED FOR A BETTER QUICKNESS MECHANISM

As with **Quick Bar** above, the immediate response of the timeline in this browser often brought a great preference, especially after being exposed to timeline bar that requires frequent clicking.

> "The rollover feature [mouse-over interaction style] is attractive and it is easier to navigate and pinpoint a specific scene." (Test User 3)

> "I like the speed of access to the shot lists..." (Test User 1)

Again, its immediate response was also felt negatively by some:

> "...automatically changing groups when moving over the mouse not needed." (Test User 2)

> "...the bar is too sensitive to accident moving." (Test User 7)

> "it is quite easily to inadvertantly change the selected segment of the timeline when moving the mouse" (Test User 12)

Noticed among the user comments were comments which highlighted both good and bad points of this widget at the same time:

> "...it is very easy to move the mouse into an adjanent section by accident. But, the same feature is very handy when you do want to move to the adjacent section." (Test User 4)

As briefly mentioned in **Quick Bar** browser above, this leads to the need for a better widget that can be quick but less sensitive to inadvertent mouse movement. Some form of scroll bar where dragging the block immediately changes the keyframe only when the mouse button is pressed while dragging would be one solution to this. One idea suggested by Test User 7 was a timeline bar with inverted trapezoid-shape segments rather than rectangular, thus reducing the chance of the mouse inadvertently touching the neighbouring segment, as illustrated in Figure 7.5 and 7.6 below.

Figure 7.5 - Normal Quick Bar with rectangular segments. When the user brings the mouse cursor down to keyframes, it is easy to touch a neighbouring segment

Figure 7.6 - A modified timeline bar with inverted trapezoid-shape segments to reduce inadvertent mouse movement when the user brings the cursor down to keyframes

SMALL SIZE OF OVERVIEW KEYFRAMES

As in **Overview/Detail** browser, the size of each keyframe was smaller than in detail view, and was mentioned as being too small.

> "I felt that the keyframes in the overview browser were too small but again I don't want to see to many keyframes on one screen..." (Test User 5)

> "The images in the detail view are a better size than in the overview." (Test User 12)

Considering the frequency of this complaint, this might be an obvious and simple usability problem that can be improved. Using a slightly larger size for each keyframe, and probably making the mouse-pointed keyframe larger and back to small when the mouse pointer moves away, it would be possible to resolve this problem (this was the idea commented by Test User 6 on using **Overview/Detail** browser). A simpler and focused user testing on the size of the keyframes would be useful.

Scroll Bar browser

MORE ORGANISATION REQUIRED

It seemed that a more elaborate, refined arrangement of keyframes is perceived as better than a rather "raw" arrangement as in this browser. Also this seems to bring the sense of having too much information on one screen, whereas in fact the actual amount of information is not very different from, for example, **Timeline Bar**. Some test users referred to this browser as showing "too much."

> "Negative features: + not a hierarchical, not organised browser." (Test User 18)

> "There are an awful lot of pictures on one page, and there must be a shot taken at pretty regular intervals... feeling of information-overload..." (Test User 19)

Although it seems that the large number of keyframes on a single page makes a user feel overloaded, it was preferred by some users for search efficiency - one can browse large number of keyframes very quickly using the scroll bar (**Scroll Bar** was classified as a static browser in our framework, but quickly responds to action when one drags the block on the scroll bar with a mouse). As noticed in many informal discussion and also some test users' comments as:

> "[**Scroll Bar** is better than **Timeline Bar** because it's quicker to browse.] (verbal discussion)" (Test User 17)

> "As for the task, it took much longer to complete on this browser [**Timeline Bar**] than it would have taken on the previous browser i.e. Browser 1 [**Scroll Bar**]." (Test User 19)

Slide Show browser

CONTROL MECHANISM REQUIRED

Overall opinions on **Slide Show** browser was very low. The definite and obvious reason was its constantly changing keyframes. Whereas it allows less user interaction with less mouse clicking or movement, having no user-based control over the pace of keyframe flipping was considered as a definite problem. This point was mentioned in numerous informal comments after the sessions and also in several typed comments:

> "**...**speed of the frame change was too fast for me... some kind of system which would allow me to control the rate of frame change." (Test User 9)

> "...as it [**Slide Show**] relise on reaction time you are more likely to click in the neighbourhood..." (Test User 13)

> "The constant movement is unrestful and distracting to the eye. An improvement in my view would be to have the browser move frames only when the mouse button is clicked/held down over it." (Test User 15)

The idea of being unable to have more control over the flipping speed was also frequently mentioned with **Dynamic Overview**'s flipping feature (see analysis of **Dynamic Overview** below). Having a temporal presentation by automatically flipping through keyframes, although expected to let the user passively watch without requiring a frequent interaction, seems to be more on the disadvantage side. But it could be more related to the nature of the task given in this evaluation - searching for a particular scene. If the given task was to simply get an overall gist of the video content rather than finding out a particular scene, it might require less desire for flipping control and thus test users might have given less negative feedback on the browser. The possible problem of the finding-out task as given to users will be mentioned later in this section.

Hierarchical browser

INITIAL SURPRISE AND PREFERENCE

Having never used such a novel and immediately responding browser before, people also showed their initial liking to the browser (this was similar to the users' preference to **Dynamic Overview**). It seems that the idea of hierarchically arranged pictures quickly responding to his/her mouse movement is in itself an attractive and interesting feature, though it often caused confusion and difficulty in knowing how to use.

> "...putting the mouse gives a lot more detail which is excellent and unexpected." (Test User 3)

> "I like the fact that you browse faster and more efficiently with this browser than with browser I [**Dynamic Overview**]." (Test User 5)

DIFFICULTY IN UNDERSTANDING HOW TO USE

Although often amused and impressed by the browser, users seemed to have difficulty understanding the way keyframes were presented in this browser - probably because they have never seen this kind of interactive, hierarchically arranged picture browser before. In many of the informal think-aloud sessions, discussions and some observation during the evaluation, it was noticeable that users were browsing the beginning part of the video when they were actually looking for a scene closer to the ending, and vice versa, indicating the difficulty in understanding the structural arrangement in this browser. As some test users also commented:

> "It seems to be more difficult to follow this browser." (Test User 16)

"...very (opposite of obvious) [unclear] to use and too many images and boxes opening when using it." (Test User 4)

"It was harder to find things [than in **Dynamic Overview** and **Overview/Detail+Quick**]." (Test User 20)

Dynamic Overview browser

NOVEL AND INTERESTING

This browser was considered overall very favourably by many people during initial discussions as well as in the evaluation, which was quite surprising because it had a rather artificially designed look and might look complicated to a first time user. Perhaps it is more related to the interesting and new way of interaction that people liked about this browser, as was often the case with **Hierarchical** browser. As Test User 15 commented:

"I've used the Fischlar system before and I feel that the browser which looks like an overview of the whole program [**Dynamic Overview**] - but each scene actually moves when you hold the mouse over it - is excellent, it's my favourite by far."

NEED FOR THE KEYFRAME FLIPPING CONTROL

Negative comments on this browser were mainly about the flipping speed not being controllable - this is basically the same problem mentioned by most people who used **Slide Show** (see above), but less negatively commented in this browser, possibly because the feeling of control is present to a certain degree here as the flipping of keyframes can be stopped and started by positioning the mouse cursor over the keyframe, also because the length of the flipping loop in each segment is usually shorter than in **Slide Show** - once a keyframe is passed, the user can wait for a while until the next loop comes.

"...picture flips maybe a bit too quickly and it is easier to get lost than the other two browsers [**Overview/Detail+Quick** and **Hierarchical**]..." (Test User 3)

"The interface took over and I had to passively (and not too passively becuase if I binked I'd miss it) watch for the scene which is hard to recognise... more seeking [mechanism needs to be] supported in that interface " (Test User 1)

"...if the user loses a picture, it has to wait for the next occurence..." (Test User 7)

But one test user mentioned this as a positive point that the automatic flipping *saved clicking*, which was partly the rationale for presenting keyframes temporally:

> "I like the way that on this browser it skipped through the images - so there was no need for extra & annoying clicking." (Test User 20)

TEMPORAL OVERVIEW AND SPATIAL DETAIL VIEW INSTEAD OF SPATIAL OVERVIEW AND TEMPORAL DETAIL VIEW

An interesting point was made by Test User 12, on the reverse arrangement of spatial and temporal presentation: the **Dynamic Overview** browser has its overview spatially presented occupying the whole screen, whereas the detail view is provided temporally when the user brings the mouse cursor over any of the overview keyframes. This browser has the mixed presentation in the way of spatial overview and temporal detail view. Consider another browser that presents a set of keyframes on the screen spatially, then flips through the next set, and then the next, temporally presenting sets of detailed keyframes. This would be another interesting mixture of spatial and temporal presentation in a single browser - this can be understood as a **Slide Show** browser having multiple keyframes on each flipping, or **Timeline Bar** browser with automatic segment flipping feature.

7.5.2 Overall Físchlár Browsers Analysis

ENTERTAINING NATURE OF VIDEO BROWSING

It was felt that the video browsers usually engage the test users into the task well, as the data displayed are images taken from a movie content and users are normally interested in such entertainment. So sometimes it is easy to identify usability problems when the user was not engaged in the browsing task and rather fiddling with the widgets on the browser. For most browsers which have spatial presentation with a large amount of information content displayed rather than complicated widgets and indicators, it was felt that the browsers were often information-rich and the information itself (keyframes) worked as an interaction trigger. Several users wanted to watch more of the test video material after completing the given task. The ultimate goal of the evaluation and consequent refinement would be to develop the browsers (and any other user interface elements) to such an extent that the user can concentrate on the browsing task and forget the interface handling side. As [Norman 88, p180] says, "the best computer programs are the ones in which the computer itself disappears, in which you work directly on the problem without having to be aware of the computer."

DIFFERENT BROWSER FEATURES AS OPTIONS, OR AS AN INTEGRATED SINGLE BROWSER

Because some people see the potential usefulness of the presented browsers' features, suggestions were made on the use of several different browsers together as options. Also addressing the good points of presented browsers, suggestions were made on combining the good points of different browsers into a single browser.

"The general overview feature at the beginning could be applied to browsers nos 1 & 2 [**Overview/Detail** and **Timeline Bar**] as well." (Test User 14)

"From the point of view of locating a particular scene, it might be nice to have the option of reverting temporarily to a full list of images [**Scroll Bar**] like in browser one [**Timeline Bar**]." (Test User 19)

"I think the other 2 Browsers [**Hierarchical** and **Dynamic Overview**] work fine as options to the 1st browser [**Overview/Detail+Quick**], they would probaby be preferred in different situations..." (Test User 16)

"Use the 'overview' concept present in browser 3 [**Overview/Detail**] with the 'detail' concept present in browser 2 [**Overview/Detail+Quick**] and combine them both with the 'motion' concept in browser 1 [**Slide Show**] and you'd have a cracking good browser." (Test User 15)

As more and more test users' comments were received, it became clear that features can be liked by some and disliked by others at the same time. There does not seem to be one single feature that is liked by everybody. **Quick Bar**'s immediate reaction style was liked by many but there were people who did not like it. Automatic flipping of keyframes in **Dynamic Overview** was condemned by many people for its uncontrollable pace of flipping, but liked by others for not requiring annoying mouse clicking. There were people who were surprised and impressed at the novel and highly interactive feature of **Hierarchical** browser, but there were also people who did not like it for its difficulty in helping to perform the task of finding a video segment. All these conflicting comments from different users tell us that opting for one particular browser is not the best way of interface provision, and rather we should provide a way to flexibly switch between different features in some way. It can be said that being able to select the right interface to the right user is the key in a successful strategy. Being able to customise the interface, the provision of options, and ultimately the intelligent provision of the right interface to the right user performing the appropriate task, would be some directions for this idea.

MORE SMOOTH BROWSING-PLAYBACK TRANSITION OR INTEGRATION

The idea was suggested of making the Físchlár browsers more closely related to the video player in terms of user interaction.

> "[In **Slide Show**]...it needs to be more integrated with the windows media player - being able to show the stills while dragging the media player scroll bar would help a lot." (Test User 13)

> "[In **Overview/Detail**, I want to see] The correspondence of player and browser with respect to the current frame." (Test User 7)

> "[In **Dynamic Overview**] It would be very nice to have actual playback or at least synchronization between playback in the frame and in the player." (Test User 7)

> "[In **Dynamic Overview**]... play back the movie instead of the image... to allow showing of the content of a shot... (Test User 12)

When considering video browsing in the context of the users' whole experience, *browsing and playback should not be really separate, independent interactions* even though this thesis has dealt exclusively with browsing as quite separated from playback. The user browses keyframes and plays at an interesting point, then wants to go back to browsing, then play again, and so on. A more complete interface that helps the user browse the video content would blur the boundary of browsing and playback. Or it can be said that playback should be understood as one part of browsing. The idea of a more integrated browser-player especially in terms of synchronising browsing with playback is also found in some literature [Simpson-Young & Yap 96] [Mc Donald *et al.* 01], and the above user comments received in this evaluation are some glimpse into their wish to have a more seamlessly integrated browsing-playback experience.

INITIAL DIFFICULTY IN UNDERSTANDING HOW TO USE

As mentioned earlier, the Físchlár browsers are not particularly targeted for the first time, infrequent user, but still the initial learning time is something worth considering and noted as a characteristic in different Físchlár browsers. Browsers that use a timeline bar or simple **Slide Show** were relatively easily understood in terms of how to use them (timeline is something which people normally relate to playing video, and is already a familiar concept), but novel browsing interfaces such as **Hierarchical** and **Dynamic Overview** and even the idea of overview and detail were often noticed as confusing to get the initial understanding when first exposed to users. Most especially, the **Hierarchical** browser was one of the most difficult ones as the users often looked at the wrong portion of the content, and were confused about the beginning and the end of a video (see analysis on **Hierarchical** browser above). As one test user mentioned:

> "If an overview or some brief instruction at the top of these browsers [**Hierarchical** and **Dynamic Overview**] were used it might make them more usable for people not used to the system." (Test User 3)

QUESTIONS ON HOW WELL KEYFRAMES REPRESENT CONTENT

In the initial informal demonstrations and discussions while new Físchlár browsers were being implemented, people were often curious about how the keyframes were selected. A brief explanation on automatically detecting camera shots and selecting one frame from each camera shot was given, and people were impressed by the "smart" technique which the system uses. However, the way keyframes are selected and especially the way selected keyframes are arranged in multiple layer browsers are always in question. As Test User 5 commented:

> "The keyframes shown might not be representative of the content."

It is not obvious how we can measure the correctness of the right choice of keyframes and right structuring. Especially in multiple layer browsers (**Overview/Detail**, **Overview/Detail+Quick**, **Hierarchical** and **Dynamic Overview**), the way groupings are done for a more concise keyframe set on the higher layers is not currently based on any standard technique. This potential pitfall can be quite visible when the grouping becomes very concise such as in the **Hierarchical** browser where the top layer should present only 6 keyframes which supposedly represent the whole video content. The current Físchlár system does not contain any elaborate scene grouping method that does this more intelligently although this is an active area of research within the Centre for Digital Video Processing at Dublin City University. Although this concern is more on the technical side, being able to display truly representative keyframes is very much influential to users' browsing behaviour. In this thesis, however, we put aside the question of how representatively the keyframes are selected and filtered for higher layers, partly due to the technical status of this area, partly to concentrate more on the presentation and interaction side of the user interface and partly because it is an active area of research addressed elsewhere and outside the scope of this thesis. As mentioned in Chapter 3, the development is underway to group the detected camera shots thus further and select better keyframes for higher layers such as the overview. As these new developments are implemented, the Físchlár system will incorporate them and thus will be able to provide more representative keyframes with better structuring.

OTHER COMMENTS

Several test users who had seen the test video prior to the evaluation mentioned in their comments that if they have not seen the video, it would have been difficult to find the scene. Other interesting comments from test users were:

- Detect advertisement in the video and do not show them on the browser (Test User 15)
- Show text description on each keyframe when the mouse pointer is on (Test User 6)

These are also possible features that the Físchlár system will incorporate as more technical developments are achieved and implemented in the future.

SHORT SUMMARY

The following is a short list of findings from the evaluation that can be understood in a more general sense:

- Timeline is useful and easily understood.
- Overview is useful.
- User control is important.
- If a control widget is less dynamic, provide its suitable interaction mechanism to make up for this (e.g. arrows buttons in **Timeline Bar** to reduce mouse movement).
- Do not squeeze in too many keyframes on one screen.
- Invent a control widget that quickly responses to user actions but prevents inadvertent changes by mistake.
- Consider providing different or conflicting features as options, rather than opting for only one.
- Work on underlying techniques to enrich or better abstract video content (e.g. better keyframe selection algorithm, better scene grouping methods, advertisement detection, speech recognition or teletext display for extra information along with keyframes).

7.5.3 Consideration Regarding the Design Space and User Classification

Comments and opinions analysed in the evaluation were useful for finding out the usability problems of each of the Físchlár browsers in detail (such as the size of each keyframe being too small), and helped in coming up with some more generally applicable ideas in designing keyframe based browsers. From these findings we also re-considered some of the concerns in the browser design framework with its dimensions and values, and the way users were classified and the way they were matched to the browser attributes. The following are the details in regard to this.

FURTHER DISTINCTION IN STATIC/DYNAMIC ATTRIBUTES OF THE BROWSING INTERFACES

Frequently mentioned is the problem of control of keyframe flipping when a temporal presentation is used (in **Slide Show** and **Dynamic Overview**), and it is quite obviously understood that users' browser preferences would have been quite different if there was a flipping speed control mechanism. For example, if the **Slide Show** browser was revised to have a keyframe flipping controller in some way, the user would interact with the browser far more than with the current version. This will likely make **Slide Show** browser fall into having the Dynamic attribute rather than Static. However, in **Dynamic Overview** browser (which we categorised as Dynamic already) if we revise it to have a keyframe flipping controller for each of the overview keyframes, it can become an even more Dynamic browser, with the user positioning the mouse cursor on an interesting keyframe, then using some kind of controller to either manually flip through or pausing/resuming automatic flipping. It might even depend on exactly what way the flipping controlling is provided. Overall, it will make more sense to further distinguish the degree of Static-ness and Dynamic-ness, rather than having a simple binary categorisation. This would make the design alternative check list in Chapter 5 more complicated. We might have to consider further elaborating the interaction style as a complete new dimension similar to the keyframe dimension space. For example, [Laurel 93] divided "level of interaction" into three dimensions:

- Frequency of the interaction
- Significance of the interaction
- Range of the interaction

Revising the Static/Dynamic attribute might start with further making the distinction between the system's presentation and the user's interaction. For example, **Slide Show** browser is currently classified as a Static browser, but in fact we could further state that it has Dynamic presentation (automatically flipping through keyframes by itself) and Static interaction (user does not require any action).

Scroll Bar browser was classified as Static browser, as the screen is static and overall it is suited more to users with a Static interaction style preference. However, the positive comments on the browser were mainly about its quick browsability, when the user drags the block on the scroll bar and quickly moves up and down. In this sense, the **Scroll Bar** has both characteristics of Static and Dynamic attributes, and there might be a need for another category for this kind of browser.

USER PREFERENCE CAN BE TAKEN OVER BY THE TASK EFFICIENCY

During the evaluation, the tasks given for the users to perform on each of the browsers were specifically finding out a particular scene in the video where something happens. For example, on presenting the second browser, the given task was to find out:

"The scene where a military tank runs over the exploding cars with fire"

This particular task of finding out a scene might entail the user's inherent preference for a certain browsing strategy/style. For example, a user who prefers a Static interaction style in general might still be frustrated at the frequent clicking required in **Timeline Bar** to find out the scene which s/he does not know where in the video it might be found. If presented with a **Quick Bar**, the finding-out task can be very efficiently completed as the browser allows very quick browsing of all the keyframes. For another example found in user comments on **Slide Show** browser, a user simply wanting to get a rough gist of the video content might be happy with the current **Slide Show** with no flipping control mechanism, whereas if the same user needs to find out a particular scene in the video s/he might get frustrated at the fact that there is no control mechanism (as this evaluation revealed). Having a particular *task context*, the user's preference for a browser becomes not only a matter of his/her inherent or general preference but is inevitably connected to that task context at hand. This tells us that inclusion of some kind of task classification into the currently constructed framework would better represent the whole picture. Or it could be understood that a user's preference in the current framework is actually the one *after* the user perceives a particular task, thus being of changeable nature, rather than inherent and constant to a user.

Whether personal preferences or task efficiency becomes more important depends on many different factors, including the person and the task themselves. Consider a file-copying task where the user wants to copy a file from one disk drive to another. For this same task, person A might prefer using a GUI interface by selecting a file with a mouse and dragging it to another drive, whereas person B might prefer using a command line interface by typing in file copy commands with arguments. With this particular task, person A and person B will initially use the interface they prefer. If the task becomes copying multiple files with a certain file attribute, for example, file extension name called ".doc", person A might start feeling uncomfortable as selecting all the files with .doc extension can be time-consuming and tedious if the directory is large, whereas person B can still complete the task with same amount of effort by using wild card names with a command such as "copy *.doc". Will person A then abandon his/her own preference and want to use a command line interface to get the task done more efficiently? It partly depends on the person A - s/he may be the kind of person who wants to stick to their own preference even at the cost of lower task efficiency, or the kind of person who easily adapts the interface that allows quickest task completion, regardless of his own preference. It also partly depends on the task itself - there might be a limit where person A still wants to stick to his own preference while being aware of the lower task efficiency, but if the task makes the use of GUI interface unacceptably inefficient (for example, working on 1,000 similarly named files to check), after a certain point person A might just decide to switch to the command line interface. The nature of the task and the user's characteristics together seem to determine which interface a person will want to use eventually. The priority could be given for different usage context, too, for example an office environment where work efficiency is more important than personal preferences compared to a home entertainment environment where the purpose is mainly to enjoy spending time for fun.

7.5.4 Problems of the Evaluation Design

From the analysis of the evaluation result, a few problems regarding the evaluation procedure design were found.

SEGMENTING THE VIDEO MATERIAL INTO 3 PARTS

User's knowledge about the video content (i.e., whether s/he has seen the video or not) was one of the parameters that determined what 3 browsing interfaces to present in the evaluation. If the user sees the same video content in the second browser test as the first, his knowledge on the video content already has changed because of having been acquainted with it while doing the first browser test, and in the third test he will have become even more familiar with it, possibly having an idea beforehand on the whereabouts of a scene asked for in that third test. Because of this, the same video's different parts were presented for the three browsing tests - in this way, it was expected that the user's knowledge about the video (asked at the User Type identification stage) holds correct throughout the three browsing tests. However, by dividing the video into three, timelines for each browsing test were sometimes perceived as confusing - the timeline bar was supposed to represent the whole length of the video, but in fact it represented 1/3 of the video in each browser. Browsers showing absolute time were also problematic. For example, the timeline bar of the second browser started with 0 minute 0 seconds, but in fact the content was about 30 minutes into the video. Although most test users' comments were general enough not to complicate the usefulness of widgets under test ("timeline is useful because it shows my current browsing point"), some confusion was noticed. For example, comments from Test User 10 were mainly about the confusion caused by dividing the video into three parts:

> "...dislikes: - the arrow buttons (in the top right corner) [in **Timeline Bar**], do these allow you to navigate through the third of the film or between each third of the film or both, use not obviously clear... suggestions:- is it possible to navigate an overview of the entire film?"

The reason for this confusion was the design of the evaluation procedure, from the need for having a constant user's knowledge on the video.

KEYFRAME LOADING DELAY

Before starting each evaluation session, all the keyframes were pre-loaded to allow quicker image loading while the user browses keyframes. However, cache memory management on the user's machine is not under direct control by us and thus sometimes delays in keyframe loading were noticed. It would have been possible that a user felt less satisfaction when the second browser loaded keyframes more slowly than the first. Also, browsers using the mouse-over timeline bar (i.e. **Quick Bar** browser and **Overview/Detail+Quick**) took a few seconds at the beginning of loading each video, whereas other browsers did not. This is the result of technical incompleteness in browser implementation and hardware and network speed, which we could not control.

USER TYPE 5 WAS NEVER TESTED

In the evaluation, a user is recruited and the User Type is identified *during* the test session, rather than before. The number of each identified User Types was as follows:

Table 7.5 - Number of test users in each User Type

User Type	Number of test users
User Type 1	1
User Type 2	3
User Type 3	7
User Type 4	4
User Type 5	0
User Type 6	1
User Type 7	1
User Type 8	3
Total	20

As can be seen, User Type 5 (a user who has not seen the video, prefers a Linear browsing strategy and prefers Static interaction style) was not tested at all. Given that the goal of the evaluation is the testing of the 8 Físchlár browsers, not covering all the 8 User Types did not create that much of a problem. More important was to get all the browsers tested at least once.

After the 16th user testing, the allocation of the three browsers were modified as the **Scroll Bar** browser had not been tested at all by any of the users (this is mentioned in Section 7.4 in this chapter), so an alternative allocation schedule had to be made and used after the 16th user, to make the **Scroll Bar** browser more likely to be one of the three test browsers.

PREFERENCE BIAS TOWARDS THE GIVEN TASK

During the testing, the tasks were given for each of the three Físchlár browsers, namely a finding-out task for a particular scene. The purpose of the task was to get the users to more realistically assess the browsers, rather than simply allowing them to "browse" (the term "browsing" is rather broad and refers more to a physical activity instead of some purposeful cognitive activity). However, this particular task of finding out a particular scene might be more suitable for some of the browsers than others. Having done this task, there is a chance that the test user might say they preferred the browsers that helped in the finding-out task more efficiently, rather than the one which should be their best general preference. This complexity of the task and user preference were mentioned in Section 7.5 above.

7.6 Usability Testing of the Whole System

In this chapter, user testing on the comparative evaluation of eight Físchlár browsers and the nature of their similarity has been checked, the main concern being individual browsers themselves. Although the whole interaction of a complete digital video system is outside the scope of this work, if we are to better understand the future digital video library and the interaction with this in more detail, we should go about user testing of a complete system interaction - starting with a user coming into the system, initiating the querying/browsing of the library collection, then locating a video item, browsing the content of that item, going back to the library, playing clips of an item, and all the transition styles among these stages. A system has to be understood as a whole, and thus the user testing of usability also needs to be conducted as part of the whole information seeking behaviour of a user, instead of at any particular stage - for example content browsing as this work has focused on - in its segregated state. In terms of finding out usability problems from the interface (as is done with individual browsers in some part of this chapter), there are different methods developed [Nielsen 95], such as heuristic evaluation [Nielsen 93, p155] [Nielsen 92] [Nielsen 94], cognitive walkthrough [Wharton *et al.* 92], think-aloud [Nielsen 93, p195] [Dumas & Redish 99, p278], and several experiments have been done to compare the strengths of these methods [Karat *et al.* 92] [Savage 96] [Jeffries *et al.* 91]. The results of these comparative studies show that different methods tend to find different kinds and severity of usability problems, thus using some of the methods together has often been suggested. The Físchlár system's current full interface - not only the individual browsers as done in this thesis but also the whole interface including the TV programme recording interface, recorded programme listing and the transition between these elements - will benefit from these usability inspection methods.

Conventionally, the purpose of usability testing of a complete system is to see how well the task specification is satisfied by actually allowing test users to try the tasks. This is especially important in evaluating IT products in office environments where very focused, special, optimised tasks already exist are identified and systems developed accordingly. Often a rough indication of the usability of a system is simply found from the popularity and frequency of its usage, and often this becomes a measure of overall success. The *degree of a system's success or failure*, though, can be subtle and difficult to capture such as small degrees of inefficiency at work and minor irritations. To more precisely assess the usability of a system, some evaluation with test users based on some *specific measures* could be conducted to find out about not only the system performance but also to get ideas on improving the system (as emphasised in [Newman 97] and experimented on in numerous studies such as [Brajnik *et al.* 96] [Hersh *et al.* 96] [Jose *et al.* 98] [Koenemann & Belkin 96]).

In video browsing systems and other future technologies where exact tasks are still to be explored by users and more new tasks will be developed by the users themselves (as have happened with spreadsheet software and others), it is not possible to set an optimised task and run an evaluation according to that task. Often it is suggested in the literature that for future systems where tasks are ill-defined, drawing out some imaginary but plausible scenario would be a useful thing [Nielsen 93, p99] [Shneiderman 98, p111] at the start of a project.

It might even be the case that we should *not* set a pre-conceived, assumed task and try to measure systems accordingly. Rather, we should design the system as open and flexible as possible, so that a wide range of possible, unknown future tasks could be accommodated as the technology starts to settle down. This idea of an open approach to design [Stary 00] is more and more recognised as important, as is the diverse nature of users themselves, the possible task environment, unpredictable contexts and the expansion in the range of devices coming out.

Currently underway (though out of the scope of this work) is to deploy the complete Físchlár system to the university's campus residences and allow residence students to freely use the system. Without any particular narrow-focused task or intention from system developers, our expectation is that the student users will use the system in their own way to fit their own purposes, while we observe this usage through interaction logging, interview and survey.

Once a system is used by actual users, the feedback from them can provide valuable clues on how to go about improving the system. TiVo's lively user feedback site [TiVo Guide] is a good example of trying to capture the usage data from actual users, and it incorporates the features these feedback items cry for. Some initial ethnographic studies with set top boxes have also been carried out. Qualitative studies of the impact and use of new technology as the recent ethnographic studies done for understanding the social context of the impact of a set top box at home [O'Brien *et al.* 99] [Lee 00] also provide important insights into the usage and context of such a system.

It would also be useful to conduct user satisfaction surveys with some standardised questionnaires such as QUIS (Questionnaire for User Interaction Satisfaction) [Harper & Norman 93] and MUMMS (Measuring the Usability of Multi-Media Systems) [MUMMS]. Subjective user satisfaction is getting more and more weight as one of the important measures in attempting to evaluate an IR system, rather than the retrieval effectiveness measures [Furner 96].

Eventually, all the different levels of system evaluations (such as input level, output level and social level, as categorised in [Saracevic 95]) would have to be continuously conducted to be able to fully understand and assess a system over a long term basis.

7.7 Summary

This chapter conducted an evaluation of the 8 Físchlár video browsers. The procedure and schedule were designed based on the design space and user classification constructed from the previous chapters. The purpose of the evaluation was to obtain users' comments and ideas on the individual browsers, thus finding out usability problems for future refinement, also on comparative ideas among different browsers to gain further insight into more fundamental problems, elements and features of keyframe browsers in general. Twenty test users participated in the evaluation, individually working step by step using a self-contained, automated evaluation package specially designed for the purpose. This identified the current user's User Type as classified in the previous chapter, by asking questions, then depending on the identification presented 3 of the 8 Físchlár browsers. Test users provided comments and ideas that were both colourful and varied as well as similar. The analysis of the comments highlighted that test users often have different ideas and contrasting preferences on the same features, signifying the fact that a single interface cannot satisfy everyone - it shows the importance of being able to customise or providing options for different features to be accessible when a user wants. The separate interaction of browsing and playback was also highlighted, users wishing to have a more integrated, whole experience in browsing and playback - a further browser design concern will have to include a playback side into the browsing interaction. Also, for the browsing interface to be more trustworthy, the underlying techniques of video indexing such as shot boundary detection, keyframe selection, scene detection and grouping will have to be further investigated and implemented - to provide a better keyframe representation and hierarchy on the browsing interface. Although the initial design framework with the design space is a useful tool for designing and thinking about keyframe browsers, further finer-level option sets might enhance the closeness to the reality.

CHAPTER 8

Conclusions

When we say that new technology opens up new possibilities, that includes completely new activities and tasks which have not existed before, but which the new technology could support. The invention of TV created a completely new task unheard of only 50 years ago - the task of sitting on the sofa looking at the TV set for hours every day, and in the process changing people's living patterns. The invention of the remote controller has also changed the TV watching behaviour into more frequent, restless channel changing throughout the watching time, while comfortably sitting on the sofa. Instead of shaping new tasks as new technology is developed, maybe it is time to think first about the new tasks that could be useful and enriching for our lives, and then try to think of the technology to support these new tasks. In such a scenario, the technological possibilities that computer developers come up with could go more hand in hand with people's values and objectives, and cause less of "techno-phobia" and frustration arising from the mismatch between people's expectations and what is provided for them. As mentioned in [Gaver & Martin 00], perhaps we should start thinking about what are the values that we have in life, other than work efficiency and entertainment which the current technology-driven development seems to be only focusing on.

The main theme of this thesis, video browsing, is an example of one such new task that has become possible and realised because new technological possibilities are in place to support it now. From the point of view of our research, this newness of the area has meant that there are scarce example systems and their user interface ideas, not well-established, long-term user base or usage environment, and a constantly changing new technology itself which could make any in-depth study of the current application less useful. All these factors led our research into following directions:

- Continued observation and review of whatever small number of projects, systems, products in the field of digital video browsing and their user interface side;

- A generic and holistic design approach where we develop a possible design space rather than a specific instance, so that it can be applied flexibly in many possible situations.

Taking this approach, this thesis analysed and identified important elements of keyframe-based video browsing interfaces based on the current state-of-the-art, and constructed a design space specially for designing such an interface. From this design space a number of browsing interfaces were specified and implemented, and possible application ideas were demonstrated, including the browsing interfaces on a PDA device, and different browsing interfaces supporting different individual user characteristics. The usefulness of the design space lies in the fact that it allows us to come up with various interfaces for these different application areas. User-testing was conducted for several browsing interfaces whose design was based on the design space, to capture the ideas, feedback and opinions of people. The collected ideas and opinions were analysed and understood in terms of the constructed design space.

This point brings us back to the thesis we proposed at the end of Chapter 5. Just after the design space was completed, a thesis was proposed that this constructed design space was reasonably complete and comprehensive. Eight browsing interfaces were designed by selecting various combinations of values from the design space, and the user evaluation was conducted to prove the thesis. The evaluation results show that simple single layer presentation and layered presentation with its overview feature in different combinations of widgets, time indication in its relative and absolute values, spatial and temporal presentation and their different combinations, and all in static or dynamic interaction styles, were addressed and pointed out by test users as “good” or “bad”, using their own terminology. This illustrated that when presented with browsers with different combinations of these values, the users showed conflicting preferences which a single interface or a single feature set cannot satisfy. This implies that these different values identified in our analysis and addressed in our evaluation can, and should, exist together in the design space, to be selected when a proper browser is designed, depending upon the situation required. A possible defect of the design space identified from the evaluation is the lack of finer specificities in some options. The multiple layer with navigational link probably needs further elaboration considering the frequent user comments on wishing to have the navigational link when presented with the browsers without it. Also the static/dynamic interaction styles probably need further division or distinction within themselves, and this matter has been pointed out in regard to the user-browser matching concerns. This potential specificity problem was in fact considered and mentioned at the time of the construction of the design space and elsewhere, but still the simpler structure was insisted on the basis that it helped the initial construction to be clearer and that it could be further elaborated in the future.

Another important point illuminated from the user evaluation is the need for a more integrated browsing-playback interaction. Maybe there should be a more comprehensive browsing interface design framework where playback can be addressed as one integral element of the browsing. However, the Físchlár system where we based all our user interface experiments provides keyframe-based browsing as an independent, separate interface from the playback interface, and indeed, playback concern is out of the scope of our work and of the current design space framework. Further technical and hardware development which can allow an internal video playback mechanism which is computationally lighter and faster will make experimentation on this aspect possible.

Thus, when strictly limited to the keyframe-based browsing interaction only, our thesis on the completeness and comprehensiveness of the design space seems quite proven. Although in this research only 8 browsing interfaces for the desktop and 2 for the PDA were actually implemented and demonstrated, more precise assessment on the completeness and comprehensiveness of the design space would occur as more and more different designs for different contexts are implemented from this design space and evaluated.

In designing any user interface for any computer system, many different disciplines need to be considered. It is obvious that it helps if the designer him/herself is knowledgeable and competent in graphic design, psychology, human perception as well as sensitive to people's subjective ideas and feelings. In the current situation with interface design, where few concrete design theories are available, maybe it is still important for the designer to be aware of and equipped with these supporting disciplines. Graphic design experts do actively address and provide useful tips for computer user interface design [Tufte 83] [Tufte 90] [Tufte 97] and there are plenty of elements that computer user interface designers can learn and take from them. For example, the graphic design theory from [Dondis 74] was ported in detail and used to design a better computer user interface screen in [Vanderdonckt & Gillo 94]. Gestaltist theories from cognitive psychology [Eysenck & Keane 95, p33] is often used in designing a computer user interface for effective layout, emphasis, font size, colour and lines, and has become background knowledge for heuristic evaluation methods [Nielsen 93, p117]. Also, it helps to be knowledgeable in the currently available design heuristics and general guidelines which have been accumulated through designers' experiences.

However, for each application area, elaborated, focused and in-depth design consideration will have to be made. Maybe it comes down to the need to paying far more attention, elaboration and effort in the user interface design side of a particular system under development. As other novel video browsing ideas are realised and implemented, a specially elaborated consideration of those interfaces will have to be made. For example, in Chapter 3 an automatic movie trailer generation was mentioned, as one of the small number of new video content browsing ideas. How should a user interact with this video summary? Surely there are different ways of providing interaction mechanisms – we could provide a way to pause and play the generated movie trailer, maybe allowing a user to jump back and forth the beginning of scenes within the trailer, allowing faster playback of the trailer, allowing different lengths of trailers to be played based on the user's choice and preferences, some form of spatial presentation, and so on. When the technology allows us to automatically generate a 5-minute movie trailer from a 2-hour original movie, it is time for us to think in a great detail on how this technology could be provided and presented to the user, rather than merely providing a playback of the 5-minute trailer on the screen. It will require its own analysis and identification of the important dimensions and design alternatives, as done in this thesis. In part, it is hoped that this thesis demonstrated the point that building such design spaces is a useful part of interface design. Although hoped useful for other video browsing interfaces, the constructed design space in this thesis applies only to the keyframe-based content browsing interface. We do not consider this particularly a weak point of the work. Different applications and browsing methods require and deserve their own, specialised consideration in design methodology suitable to their own individual characteristics. This includes the consideration of every detailed level, including particular widgets and their behaviour suitable for the browsing method, and possible interaction styles on currently available operational systems.

It is true that the whole user-system experience has to be considered soon, as mentioned in Chapter 7. Individually designed content browsing interfaces should be fully integrated into the whole system, with its video collection searching features, playback and browsing features, as well as with its overall operating system and eventually with the place where the system is used. Focusing on every level of the user interface provision, from its individual elements, to the transition among them, and to the whole usage environment, will eventually have to be covered. There is a long way to go in building this overall picture, but the starting step will be by trying to understand each of the basic levels of a user interface.

References

[Ahanger *et al.* 95] Ahanger, G., Benson, D. and Little, TDC. Video query formulation. *Proceedings of SPIE,* **2420***, Storage and Retrieval for Image and Video Databases III*, San Jose, CA, 5-10 February, 1995, 280-291.

[Ahlberg & Shneiderman 94] Ahlberg, C. and Shneiderman, B. Visual information seeking: tight coupling of dynamic query filters with Starfield displays. *Proceedings of the ACM Conference on Human Factors in Computing Systems: "Celebrating Independence" (CHI '94),* Boston, MA, 24-28 April, 1994, 313-317.

[Aigrain *et al.* 95] Aigrain, P., Joly, P., Lepain, P. and Longueville, V. Representation-based user interfaces for the audiovisual library of year 2000. *Proceedings of SPIE,* **2417***, Multimedia Computing and Networking*, San Jose, CA, 5-11 February, 1995, 35-45.

[Akoumianakis *et al.* 00] Akoumianakis, D., Savidis, A. and Stephanidis, C. Encapsulating intelligent interactive behaviour in unified user interface artefacts. *International Journal on Interacting with Computers, special issue on "The Reality of Intelligent Interface Technology",* **12**(4), 383-408.

[Aoki *et al.* 96] Aoki, H., Shimotsuji, S. and Hori, O. A shot classification method of selecting effective key-frames for video browsing. *Proceedings of the 4nd ACM International Conference on Multimedia (MM '96),* Boston, MA, 18-22 November, 1996, 1-10.

[Arman *et al.* 94] Arman, F., Depommier, R., Hsu, A. and Chiu, M-Y. Content-based browsing of video sequences. *Proceedings of the 2nd ACM International Conference on Multimedia (MM '94),* San Francisco, CA, 15-20 October, 1994, 97-103.

[Arons 97] Arons, B. SpeechSkimmer: a system for interactively skimming recorded speech. *ACM Transactions on Computer-Human Interaction*, **4**(1), 1997, 3-38.

[Ashley *et al.* 95] Ashley, J., Barber, R., Flickner, M., Hafner, J., Lee, D., Niblack, W. and Petkovic, D. Automatic and semiautomatic methods for image annotation and retrieval in query by image content (QBIC). *Proceedings of SPIE,* **2420**, *Storage and Retrieval for Image and Video Databases III*, San Jose, CA, 5-10 February, 1995, 24-35.

[Bawden 90] Bawden, D. *User-oriented evaluation of information systems and services.* Gower, 1990.

[Besser 90] Besser, H. Visual access to visual images: the UC Berkeley image database project. *Library Trends*, **38**(4), 1990, 787-798.

[Boreczky *et al.* 00] Boreczky, J., Girgensohn, A., Golovchinsky, G., Uchihashi, S. An interactive comic book presentation for exploring video. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2000),* The Hague, Netherlands, 1-6 April, 2000, 185-192.

[Brajnik *et al.* 96] Brajnik, G., Mizzaro, S. and Tasso, C. Evaluating user interfaces to information retrieval systems: a case study on user support. *Proceedings of the 19th Annual ACM International Conference on Research and Development in Information Retrieval (SIGIR '96)*, Zurich, Switzerland, 18-22 August, 1996, 128-136.

[Browne *et al.* 00] Browne, P., Smeanton, A., Murphy, N., O'Connor, N., Marlow, S. and Berrut, C. Evaluating and combining digital video shot boundary detection algorithms. *4th Irish Machine Vision* *and Image Processing Conference (IMVIP 2000),* Belfast, Northern Ireland, 31 August - 2 September, 2000.

[Calvary *et al.* 00] Calvary, G., Coutaz, J. and Thevenin, D. Embedding plasticity in the development process of interactive systems. *Proceedings of the 6th ERCIM Workshop "User Interfaces for All"*, Florence, Italy, 25-26 October, 2000, 171-176.

[Chang *et al.* 97] Chang, S., Chen, W., Meng, H., Sundaram, H. and Zhong, D. VideoQ: an automated content based video search system using visual cues. *Proceedings of the 5th ACM International Conference on Multimedia (MM '97)*, Seattle, WA, 9-13 November, 1997, 313-324. Project WWW site at URL: http://www.ctr.columbia.edu/VideoQ/

[Christel *et al.* 97] Christel, M., Winkler, D. and Taylor, C. Multimedia abstractions for a digital video library. *Proceedings of the 2nd ACM International Conference on Digital Libraries (DL '97)*, Philadelphia, PA, 23-26 July, 1997, 21-29.

[Christel *et al.* 99] Christel, M., Hauptmann, A., Warmack, A. and Crosby, S. Adjustable filmstrips and skims as abstractions for a digital video library. *Proceedings of the IEEE Conference on Advances in Digital Libraries*, Baltimore, MD, 19-21 May, 1999.

[Christel & Martin 98] Christel, M. and Martin, D. Information visualization within a digital video library. *Journal of Intelligent Information Systems*, **11**(3), 1998, 235-257.

[Compton & Bosco 95] Compton, C. and Bosco, P. Internet CNN NEWSROOM: A digital video news magazine and library. *IEEE International Conference on Multimedia Computing and Systems (ICMCS '95)*, 1995, 296-301.

[Deng *et al.* 98] Deng, Y., Mukherjee, D. and Manjunath, B. NeTra-V: towards an object-based video representation. *Proceedings of SPIE*, **3312**, *Storage and Retrieval for Image and Video Databases VI*, San Jose, CA, 24-30 January, 1998, 202-213.
Project WWW site at URL: http://copland.ece.ucsb.edu/Demo/video/

[Dimitrova 95] Dimitrova, N. The myth of semantic video retrieval. *ACM Computing Surveys*, **27**(4), 1995, 584-586.

[Dimitrova *et al.* 97] Dimitrova, N., McGee, T. and Elenbaas H. Video keyframe extraction and filtering: a keyframe is not a keyframe to everyone. *Proceedings of the 6th ACM International Conference on Information and Knowledge Management (CIKM '97)*, Las Vegas, NV, 10-14 November, 1997, 113-120.

[Dondis 74] Dondis, D. *A primer of visual literacy*. MIT Press, 1974.

[Dunlop & Mc Donald 00] Dunlop, M. and Mc Donald, K. Supporting different search strategies in a video query interface. *RIAO '2000: Content-Based Multimedia Information Access*, Paris, France, 12-14 April, 2000, 21-31.

[Dumas & Redish 99] Dumas, J and Redish, J. *A practical guide to usability testing (Revised edition).* Intellect Books, 1999.

[DVB-VCR 98] DVB-VCR. In: *SMASH (Storage for Multimedia Applications Systems in the Home) Project final report* (1998).

[Elliott & Davenport 94] Elliott, E. and Davenport, G. Video Streamer. *Proceedings of the ACM Conference on Human Factors in Computing Systems: "Celebrating Independence" (CHI '94 Conference Companion)*, Boston, MA, 24-28 April, 1994, 65-66.

[Enser 95] Enser, P. Pictorial information retrieval. *Journal of Documentation*, **51**(2), 1995, 126-170.

[EUROMEDIA] EUROMEDIA. *Distributed Multimedia Archives for Cooperative TV Production*. Available online at URL: http://www.foyer.de/euromedia/home.html (last visited November 2000)

[Evans 87] Evans, A. TELCLASS: a structural approach to TV classification. *Audiovisual Librarian*, **13**(4), 1987, 215-216.

[Excalibur] Excalibur Technologies corporation. *Screening Room.* Available online at URL: http://www.excalib.com/products/sr/index.shtml (last visited November 2000)

[Eysenck & Keane 95] Eysenck, M and Keane, M. *Cognitive psychology: a student's handbook (3rd ed).* Lawrence Erlbaum, 1995.

[Furner 96] Furner, J. Session 3.3.1. The evaluation of hypermedia IR systems: a statement of the problems. In: Dunlop, M. (ed) *Proceedings of the 2nd Mira Workshop*, Monselice, Italy, 14-15 November, 1996.

[Gaver & Martin 00] Gaver, B. and Martin, H. Alternatives: exploring information appliances through conceptual design proposals. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2000),* The Hague, Netherlands, 1-6 April, 2000, 209-216.

[Graham *et al.* 00] Graham, T C N., Watts, L., Calvary, G., Coutaz, J., Dubois, E. and Nigay, L. A dimension space for the design of interactive systems within their physical environments. *Proceedings of the ACM Symposium on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS 2000),* Brooklyn, NY, 17-19 August, 2000, 406-416.

[Grund 93] Grund, A. ICONCLASS: on subject analysis of iconographic representations of works of art. *Knowledge Organization*, **20**(1), 1993, 20-29.

[Harrison & Vicente 96] Harrison, B. and Vicente, K. An experimental evaluation of transparent menu usage. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '96),* Vancouver, Canada, 13-18 April, 1996, 391-398.

[Harper 96] Harper, D. Session 4.2 Evaluation light. In: Dunlop, M. (ed) *Proceedings of the 2nd Mira Workshop,* Monselice, Italy, 14-15 November, 1996.

[Harper & Norman 93] Harper, B. and Norman, K. Improving user satisfaction: the questionnaire for user interaction satisfaction version 5.5. *Proceedings of the 1st Annual Mid-Atlantic Human Factors Conference,* Virginia Beach, VA, 1993, 224-228.

[Healey 00] Healey, N. Chapter 6. The EPOC user interface in the Psion Series 5. In: Bergman, E. (ed) *Information appliances and beyond: interaction design for consumer products*. Academic Press, 2000, 131-168.

[Hearst 95] Hearst, M. TileBars: visualization of term distribution information in full text information access. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '95),* Denver, CO, 7-11 May, 1995, 59-66.

[Hearst 99] Hearst, M. Chapter 10. User interfaces and visualization. In: Baeza-Yates, R. and Ribeiro-Neto, B. *Modern information retrieval.* Addison Wesley Longman, 1999, 257-323.

[Hersh *et al.* 96] Hersh, W., Pentecost, J. and Hickam, D. A task-oriented approach to information retrieval evaluation. *Journal of the American Society for Informaion Science*, **47**(1), 1996, 50-56.

[Hindus *et al.* 95] Hindus, D., Arons, B., Stifelman, L., Gaver, B., Mynatt, E. and Back, M. Designing auditory interactions for PDAs. *ACM Symposium on User Interface Software and Technology (UIST '95)*, Pittsburg, PA, 15-17 November, 1995, 143-146.

[Hjelsvold *et al.* 95] Hjelsvold, R., Lagorgen, S., Midtstraum, R. and Sandsta, O. Integrated video archive tools. *Proceedings of 3rd ACM International Conference on Multimedia (MM '95)*, San Francisco, CA, 5-9 November, 1995, 283-293.

[Iyengar & Lippman 98] Iyengar, G. and Lippman, A. Semantically controlled content-based retrieval of video sequences. *Proceedings of the SPIE, 3527, Multimedia Storage and Archiving Systems III*, Boston, MA, 1-6 November, 1998, 223-232.

[Jeffries *et al.* 91] Jeffries, R., Miller, J., Wharton, C. and Uyeda, K. User interface evaluation in the real world: a comparison of four techniques. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '91)*, New Orleans, LA, 27 April - 2 May, 1991, 119-124.

[Jose *et al.* 98] Jose, J., Furner, J. and Harper, D. Spatial querying for image retrieval: a user-oriented evaluation. *Proceedings of the 21st Annual ACM International Conference on Research and Development in Information Retrieval (SIGIR ' 98)*, Melbourne, Australia, 24-28, August, 1998, 232-240.

[Kamba *et al.* 96] Kamba, T., Elson, S., Harpold, T., Stamper, T. and Sukaviriya, P. Using small screen space more efficiently. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '96)*, Vancouver, Canada, 13-18 April, 1996, 383 - 390.

[Karat *et al.* 92] Karat, C-M., Campbell, R. and Fiegel, T. Comparison of empirical testing and walkthrough methods in user interface evaluation. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '92)*, Monterey, CA, 3-7 May, 1992, 397-404.

[Kiristoffersen & Ljungberg 99] Kristoffersen S and Ljungberg F. "Making place" to make IT work: empirical explorations of HCI for mobile CSCW. *ACM Conference on Supporting Group Work (GROUP '99)*, Phoenix, AZ, 14-17 November, 1999, 276-285.

[Koenemann & Belkin 96] Koenemann, J. and Belkin, N. A case for interaction: a study of interactive information retrieval behavior and effectiveness. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '96)*, Vancouver, Canada, 13-18 April, 1996, 205-212.

[Komlodi & Marchionini 98] Komlodi, A. and Marchionini, G. Key frame preview techniques for video browsing. *Proceedings of the 3rd ACM International Conference on Digital Libraries (DL '98)*, Pittsburgh, PA, 23-26 June, 1998, 118-125.

[Krause 88] Krause, M. Intellectual problems of indexing picture collections. *Audiovisual Librarian*, **14**(2), 1988, 73-81.

[Kreyss *et al.* 97] Kreyss, J., Roper, M., Alshuth, P., Hermes, T. and Herzog, O. Video retrieval by still image analysis with ImageMiner. *Proceedings of SPIE,* **3022**, *Storage and Retrieval for Image and Video Databases V*, San Jose, CA, 8-14 February, 1997, 36-44.

[Laurel 93] Laurel, B. *Computers as theatre.* Addison Wesley, 1993.

[Lee 00] Lee, W. Introducing Internet terminals to the home: interaction between social, physical, and technological spaces. *Proceedings of the 14th Annual Conference on Human Computer Interaction (HCI 2000),* Suntherland, UK, 5-8 September 2000, 119-132.

[Lee *et al.* 00a] Lee, H., Smeaton, A., O'Toole, C., Murphy, N., Marlow, S. and O'Connor, N. The Físchlár digital video recording, analysis, and browsing system. *RIAO '2000: Content-Based Multimedia Information Access,* Paris, France, 12-14 April, 2000, 1390-1399.

[Li *et al.* 96] Li, W., Gauch, S., Gauch, J. and Pua, K. VISION: a digital video library. *Proceedings of 1st ACM International Conference on Digital Libraries (DL '96),* Bethesda, MD, 20-23 March, 1996, 19-27.

[Li *et al.* 00] Li, F., Gupta, A., Sanocki, E., He, L-W. and Rui, Y. Browsing digital video. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2000),* The Hague, Netherlands, 1-6 April, 2000, 169-176.

[Lienhart *et al.* 97] Lienhart, R., Pfeiffer, S. and Effelsberg, W. Video abstracting. *Communications of the ACM*, **40**(12), 1997, 54 - 62.

[MacLean *et al.* 89] MacLean, A., Young, R., Moran, T. Design rationale: the argument behind the artifact. *Proceedings of the ACM Conference on Wings for the Mind (SIGCHI '89),* Austin, TX, 30 April - 4 May 1989, 247-252.

[Marchionini 95] Marchionini, G. Information seeking in electronic environments. *Cambridge Series on Human-Computer Interaction*, Cambridge University Press, 1995.

[Marchionini *et al.* 98] Marchionini, G., Plaisant, C. and Komlodi, A. Interfaces and tools for the library of congress National Digital Library Program. *Information Processing & Management*, **34**(5), 1998, 535-555.

[Marcus *et al.* 98] Marcus A, Ferrante J, Kinnunen T, Kuutti K and Sparre E. Baby faces: user-interface design for small displays. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '98),* Los Angeles, CA, 18-23 April, 1998, 96-97.

[McCracken 88] McCracken, G. The long interview. *Qualitative Research Methods Series 13*. Sage Publications, 1988.

[Mc Donald *et al.* 01] Mc Donald, K., Smeaton, A., Marlow, S., Murphy, N. and O'Connor, N. Online television library: organisation and content browsing for general users. *SPIE Electronic Imaging: Storage and Retrieval for Media Databases 2001*, San Jose, CA, 2001.

[MediaSite] MediaSite. *MediaSite Publisher 4.0.*
Available online at URL: http://www.islip.com/corporateweb/products/publisher.html (last visited November 2000)

[Media Player] Microsoft Corp. *Windows Media Player.*
Available online at URL:
http://www.microsoft.com/windows/windowsmedia/en/software/Playerv7.asp (last visited November 2000)

[Mills *et al.* 92] Mills, M., Cohen, J. and Wong, Y-Y. A magnifier tool for video data. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '92),* Monterey, CA, 3-7 May, 1992, 93-98.

[Mills *et al.* 00] Mills, T., Pye, D., Hollinghurst, N. and Wood, K. AT&TV: broadcast television and radio retrieval. *RIAO '2000: Content-Based Multimedia Information Access,* Paris, France, 2000, 1135-1144.

[Molholt & Petersen 93] Molholt, P. and Petersen, T. The role of the 'Art and Architecture Thesaurus' in communicating about visual art. *Knowledge Organization*, **20**(1), 1993, 30-34.

[MUMMS] *MUMMS. Measuring the Usability of Multi-Media Software.*
Available online at URL: http://www.ucc.ie/hfrg/questionnaires/mumms/ (last visited November 2000)

[Myers *et al.* 00] Myers, B., Lie, K. and Yang, B-C. Two-handed input using a PDA and a mouse. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2000),* The Hague, Netherlands, 1-6 April, 2000, 41-48.

[Newman 97] Newman, W. Better or just different? On the benefits of designing interactive systems in terms of critical parameters. *Proceedings of the ACM Symposium on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '97),* Amsterdam, The Netherlands, 18-20 August, 1997, 239-245.

[Nielsen 92] Nielsen, J. Finding usability problems through heuristic evaluation. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '92),* Monterey, CA, 3-7 May, 1992, 373-380.

[Nielsen 93] Nielsen, J. *Usability Engineering*. Academic Press, 1993.

[Nielsen 94] Nielsen, J. Enhancing the explanatory power of usability heuristics. *Proceedings of the ACM Conference on Human Factors in Computing Systems: "Celebrating Independence" (CHI '94),* Boston, MA, 24-28 April, 1994, 152-158.

[Nielsen 95] Nielsen, J. Usability inspection methods. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '95),* Denver, CO, 7-11 May, 1995, 377-378.

[Norman 88] Norman, D. *The psychology of everyday things*. BasicBooks, 1988.

[O'Brien *et al.* 99] O'Brien, J., Rodden, T., Rouncefield, M. and Hughes, J. At home with the technology: an ethnographic study of a set-top-box trial. *ACM Transactions on Computer-Human Interaction*, **6**(3), 1999, 282-308.

[O'Connor *et al.* 01] O'Connor, N., Marlow, S., Murphy, N., Smeaton, A., Browne, P., Deasy, S., Lee, H. and Mc Donald, K. Físchlár: an on-line system for indexing and browsing of broadcast television content. Submitted to the *26th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001),* Salt Lake City, UT, 7-11 May, 2001.

[O'Toole *et al.* 99] O'Toole, C., Smeaton, A., Murphy, N. and Marlow, S. Evaluation of automatic shot boundary detection on a large video test suite. *The Challenge of Image Retrieval (CIR '99): 2nd UK Conference on Image Retrieval,* Newcastle, UK, 25-26 February, 1999.

[Potosnak *et al.* 86] Potosnak, K., Hayes, P., Rosson, M., Schneider, M. and Whiteside, J. Classifying users: a hard look at some controversial issues. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '86),* Boston, MA, 13-17 April, 1986, 84-88.

[QuickTime] Apple Computer, Inc. *QuickTime.*
Available online at URL: http://www.apple.com/quicktime/ (last visited November 2000)

[Rao *et al.* 95] Rao, R., Pedersen, J., Hearst, M., Mackinlay, J., Card, S., Masinter, L., Maivorsen, P-K. and Robertson, G. Rich interaction in the digital library. *Communications of the ACM*, **38**(4), 1995, 29-39.

[RealPlayer] RealNetworks, Inc. *RealPlayer.*
Available online at URL: http://www.real.com

[ReplayTV] *ReplayTV.*
Available online at URL: http://www.replaytv.com

[Sanderson 97] Sanderson, D. Technology design and mimicry. *Proceedings of the ACM Symposium on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '97),* Amsterdam, The Netherlands, 18-20 August, 1997, 311-313.

[Saracevic 95] Saracevic, T. Evaluation of evaluation in information retrieval. *Proceedings of the 18th Annual ACM International Conference on Research and Development in Information Retrieval (SIGIR '95),* Seattle, WA, 9-13 July, 1995, 138-146.

[Savage 96] Savage, P. User interface evaluation in an iterative design process: a comparison of three techniques. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '96 Conference Companion),* Vancouver, Canada, 13-18 April, 1996, 307-308.

[Shatford 86] Shatford, S. Analyzing the subject of a picture: a theoretical approach. *Cataloging & Classification Quarterly*, **6**(3), 1986, 39-62.

[Shatford 94] Shatford, S. Some issues in the indexing of images. *Journal of the American Society for Information Science*, **45**(8), 1994, 583-588.

[Shneiderman *et al.* 97] Shneiderman, B., Byrd, D. and Croft, B. Clarifying search: a user-interface framework for text searches. *D-Lib Magazine,* January 1997.

[Shneiderman 98] Shneiderman, B. *Designing the user interface: strategies for effective human-computer interaction (3rd ed).* Addison Wesley Longman, 1998.

[Simpson-Young & Yap 96] Simpson-Young, B. and Yap, K. FRANK: trialing a system for remote navigation of film archives. *Proceedings of SPIE,* **2915**, *International Symposium on Voice, Video and Data Communications*, Boston, MA, November 18-22, 1996, 126-135.

[Smeaton *et al.* 99] Smeaton, A., Gilvarry, J., Gormley, G., Tobin, B., Marlow, S. and Murphy, N. An evaluation of alternative techniques for automatic detection of shot boundaries in digital video. *3rd Irish Machine Vision and Image Processing Conference (IMVIP '99),* Dublin, Ireland, 8-9 September, 1999.

[Smith 96] Smith, J. Searching for images and videos on the World-Wide Web. *Technical report #459-96-25*, Center for Telecommunications Research, Columbia University, 1996.

[Smith 97] Smith, A. *Human-Computer Factors: a study of users and information systems*. McGraw-Hill, 1997.

[Smith *et al.* 00] Smith, D., Paradice, D. and Smith, S. Prepare your mind for creativity. *Communications of the ACM*, **43**(7), 2000, 111-116.

[Smyth & Cotter 00] Smyth, B. and Cotter, P. A personalized television listings service. *Communications of the ACM*, **43**(8), 2000, 107-111.

[Stary 00] Stary, C. A structured contextual approach to design for all. *Proceedings of the 6th ERCIM Workshop "User Interfaces for All"*, Florence, Italy, 25-26 October, 2000, 83-97.

[Taniguchi *et al.* 95] Taniguchi, Y., Akutsu, A., Tonomura, Y. and Hamada, H. An intuitive and efficient access interface to real-time incoming video based on automatic indexing. *Proceedings of the 3rd ACM International Conference on Multimedia (MM '95),* San Francisco, CA, 5-9 November, 1995, 25-33.

[TiVo] *TiVo.*
Available online at URL: http://www.tivo.com (last visited November 2000)

[TiVo Guide] iwantptv.com. *TiVo Guide*.
Available online at URL: http://www.iwantptv.com/tivo/ (last visited November 2000)

[Teodosio & Bender 93] Teodosio, L and Bender, W. Salient video stills: content and context preserved. *Proceedings of the 1th ACM International Conference on Multimedia (MM '93),* Anaheim, CA, 2-6 August, 1993, 39-46.

[Tonomura 97] Tonomura, Y. Chapter 7. Multimedia interfaces - multimedia content indication.. Grosky, W., Jain, R. and Mehrotra, R. (edit). *The Handbook of multimedia informaiton management*. Prentice Hall, 1997, 189-210.

[Tse *et al.* 99] Tse, T., Vegh, S., Marchionini, G. and Shneiderman, B. An exploratory study of video browsing user interface deisgn and research methodologies: effectiveness in information seeking tasks. *Proceedings of Annual Meetings of the American Society for Information Science (ASIS '99),* 1999, 681-692.

[Tufte 83] Tufte, E. The visual display of quantitative information. Graphics Press, 1983.

[Tufte 90] Tufte, E. Envisioning information. Graphics Press, 1990.

[Tufte 97] Tufte, E. Visual explanations. Graphics Press, 1997.

[Turner 94] Turner, J. Indexing "Ordinary" pictures for storage and retrieval. *Visual Resources*, **10**, 1994, 265-273.

[Uchihashi *et al.* 99] Uchihashi, S., Foote, J., Girgensohn, A. and Boreczky, J. Video Manga: generating semantically meaningful video summaries. *Proceedings of the 7th ACM International Conference on Multimedia (MM '99),* Orlando, FL, 30 October - 5 November, 1999, 383-392.

[van der Zwan *et al.* 99] van der Zwan, R., Kukulska-Hulme, A., Dipaolo, T., Evers, V. and Clarke, S. The Informedia Digital Video Library System at the Open University. *The Challenge of Image Retrieval (CIR '99): 2nd UK Conference on Image Retrieval,* Newcastle, UK, 25-26 February, 1999.

[Vanderdonckt & Gillo 94] Vanderdonckt, J. and Gillo, X. Visual techniques for traditional and multimedia layouts. *Proceedings of the ACM Workshop on Advanced Visual Interfaces (AVI '94),* Bari, Italy, 1-4 July, 1994, 95-104.

[Veerasamy & Belkin 96] Veerasamy, A. and Belkin, N. Evaluation of a Tool for Visualization of Information Retrieval Results. *Proceedings of the 19th Annual ACM International Conference on Research and Development in Information Retrieval (SIGIR '96),* Zurich, Switzerland, 18-22 August, 1996, 85-92.

[VICAR] *VICAR project - Video Indexing, Classification, Annotation and Retrieval.*
Available online at URL: http://iis.joanneum.ac.at/vicar/ (last visited November 2000)

[Virage] Virage Inc. *VideoLogger 4.0*.
Available online at URL: http://www.virage.com/products/videologger.html (last visited November 2000)

[Wactler *et al.* 96] Wactlar, H., Kanade, T., Smith, M. and Stevens, S. Intelligent access to digital video: Informedia project. *Computer*, **29**(5), 1996, 46-52.

[Wharton *et al.* 92] Wharton, C., Bradford, J. and Franzke, M. Applying cognitive walkthroughs to more complex user interfaces: experiences, issues, and recommendations. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '92)*, Monterey, CA, 3-7 May, 1992, 381-388.

[Wolf 96] Wolf, W. Key frame selection by motion analysis. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '96),*, Atlanta, GA, 7-10 May, 1996, 1228-1231.

[Wolf *et al.* 96] Wolf, W., Liang, Y., Kozuch, M., Yu, H., Phillips, M., Weekes, M. and Debruyne, A. A digital video library on the World Wide Web. *Proceedings of the 4th ACM International Conference on Multimedia (MM '96)*, Boston, MA, 18-22 November, 1996, 433-434.

[Yeo & Yeung 97] Yeo, B-L. and Yeung, M. Retrieving and visualizing video. *Communications of the ACM*, **40**(12), 1997, 43-52.

[Zhang *et al.* 95] Zhang, H., Low, C., Smoliar, S. and Wu, J. Video parsing, retrieval and browsing: an integrated and content-based solution. *Proceedings of 3rd ACM International Conference on Multimedia (MM ’95)*, San Francisco, CA, 5-9 November, 1995, 503-512.

Papers Published on This Work

Lee, H., Smeaton, A. and Furner, J. User-interface issues for browsing digital video. *Proceedings of the 21st Annual Colloquium on IR Research (IRSG '99)*, Glasgow, UK, 19-20 April, 1999.

Lee, H., Smeaton, A., O'Toole, C., Murphy, N., Marlow, S. and O'Connor, N. The Físchlár digital video recording, analysis, and browsing system. *RIAO '2000: Content-Based Multimedia Information Access*, Paris, France, 12-14 April, 2000, 1390-1399.

Lee, H., Smeaton, A., Berrut, C., Murphy, N., Marlow, N. and O'Connor, N. Implementation and analysis of several keyframe-based browsing interfaces to digital video. *Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2000)*, Lisbon, Portugal, 18-20 September, 2000, 206-218.

Lee, H., Smeaton, A., McCann, P., Murphy, N., O'Connor, N. and Marlow, S. Físchlár on a PDA: a handheld user interface to a video indexing, browsing and playback system. *Proceedings of the 6th ERCIM Workshop "User Interfaces for All"*, Florence, Italy, 25-26 October, 2000, 352-353.

O'Connor, N., Marlow, S., Murphy, N., Smeaton, A., Browne, P., Deasy, S., Lee, H. and Mc Donald, K. Físchlár: an on-line system for indexing and browsing of broadcast television content. Submitted to the *26th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, Salt Lake City, UT, 7-11 May, 2001.

APPENDIX A

The Design of the Evaluation Package

1 Introduction: Special Care in Designing Evaluation Software

Most guidelines on general user-interface design apply to evaluation package design, as an evaluation package is also *a kind of system that a user interacts with*. Generally, evaluation package design does not seem to be elaborated enough in the literature probably because developers do not think such software as a proper end-product. However it is obvious that designing an evaluation package should get even more attention and care than normal software in its user-interface design, partly because the users of such software ("test users") have different attitudes or mindsets than normal software users and also because the evaluation result should not be contaminated with unnecessary side effects caused by the evaluation software's interface usability problems. Special considerations are:

- Extra care has to be taken in verbal messages: instructions have to be present at every stage of the session, but have to be brief, polite, neutral (rather than using 'I' or 'you', as suggested by [Shneiderman 98, p383]).
- Make an effort to make the session a pleasant and wholesome experience to the test user. A clear indication of starting/ middle/ ending can enhance this, as TV shows do [Laurel 93]).
- Avoid too fancy a design but try a simple and tidy design.
- Indicate at the beginning roughly how long the session will be. During the session, always indicate the whereabouts the test user is in and roughly how much more s/he will be doing.

This design tried to apply available design guidelines available from graphic design (which partly borrows ideas from cognitive psychology) and user-interface design, and took particular care of the above points.

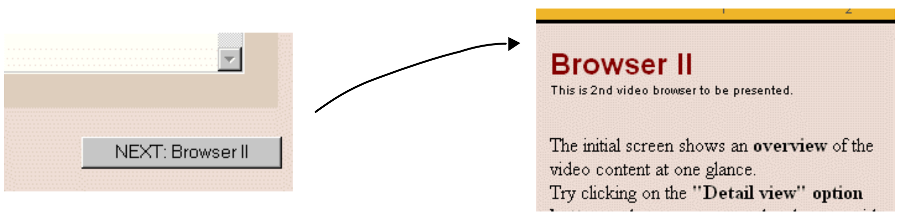
2 Indication of Progress

Respondents of a test package or online survey are often frustrated when the procedure goes on and on and on, question after question, without knowing how near the end is. Even quitting in the middle can feel so bad with the suspicion that the survey might be very near the end and the effort put in up to that point may be considered as wasted time. Thus it becomes even more important in an evaluation package to show the current stage of the whole session, and indicate clearly the progression of the session (this idea is similar the the navigation bar for web page design to provide navigational orientation, but instead of indicating what page s/he is in, an evaluation package should additionally indicate what has been done and what is left to be done, thus the same concept as a 'progress bar').

A form of navigation bar - but not navigable, only indicative - namely a *progress bar* should be present at all times during the evaluation indicating the current point and what has been done so far and what is left to be done. Using distinct (but not distracting) colours for the bar at the top of the screen can effectively provide low-key feedback at all times, indicating the progression. The designed progress bar for this evaluation (Figure A1 below) shows a self-evident colour scheme used to indicate the current point, the past and the future stages.

Figure A1 - Progress bar: indicates current stage (Question 3) of the session, what has been done and what has to be done using an effective and pleasant colour scheme

Also importantly, the flow of progression should be natural and connected (for example, the previous screen's NEXT button label should match the following screen's title) so that it provides the user with the *feeling of continuity* (see Figure A2 below).



(Button label of the previous screen)

(Title of the next screen)

Figure A2 - Continuity between screens

3 Titles, Instruction and Messages

Every screen displayed has been labelled with a short title on the top left. Much care has to be taken to reduce all the instruction and messages, so that a quick glance will be enough to know what each message is trying to say. Use of different font size, emphasising important words in sentences (by colours and bolding) were used, though not so much as to distract, to make the messages clear and quickly recognisable.

4 Error Handling

User-driven errors that can happen in the package are very few. Not filling one of the 7-point scales is one of those few errors that could happen. Instead of a harsh pop-up error message saying "not all questions answered!", it was designed to do simply nothing. When the user clicks the NEXT button and nothing happens, she will look at what she has done wrong and the unchecked scale is easy to notice. No single screen has more than 3 questions, which makes it very easy to notice if one of these are not answered.

5 Colours and Fonts

A very light greyish beige-warm colour is consistently used as the background colour throughout the screens, muted and calmed down [Tufte 90, p90] enough so that other more important elements can be easily emphasised. Typefaces are standard Times New Roman, and central term(s) in sentences are bolded, which make a sentence more easily scanned. The use of short enough sentences and highlighting of important words can make a sentence as *understood by a quick glance*, without having to actually read it.

6 Screen by Screen Design Details

The following are the detail of each screen design.

Starting screen

The starting page is important in the sense that it gives the initial impression of the evaluation package. Actual instructions and cautions are in the following 'Introduction' page. Bearing in mind that it is important to make this page very simple, the following verbal information is provided on this page:

- Thank you message;
- What this package is about;
- How long this session will take (about 30-40 minutes, but also mentioning to feel free to do longer if s/he wants);

- The bar at the top will show which stage you are in. When the yellow bar reaches the right side, that will finish the session;
- Interaction will be logged, but complete anonymity will be kept;
- Finally, it is the browsers that are under testing, not the user.

Introduction screen

The introduction page mentions actual procedural information and instructions. It will mention the following information:

- Very brief overview of the session (that it is composed of 3 questions, followed by 3 video browsers one by one, each with short questions to be answered, then summary questions at the end and finishing remarks), with a diagram showing the stage progression.
- At the end of each stage, click the 'NEXT' button to move on to the next stage.

Questions screens

These are screens to determine the test user's User Type. Layout of questions screens are as Figure A3. On the left of the screen, the instructions and questions are displayed. On the right, the example interfaces are displayed. Question 2 (to find out the test user's preference on browsing strategy) is as Figure A4 below.

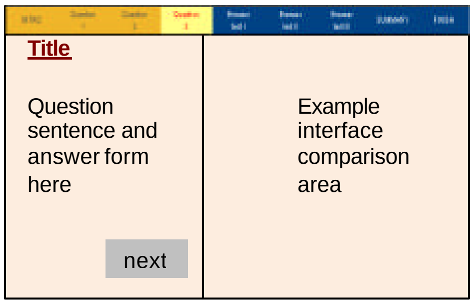


Figure A3 - Questions page layout

Figure A4 - Question 2 to determine
Linear/Structured browsing strategy preference

Question 3 (to find out the test user's preference on interaction style) is as Figure A5. A set of pictures are displayed as the user either clicks on or moves the mouse cursor over the titles.

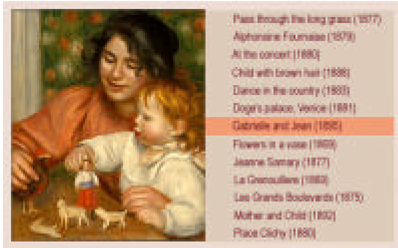


Figure A5 - Question 3 to determine Static/Dynamic interaction preference

Video browser screens

Each of the three video browser stages have 3 sub-sequences, where the right half of the screen is always the browser to be tested. The 3 sub-sequences are a simple introduction, a task assignment and post-task questions. These sub-sequences are illustrated in Figure A6 below:

1 Very short introduction to the browser

Right half of the screen shows the browser with example video content (different from the video for actual task performance checking).
Left half of the screen shows very short directions on how to use the browser, and asks to try it now.

2 Task assignment

When the test user clicks the NEXT button, the left half of the screen changes showing the task for the user to conduct.
Right half of the screen is the same browser, but the content changes to the prepared video.

3 Post-task questions

When the test user clicks the NEXT button, the left half of the screen changes now showing a set of questions for the browser. An open question reply text area was made as large as possible, to encourage the user to type in more.
The Right half screen is not changed from the previous screen.

Figure A6 - Three sub-sequences of video browser screens

Summary screen

The summary page shows all three browsers the user has used at one glance (see Figure A7). Small screen shots for each of the 3 browsers are displayed in a comparable way. Final questions are asked below these screen shots, about their preferences comparative to each other and any other comments. The answer text area was made large, to induce the user to type in as much as possible.

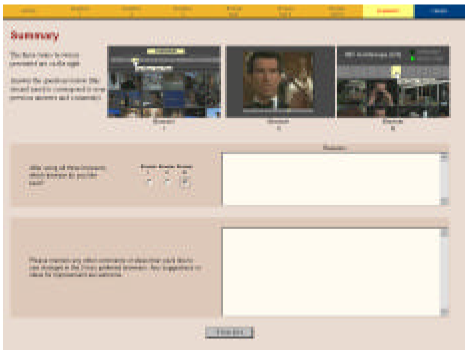


Figure A7 - Summary screen with 3 browsers tested shown

Finishing screen

This screen is important because it gives the final impression before the test user leaves. Again, a simple and polite tone has to be used. This is the only page in the whole package that *does not have a button*, as this is the last screen. The verbal information mentioned in this screen is:

- Now all tasks have been completed;
- We presented three video browsers and asked for your opinions;
- The gathered data will be analysed and used for developing better video browsers;
- Contact point (author's name and email address);
- Final thank you message;
- At the bottom, the affiliation of the package (Centre for Digital Video Processing, DCU), and the date.

7 Conclusion

User interface design requires a certain amount of craftsmanship. It requires careful consideration of all aspects of interface, from elements such as font colour and size, line width, location of a button, titles and labels, background colour, tone of voice in messages and instructions, harmonious effects of these elements when put together, to the layout of overall screen, sequence of screens and overall feeling the user gets throughout the interaction.

Designing the user-interface for an evaluation or test package needs extra special consideration above that for normal software. This design attempted to fulfil these ideas concentrating on detailed and careful crafting of the interface elements and combinations.