FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Mobile Metadata Logging System for Audiovisual Content

André Maciel Machado Miranda Duarte

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Maria Teresa Galvão Dias (PhD)

July 30, 2012

Mobile Metadata Logging System for Audiovisual Content

André Maciel Machado Miranda Duarte

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Doctor Jorge Manuel Gomes Barbosa

External Examiner: Doctor Paulo Miguel de Jesus Dias

Supervisor: Doctor Maria Teresa Galvão Dias

July 30, 2012

Abstract

From TV sets to tablets, from VCRs to Media Centers, informatics changed the way television is watched nowadays. But TV production, invisible to the main audience, was also revolutionized by a new paradigm that changed the tape-based systems into the new file-based systems. Among the many advantages of the digital world, one of the most important was the introduction of descriptive metadata into files. This data containing information about captured footage allowed a new level of organization that was impossible so far.

Nowadays, metadata is indispensable for tasks such as planning or archiving in the broadcast world. TV stations and production companies use metadata in order for their processes to achieve high levels of effectiveness. But sometimes efficiency can be compromised since most metadata needs to be manually inserted, representing an operational cost to these companies.

Developed in partnership with MOG Technologies, leader company in MXF file-based solutions for the broadcast and post-production world, this thesis aimed to investigate what are the existent problems in metadata logging nowadays and tried to develop a solution that would increase both effectiveness in the use of metadata and efficiency in its insertion.

Through the creation of a partnership with Rádio e Televisão de Portugal, the Portuguese state-owned TV channel and major news producer for television, we were able to observe how metadata is used in their daily tasks. Focusing our work in the news production ecosystem, we mainly observed how reporters, journalists and archivists interacted and made use of metadata to improve their work.

This investigation revealed many communication problems in the way these three entities communicate between them. They turned out to communicate very little and the little communication between the three took no advantage of metadata to swiftly interchange information about common resources. By adopting a user-centered design approach to this problem, we tried to find, together with the users, a solution for this problem.

Taking advantage of the mobile computing trend, we designed a solution that could be used by the three entities at the same time in a centralized yet mobile environment, providing metadata logging without the typical computer desktop configuration.

The use of touchscreen input methods also greatly improved efficiency in the logging activity of some kind of contents with repetitive events such as sports or politics, achieving an almost real-time logging where it used to take 8 hours of logging per hour of video.

Resumo

Dos televisores aos tablets, dos VCRs aos Media Centers, a informática mudou a forma como vemos televisão nos dias de hoje. Mas também o mundo da produção de televisão, invisível ao comum telespectador, foi revolucionado pelo novo paradigma que substituiu os sistemas baseados em cassete para os novos sistemas baseados em ficheiros. Entre as muitas vantagens do mundo digital, uma das mais importantes foi a introdução de metadados descritivos nos ficheiros de vídeo. Estes dados contendo informação sobre as imagens capturadas permitiram um novo nível de organização impossível até então.

Hoje em dia, no mundo do *broadcast*, estes metadados são indispensáveis para tarefas tais como planeamento ou arquivo. Estações de TV e produtoras usam metadados por forma a atingir altos níveis de eficácia nos seus processos. Mas por vezes a eficiência dos mesmos é posta em causa, dado que grande parte destes metadados requerem uma inserção manual, representando um custo operacional para estas empresas.

Desenvolvida em pareceria com a MOG Technologies, empresa líder em soluções baseadas em ficheiros MXF para o mundo do *broadcast* e pós-produção, esta tese teve como objectivo investigar quais os problemas na actividade de notação nos dias de hoje, bem como desenvolver uma solução que aumentasse a eficácia no uso de metadados e a eficiência na sua inserção.

Através da criação de uma pareceria com a Rádio e Televisão de Portugal, pudemos observar o modo como estes metadados são usados na prática. Focando o nosso trabalho na produção noticiosa, observamos principalmente a interacção entre reportéres, jornalistas e arquivistas e o uso que era dado aos tais metadados para melhorar o seu trabalho.

Esta investigação revelou muitos problemas de comunicação no modo como estas três entidades comunicam entre elas. Apercebemo-nos que elas comunicam muito pouco e o pouco que comunicam não tira qualquer partido dos metadados para partilhar com facilidade informação sobre recursos comuns. Através de uma abordagem centrada no utilizador, tentamos encontrar uma solução para este problema em conjuntos com os utilizadores.

Tirando partido das novas tendências para a computação móvel, desenhamos uma solução que pode ser usada pelas três entidades ao mesmo tempo, de um modo centralizado ainda que móvel, permitindo a actividade de notação sem a necessidade da típica configuração de secretária.

O uso de ecrãs táteis como método de introdução de dados permitiu também um grande avanço em termos de eficiência na anotação de certos tipos de conteúdos com eventos repetitivos, como futebol ou política, alcançando uma notação quase em tempo real onde era antes necessário 8 horas de anotação por cada hora de vídeo.

Acknowledgements

This dissertation would not be possible without the help and support of several people.

I would like to give my appreciation for my supervisor, Teresa Galvão, for her guidance throughout the work and all the help in defining and redefining the theme of this dissertation until we reached the final agreement. For the same reason, I would also like to thank Pedro Ferreira for kindly allowing the theme change from what was initially proposed by MOG Technologies.

In a thesis made in a company, working in a good team is fundamental. I would like to thank all my co-workers at MOG for their precious help, in particular to Ricardo Serra for supervising my work in the company and Miguel Nabuco for providing us the funniest moments in there.

I would also like to thank Luís Miguel Sampaio for putting us in contact with Rádio e Televisão de Portugal, as well as Professor Artur Pimenta Alves, director of RTP Porto, and Paulo Brandão who opened us all doors of the RTP studios. Also from RTP I must thank Daniel Catalão, an early supporter of this project, as well as all the archivists who so kindly exposed and explained all their work, with a special regard to Conceição Andrade, chief archivist of RTP Porto.

I must also thank my friends who kept insisting for me to have fun and saved me from working on most weekends.

But none of this would be possible without the tremendous help, both financially and emotionally, that my mother has been giving me all these years. For her goes my greatest appreciation.

Last but not least, I could not end this section without thanking Andreia, my biggest companion throughout this thesis, whose daily virtual company evolved into so much more.

André Maciel Machado Miranda Duarte

Contents

1 Introduction 1
1.1 Context 1
1.2 Motivation 2
1.3 Goals 2
1.4 Document Outline 2

2 State of the art 3
2.1 Television production 3
2.1.1 Tape-based workflows 4
2.1.2 File-based workflows 7
2.1.3 Related technology 9
2.2 Metadata 14
2.2.1 Metadata uses for television production 16
2.2.2 Metadata standards 18
2.2.3 Related technology 20
2.3 Mobile computing 21
2.3.1 Web-based applications 22
2.3.2 Sencha Touch 23
2.4 Interaction design 24
2.4.1 Usability 26
2.4.2 User Centered Design 27

3 Case study of RTP Porto 31
3.1 Introduction 31
3.2 News production at RTP 31
3.2.1 News shows 31
3.2.2 News pieces 32
3.2.3 News-related web clips 32
3.2.4 News-related shows 32
3.3 Main stakeholders 32
3.3.1 Reporters 33
3.3.2 Journalists 33
3.3.3 Archivists/documentalists 34
3.4 Typical workflows 35
3.4.1 General workflow 36
3.4.2 Request for archived footage 37
3.4.3 Raw footage archiving 38
3.4.4 News clips archiving 39

3.5 Problems . . . 40
3.5.1 Inefficiency . . . 40
3.5.2 Indexation problems . . . 44
3.5.3 Information loss . . . 44
3.5.4 Conclusions . . . 45

4 Proposed solution **47**
4.1 User Centered Design . . . 47
4.2 Requirements . . . 48
4.2.1 Speed Marking . . . 49
4.2.2 Integration with mxfSPEEDRAIL . . . 50
4.3 Low-fidelity prototyping . . . 51
4.3.1 Asset list . . . 51
4.3.2 Video preview . . . 51
4.3.3 Edit mode . . . 52
4.3.4 In-house heuristic testing . . . 54
4.4 Medium-fidelity prototyping . . . 54
4.4.1 User testing and redesign . . . 54
4.4.2 Login . . . 55
4.4.3 Asset list . . . 55
4.4.4 Preview . . . 57
4.4.5 Edit mode - General metadata . . . 60
4.4.6 Edit mode - Markers . . . 61
4.4.7 Edit mode - Speed Markers . . . 63
4.5 Conclusions . . . 72

5 Development **73**
5.1 Introduction . . . 73
5.2 System architecture . . . 73
5.3 Software architecture . . . 74
5.4 Implementation . . . 74
5.4.1 Problems . . . 77
5.4.2 HTML5 <video> tag . . . 77

6 Conclusions **79**
6.1 Accomplished goals . . . 79
6.2 Future work . . . 80
6.2.1 Desktop application for modifying SpeedMarker library . . . 80
6.2.2 Allow offline marking . . . 80

A RTP archive entries **83**
A.1 Old fully-manual archive cards . . . 83
A.2 Snapshots from Blue Order entries . . . 83

References **89**

List of Figures

2.1 Stages in creating a TV programme [Dev02] 3
2.2 Workflow of a tape-based system [DWBT06] 4
2.3 Assortment of video tapes [Wik12e] 5
2.4 RTP archive card from the 1980's 6
2.5 Workflow of a file-based system [DWBT06] 7
2.6 Screenshot of Apple's Final Cut Pro, one of the most popular NLEs 8
2.7 Overview of mxfSPEEDRAIL capabilities [Tec12b] 10
2.8 Screenshot of mxfSPEEDRAIL S1000 11
2.9 Screenshot of mxfSPEEDRAIL F1000 12
2.10 Apple's iPhoto screenshot full of metadata 15
2.11 Apple's iPhoto technical metadata 15
2.12 Avid's Interplay Assist screenshot with notes on shots 16
2.13 Screenshot of an EPG example 17
2.14 Screenshot of Avid's Interplay Cataloger 17
2.15 Avid Interplay Workflow 21
2.16 Examples of smartphones 22
2.17 Examples of tablets 22
2.18 Smartphone OS market share [New12] 22
2.19 Checkout, a Sencha Touch application 23
2.20 The Star lifecycle model 29

3.1 RTP Porto virtual studio 32
3.2 Cameraman and Reporter from RTP ready to start recording 33
3.3 Example of a synthetic resume 35
3.4 Example of an analytic resume 35
3.5 Example of indexing entry 35
3.6 General workflow between reporters, journalists and archivists 36
3.7 Workflow of archived footage request 37
3.8 Workflow of raw footage archiving 39
3.9 Example of indexing entry 40

4.1 Screenshot of Cinergy Media Desktop 49
4.2 New workflow integrating mxfSPEEDRAIL and our solution 50
4.3 First mockups of the initial asset list 51
4.4 First mockups of the asset list with the video preview 52
4.5 First mockups of the general metadata edit mode 53
4.6 First mockups of the markers edit mode 53
4.7 First mockups of the speed markers edit mode 54

4.8 Mid-fidelity prototype of login screen 56
4.9 Mid-fidelity prototype of asset list before user testing 56
4.10 Mid-fidelity prototype of asset list after user testing 56
4.11 Person handling a tablet with both hands 58
4.12 Mid-fidelity prototype of asset list with preview before user testing 58
4.13 Mid-fidelity prototype of asset list with preview after user testing 58
4.14 Mid-fidelity prototype of video preview showing markers before user testing . . . 59
4.15 Mid-fidelity prototype of video preview showing metadata after user testing . . . 59
4.16 Mid-fidelity prototype of video preview showing markers after user testing 59
4.17 Mid-fidelity prototype of general metadata editing before user testing 60
4.18 Mid-fidelity prototype of general metadata editing after user testing 61
4.19 Mid-fidelity prototype of markers editing before user testing 62
4.20 Mid-fidelity prototype of markers editing after user testing 62
4.21 Mid-fidelity prototype of initial speed marking screen before user testing 64
4.22 Mid-fidelity prototype of speed marker list builder before user testing 64
4.23 Mid-fidelity prototype of speed marker list builder adding a collection of markers before user testing 65
4.24 Mid-fidelity prototype of speed marker list builder after adding a collection of markers, one by one, before user testing 65
4.25 Mid-fidelity prototype of speed marker disclosure before user testing 66
4.26 Mid-fidelity prototype of speed marker list before user testing 66
4.27 Mid-fidelity prototype of initial speed marking screen after user testing 68
4.28 Mid-fidelity prototype of speed list browsing after user testing 68
4.29 Mid-fidelity prototype of speed folder creation after user testing 69
4.30 Mid-fidelity prototype of speed list creation after user testing 69
4.31 Mid-fidelity prototype of speed marker ready to start marking after user testing . 70
4.32 Mid-fidelity prototype of speed marker disclosure after user testing 70
4.33 Mid-fidelity prototype of speed marker editing after user testing 71
4.34 Mid-fidelity prototype of speed marker list after user testing 71

5.1 System architecture illustration 73
5.2 Hi-fidelity prototype of login screen 75
5.3 Hi-fidelity prototype of asset list 75
5.4 Hi-fidelity prototype of asset list filtering 75
5.5 Hi-fidelity prototype of video preview 75
5.6 Hi-fidelity prototype of video preview showing metadata 76
5.7 Hi-fidelity prototype of video preview showing markers 76
5.8 Hi-fidelity prototype of edit mode in general tab 76
5.9 Hi-fidelity prototype of edit mode adding a metadata field from the internal repository 76
5.10 Hi-fidelity prototype of edit mode adding a new metadata field manually 77
5.11 Hi-fidelity prototype of edit mode in markers tab 77

A.1 RTP archive card from the 1980's 84
A.2 RTP archive entry from Blue Order 85
A.3 RTP archive entry from Blue Order 86
A.4 RTP archive entry from Blue Order 87

List of Tables

2.1 Formats supported by mxfSPEEDRAIL S1000 11
2.2 Formats supported by mxfSPEEDRAIL F1000 13
2.3 Formats supported by mxfSPEEDRAIL O1000 14
2.4 Comparison between HTML5 and native mobile applications 23
2.5 Comparison between UCD techniques 29

3.1 Checking ENPS for raw footage information 41
3.2 Checking ENPS for news clip information 41
3.3 Repeated description work 42
3.4 Lack of time-based descriptions 42
3.5 Repetitive events are not taken advantage of 43
3.6 Unwatchable video previews 43
3.7 Untouchable thesaurus 44
3.8 Non-used indexation 44
3.9 Quick-and-dirty raw footage selection 45
3.10 Lack of communication between reporters and archivists 45

LIST OF TABLES

Abbreviations

3G	3rd generation mobile telecommunications
API	Application Programming Interface
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DOM	Document Object Model
EDL	Edit Decision List
EPG	Electronic Programming Guide
FIFO	First In, First Out
GOP	Group Of Pictures
GPS	Global Positioning System
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HD-SDI	High-Defenition Serial Digital Interface
HDD	Hard Disk Drive
HTML	Hypertext Markup Language
ISBN	International Standard Book Number
JSON	JavaScript Object Notation
MAM	Media Asset Manager
MOS	Media Object Server
MVC	Model-View-Controller
MXF	Material eXchange Format
NLE	Non Linear Editor
OS	Operating System
PDF	Portable Document Format
RTP	Rádio e Televisão de Portugal
SDI	Serial Digital Interface
SIM	Subscriber Identity Module
SOA	Service-Oriented Architecture
SSD	Solid-state Drive
SOAP	Simple Object Access Protocol
TV	Television
UCD	User-Centered Design
UI	User Interface
UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locator
URN	Uniform Resource Name
VCR	VideoCassette Recorder
VPN	Virtual Private Network
VTR	Video Tape Recorder
W3C	World Wide Web Consortium

Chapter 1

Introduction

From TV sets to tablets, from VCRs to Media Centers, informatics changed the way television is watched nowadays. But TV production, invisible to the main audience, was also revolutionized by a new paradigm that changed the tape-based systems into the new file-based systems. Among the many advantages of the digital world, one of the most important was the introduction of descriptive metadata into files. This data containing information about captured footage allowed a new level of organization that was impossible so far.

Nowadays, metadata is indispensable for tasks such as planning or archiving in the broadcast world. TV stations and production companies use metadata in order for their processes to achieve high levels of effectiveness. But sometimes efficiency can be compromised since most metadata needs to be inserted manually, representing an operational cost to these companies.

1.1 Context

This thesis was developed in partnership with MOG Technologies, leader company in MXF (Material eXchange Format) file-based solutions for the broadcast and post-production world. MOG's main product, mxfSPEEDRAIL, offers a modular solution to each phase of production, from video ingest to outgest. This product offers some metadata logging functionality, but very limited and mostly focused on transferring this data from one format to another without losses.

Through the creation of a partnership with Rádio e Televisão de Portugal, the Portuguese state-owned TV channel and major news producer for television, we were able to observe how metadata is used in their daily tasks. Focusing our work in the news production ecosystem, we mainly observed how reporters, journalists and archivists interacted and made use of metadata to improve their work.

This investigation served as a basis for the design of our solution, since through direct observation and inquiring of the users helped us understand the existant problems.

1.2 Motivation

The main motivation for this dissertation are the problems we are trying to solve and the use of new technologies that were not explored yet into solving a problem that is common to every audiovisual producing company.

Taking advantage of the mobile computing trend, we wanted to design a centralized yet mobile system, providing metadata logging without the typical computer desktop configuration.

With the use of touchscreen input methods we also wanted to accelerate the process of logging from the actual 8 hours of logging per hour of video to a speed near real-time.

1.3 Goals

- To investigate how descriptive metadata is used in a real work environment;
- To understand the problems of actual methods used by professionals and how they affect the organization;
- To design a solution based on the problems from last item;
- To begin implementing a high-fidelity prototype that can later be turned into a real application.

1.4 Document Outline

The present report is divided into six main chapters. In this first chapter we present the introduction to this dissertation. In the second chapter, the State of the art, we present some concepts and technologies that need to be understood in order to fully understand our project. The third chapter if fully dedicated to our case study of the RTP studios at Porto and here we report all the relevant findings from our investigation. The fourth chapter is dedicated into explaining the solution we had in mind and how it evolved into a final solution by involving users in the design process. The fifth chapter is where we report about the implementation of the hi-fidelity prototype and the problems we ran into because of the technologies we had chosen. The last chapter is dedicated to some reflection about how the final result was appealing given our initial goals, ending with some suggestions for future work to be further developed based on what was done so far.

Chapter 2

State of the art

To fully understand this project we first need to understand some of its underlying concepts and technologies, which will be explained during the course of this chapter.

2.1 Television production

Since its invention in the 1920's and massive adoption in the 1950's, television not only survived the new technological trends and advances, but also took advantage of the advent and improved their processes and target audiences. From VCRs to media centers and black and white monitors to 3D displays, even though there is much to be said about the technological advances in the user-end world, we will focus only on the content production processes, which also suffered massive changes in the last decades.

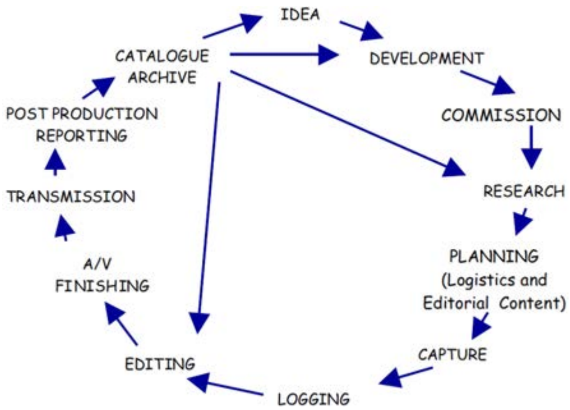


Figure 2.1: Stages in creating a TV programme [Dev02]

As shown by Bruce Devlin [Dev02] in Figure 2.1, the process of creating and broadcasting a TV programme is not as simple as one might think. The process usually starts with an idea that needs to be developed in order to be approved by a commission of directors. The development of this idea might be based on archived footage, therefore the connection from catalogue archive that we will explain in detail later. Assuming that the idea was approved, research is done in order to plan its production. Once again, the research may use old resources, let it be footage or other information such as details about its production (who directed it, who filmed it, what is the music from the soundtrack).

One of the most recent and important technical innovations was the evolution from tape-based to file-based workflows. It is from the next phase onwards that this change is most noticeable, so we will now present them separately so the whole process can be better understood.

2.1.1 Tape-based workflows

A tape-based workflow is a process in which a video tape is present throughout the whole process. Naturally, as illustrated in Figure 2.2, these workflows require the physical transportation of a videotape between every stage of the process.

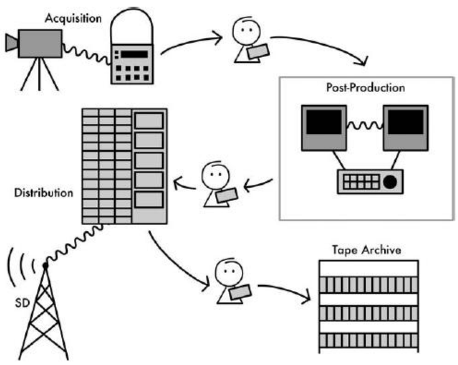


Figure 2.2: Workflow of a tape-based system [DWBT06]

2.1.1.1 Capture

The capture stage starts after all the previous stages of preparation are concluded and, as the name suggests, is where the recording of audio and video happens. This is typically executed using a video camera that records both components (audio and video) on a tape. This video tape may vary from a wide variety of formats which are typically not interoperable, since not only brands and sizes might differ but also the recording techniques may be incompatible (analog vs. digital). We can see some examples in Figure 2.3. These tapes are usually reusable but can only be reused a limited number of times, since the magnetic tape starts losing its qualities after some time.

Alternatively, the content can be recorded externally if it is being broadcasted live via, for example, a satellite feed. In this case the recording would happen at the TV station.

Figure 2.3: Assortment of video tapes [Wik12e]

2.1.1.2 Logging

One phase that is widely unknown for non-professionals lies between the capture and the post-production phases. After capturing audiovisual content, there is an organizational need of writing notes about the recorded footage. These notes depend from the organization who is requesting them and the use they want to give to them. Typical notes include production details such as title, time and date of recording, director, cameramen, staff, etc.

2.1.1.3 Post-production

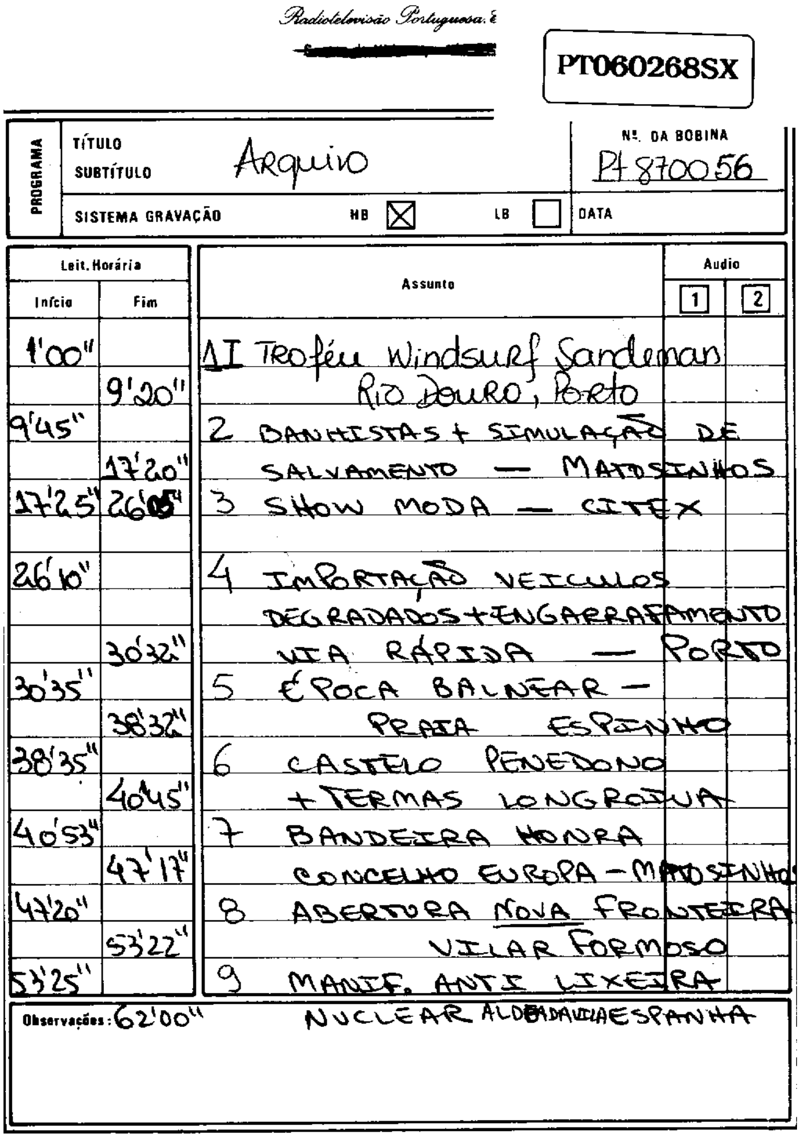
Post-production is the name typically given to the activities performed after the actual image shooting. Its main activity is known as video editing and this is where the initial images, also known as raw footage, are converted into watchable clips. In tape-based systems this is known as linear video editing where content must be accessed sequentially in order to be watched and transferred to another media. The first systems had an approach similar to film-making, where the tape was cut with a razorblade and joined to other pieces of tape to crate a clip. This was an arduous process and caused the original raw footage to be destroyed thus making it impossible to reuse. Later, the process evolved into the creation of a second tape called edit master, where specific contents from raw footage were copied to, preventing the destruction of any tape. [Wik12b]

2.1.1.4 Transmission

After the content has reached its final state, the transmission stage follows. This is when the contents are actually going to be broadcasted so they can be received in someone's TV set. This is usually done by using a playout system which controls a series of video tape players and starts playing them at specific times so the broadcast works as smoothly as possible. [Wik12c]

2.1.1.5 Archive

Last but not least, the final stage of archiving is one of the most important and complex phases in the process of television production. Its importance was already seen throughout the previous phases of the process. The development and research for new ideas is many times dependent of old footage, as well as the creation of new contents is frequently based on archived images. Furthermore, it is common knowledge that television stations repeat many of their contents, making the simple preservation of contents an important task. News production is one of the best examples about how archived images are used in the creation of new content. For example, if a musician dies, that news piece will naturally have images of his past performances. Even simpler things, like if the news was about something health-related, the background images would also have to be health-related, like images of hospitals, doctors and medical exams.



Radiotelevisão Portuguesa, E.

[redacted]

PT060268SX

PROGRAMA — TÍTULO / SUBTÍTULO: Arquivo — Nº DA BOBINA: Pt 8700 56

SISTEMA GRAVAÇÃO — HB ☒ — LB ☐ — DATA

Leit. Horária Início	Fim	Assunto	Audio 1	2
1'00"		1 I Troféu Windsurf Sandeman		
	9'20"	Rio Douro, Porto		
9'45"		2 BANHISTAS + SIMULAÇÃO DE		
	17'20"	SALVAMENTO — MATOSINHOS		
17'25"	26'05"	3 SHOW MODA — CITEX		
26'10"		4 IMPORTAÇÃO VEICULOS		
		DEGRADADOS + ENGARRAFAMENTO		
	30'32"	VIA RÁPIDA — PORTO		
30'35"		5 ÉPOCA BALNEAR —		
	38'32"	PRAIA ESPINHO		
38'35"		6 CASTELO PENEDONO		
	40'45"	+ TERMAS LONGROIVA		
40'53"		7 BANDEIRA HONRA		
	47'17"	CONCELHO EUROPA — MATOSINHOS		
47'20"		8 ABERTURA NOVA FRONTEIRA		
	53'22"	VILAR FORMOSO		
53'25"		9 MANIF. ANTI LIXEIRA		

Observações: 62'00" NUCLEAR ALDEADAVILA ESPANHA

Figure 2.4: RTP archive card from the 1980's

As we can see in Figure 2.4, this demands a huge level of organization from TV studios since the archive is constantly being accessed for contents that have to be findable by keywords such as theme, date, location, people, etc. Furthermore, this task needs to be done very quickly, not only for work efficiency reasons but also because in cases such as news production, the contents need to be produced the earliest possible since there is a competitive factor. A full version of Figure 2.4 can be found at Appendix A.1.

2.1.2 File-based workflows

Until now we were only getting to know processes which were invented many decades ago, but recently most audiovisual producers started working in file-based environments which changed the whole workflow in the process of creating television contents. As illustrated by Figure 2.5, we can see that this new paradigm no longer demands the physical transportation of a tape between every stage of the process, since digital media can be transmitted virtually as a bit stream rather than in a physical object. This was one of the most recent and important paradigm changes in television production, since it was how personal computers started entering TV studios where only dedicated standalone machines used to be operated.

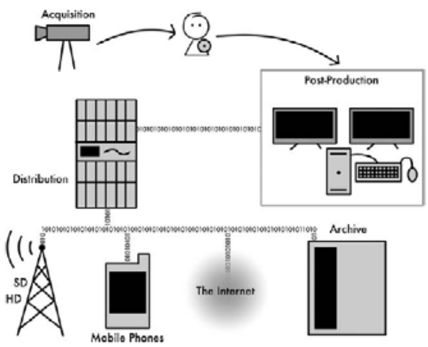


Figure 2.5: Workflow of a file-based system [DWBT06]

2.1.2.1 Capture

The capture stage was mostly untouched in the way it is practically performed, since the biggest difference lies in the media used to store the images. Tapes were mostly discarded and replaced by more reliable means such as optical discs and SSDs where the concept of a file can exist. Still, mostly due to cost implications, digital video tapes such as DVcam are still used nowadays, since their digital nature allows an easy integration with file-based systems with the advantage of not having to acquire new equipment such as cameras and storage media.

2.1.2.2 Ingest

The concept of video ingesting is a new concept that only came up with the file-based paradigm. Analogously to our body function, video ingesting is the phase where the videos are taken into the system for further work in the next phases. It is here where the videos are converted to files from whichever media they came from in whichever format we will need them to be. This allows the use of different capture systems without affecting the rest of the process, since the ingest process can normalize them all into just one format. We will explain more about this process in chapter 2.1.3.1, when we show the example of MOG's ingest system mxfSPEEDRAIL.

2.1.2.3 Logging

The logging stage was one of the most positively affected ones since file-based systems not only made it easier to execute the logging itself, but also speeded up the rest of the workflow through the use of metadata. Because of its importance, we reserve chapter 2.2.1 to explain further in detail what metadata is and how is it affects the whole process.

2.1.2.4 Post-production

In a file-based system, even though the concept of post-production is still the same as in a tape-based one, the process of converting raw footage into watchable clips changed significantly. While tape did not allow a quick random access to any part of the video, computers can access any part of a file in a question of milliseconds. This dramatically changed the way editing is done, since the process is now much quicker, allows many things that were not possible with tape-based systems and, in addition, they are generally much cheaper. This happens because these systems are usually just a standard computer and a piece of software instead of a dedicated machine built with only one purpose.

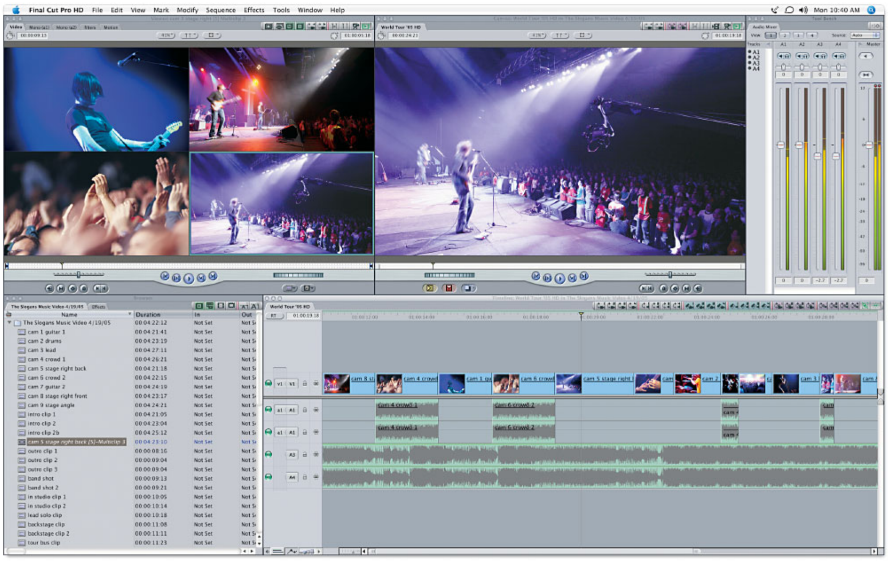


Figure 2.6: Screenshot of Apple's Final Cut Pro, one of the most popular NLEs

The simplicity of these Non-Linear video Editors (Figure 2.6), actually damaged the jobs of workers with video editing tasks, since people that wanted some footage to be edited no longer asked for someone else to do it. Instead, people such as journalists learned to use NLEs themselves and started editing their own news clips without the need of an extra person.

2.1.2.5 Transmission

Transmission, also known as broadcast, also kept the concept from the tape-based workflow, but the process also changed a lot. What used to be a collection of video tape players and a system that controlled its continuity over time was replaced by just one simple personal computer. This system typically has an operator who will build the channel's line-up by basically dragging the files with the video contents and dropping them in a specific software that will take care of starting the next program after the previous finishes and manage breaks and commercials making the process almost fully automatic.

2.1.2.6 Archiving

With this change of paradigm, the archive phase suffered a complete revolution. On the first hand, the concept of an archive as a physical collection of organized tapes completely lost its sense to something a lot simpler - a collection of files organized in such a way that we can find them by relevant criteria. Criteria such as file name and creation date are simple and would not require relevant additional work, but files allow for a much deeper way of description through the use of metadata (Chapter 2.2). Metadata allows the contents of the video to be described with whichever information we think that can be relevant for future searches.

This implied the creation of another job in the archive apart from the archivists who mainly operate the physical archive. This new job goes by the name of documentalist and has the task of inserting such information on the video files so as to make them searchable. Their work consists in watching the whole video and log every relevant detail in text form so it can be found with a quick and simple query. The best way to understand this job is to read Chapter 3.4.3 where we explain how the documentalist's work at RTP takes place.

2.1.3 Related technology

2.1.3.1 mxfSPEEDRAIL

MOG Technologies' [Tec12a] mxfSPEEDRAIL [Tec12b] is a centralized ingest/outgest solution that allows video rewrapping and transcoding from and to all the formats and codecs that are professionally used in the broadcast market. A seen in Figure 2.7 this allows the connection between systems that usually have different inputs/outputs and that would otherwise be incompatible.

Figure 2.7: Overview of mxfSPEEDRAIL capabilities [Tec12b]

The product is divided in 4 different models: mxfSPEEDRAIL S1000 [Tec12f], mxfSPEEDRAIL F1000 [Tec12c], mxfSPEEDRAIL O1000 [Tec12d] and mxfSPEEDRAIL P1000 [Tec12e], each one with a different functionality.

mxfSPEEDRAIL S1000 The mxfSPEEDRAIL S1000 (Figure 2.8) is a hardware/software system that allows ingestion of SDI and HD-SDI inputs to storages units such as Avid Unity MediaNetwork, Avid Unity ISIS, Omneon MediaGrid, removable e-SATA drives and any generic shared storages. Supporting up to 16 channels of embedded PCM stereo audio and all the main professional video codecs such as XDCAM, DVCAM, DNxHD, AVC-I, ProRes 422, among some others. A detailed table of supported formats can be examined in Table 2.1.

Some of the benefits of using mxfSPEEDRAIL S1000 include:

- Edit while capture, allowing video editing even before the file has finished recording;
- Multi-camera management with Gang Control;
- Scheduling engine for capture automation;
- Built-in VTR controller;
- Multi format/resolution;
- Multi destinations for complex workflows;
- Remote access and control by using a web GUI and a SOAP-based interface.

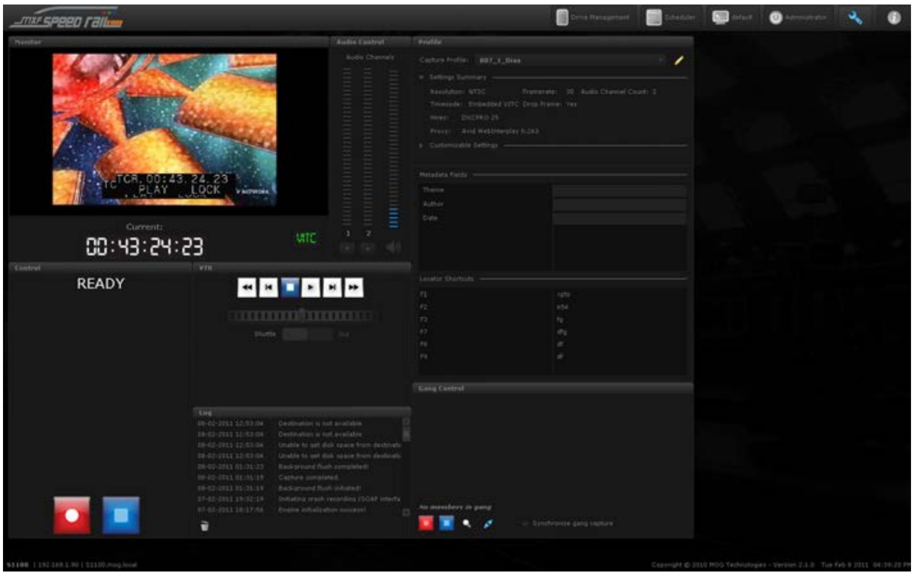


Figure 2.8: Screenshot of mxfSPEEDRAIL S1000

Table 2.1: Formats supported by mxfSPEEDRAIL S1000

INPUTS	• SDI (SMPTE 259M) • HD-SDI (SMPTE 259M / SMPTE 292M)
OUTPUTS	• 1 or 2 full resolution clips per channel • 1 or 2 proxy resolution clips per channel
PROXY ENCODING	• Avid MPEG2 (SIF - Source Input Format @ 2Mbps) • MPEG4 part 2 (SIF @ 2Mbps) • H.263 (Web Interplay) • H.264 (Web Interplay)

mxfSPEEDRAIL F1000 The mxfSPEEDRAIL F1000 (Figure 2.9) differs from the S1000 version because instead of receiving an SDI channel it receives its input from a file-based system, such as a file-storage media or networked device. The output formats are basically the same as the S1000 version but in addition, it also offers the possibility of recording proxy versions of the input video. Proxy videos are low-resolution versions of exactly the same video, with exactly the same characteristics apart from audio and video compression. This is very important in a file-based environment, since hi-resolution videos require a much bigger bandwidth and a much more powerful processing system and should only be used when needed. Considering there are some operations like previewing, web-publishing and even editing that do not require the full video resolution, using proxy versions allows money to be saved while also reducing the operational times where transfers have to be made. A detailed table of supported formats can be examined in Table 2.2.

The benefits of using mxfSPEEDRAIL F1000 are basically the same of using mxfSPEEDRAIL S1000 with files as input instead of SDI video. This has the great advantage of interacting with most systems nowadays, since they are also file-based. A great example of that is a MAM, a Media Asset Manager, which could need its files to be reencoded in order to be fed back to another system.

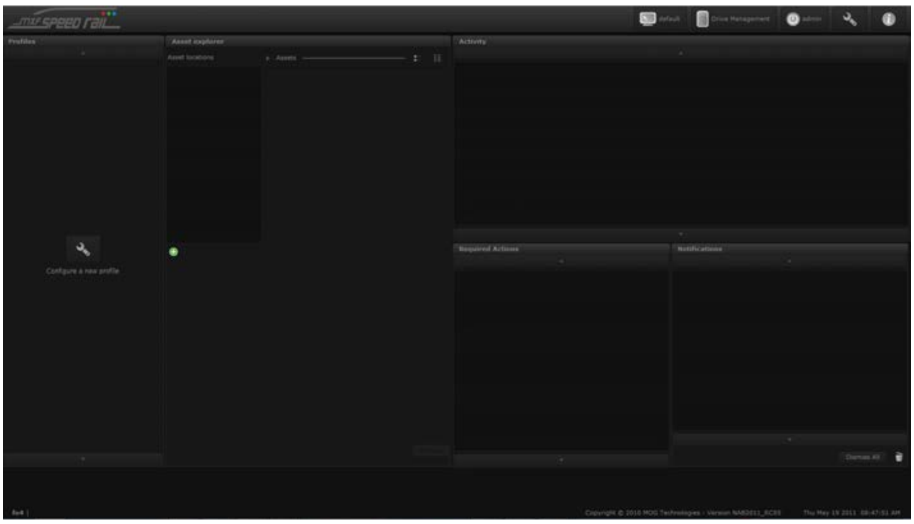


Figure 2.9: Screenshot of mxfSPEEDRAIL F1000

Table 2.2: Formats supported by mxfSPEEDRAIL F1000

INPUT FORMATS	• XDCAM - IMX D10, DVCAM • XDCAM HD - MPEG2 4:2:0, 4:2:2 • XDCAM EX • DVCPRO, DVCPRO50, DVCPRO HD • AVC-Intra
INPUT WRAPPERS	• MXF OP1a compliant • Quicktime (.mov - DV and MPEG)
PROXY ENCODING	• Avid OPAtom • MXF OP1a • H.263 web interplay

mxfSPEEDRAIL O1000 The mxfSPEEDRAIL O1000 is not an ingest system, but an outgest system. An outgest system is a system that instead of preparing videos for further editing (ingest), prepares videos to be either stored of published. In the case of mxfSPEEDRAIL O1000, it is able to export for systems such as:

- Avid Unity MediaNetwork
- Avid Unity ISIS
- Omneon MediaGrid
- Generic Shared Storage via CIFS
- Local Drive in Standalone Systems

A detailed table of supported formats can be examined in Table 2.3.

Table 2.3: Formats supported by mxfSPEEDRAIL O1000

INPUT CODECS	• IMX (30,40,50 Mbps) • DVCPRO, DVCPRO 50, DVCPRO HD • DNxHD (115, 120, 145, 175, 185, 220 Mbps) • AVC-I • MPEG LGOP
WRAPPERS	• MXF OP1a • MP4 (for web streaming)
PROXY ENCODING	• MPEG4 (H.264) for web streaming • Proxy H.263 (interplay web proxy)
AUDIO	• PCM • Dolby-E • Audio Mapping

mxfSPEEDRAIL P1000 The mxfSPEEDRAIL P1000 is a real time playback system. It basically allows an unified system to play videos of different formats in one single platform. Since its relevance for this project is small or none, we will not describe it into more detail.

2.2 Metadata

The term metadata was already used in this document as a component of some of the production stages that were described in the last chapter. [2.1.2.3, 2.1.2.6] In this chapter, we will explain what the concept means, what standards exist, how it is used for TV production and some related technologies that make use of this concept.

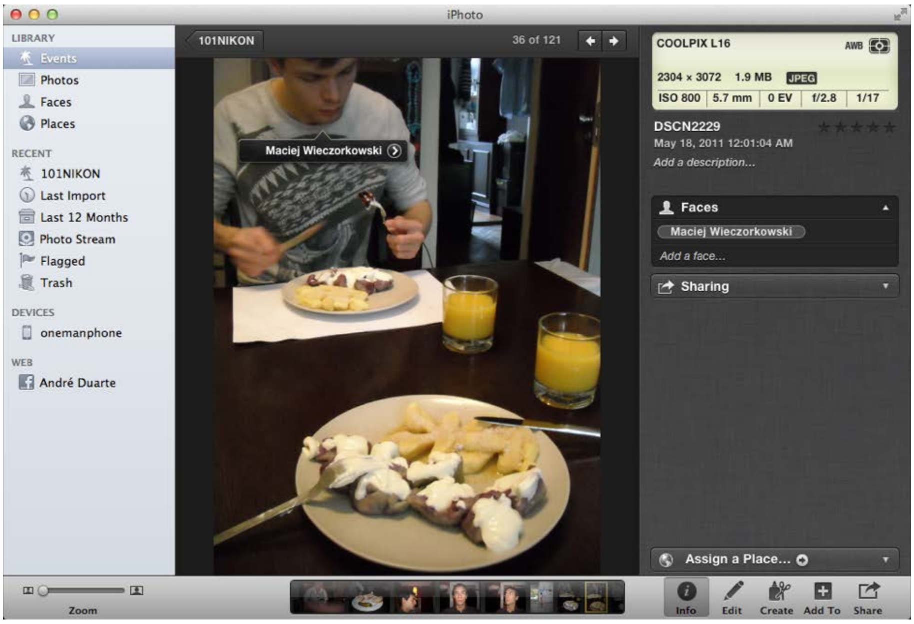


Figure 2.10: Apple's iPhoto screenshot full of metadata

Metadata, commonly known as "data about data" is a concept used to describe information that refers to another information. This is a concept easier to explain with examples, since it is constantly present in our daily uses of electronic equipment. A good example of metadata is available in any typical photo management software. When we take a photo with our digital camera and then see it in our computers, a series of data about the photo comes attached to the photo itself (Figure 2.10). This data is called metadata and it's usually divided in two categories:

Technical metadata Technical metadata is the name given to technical details about the content. As we can see in Figure 2.11, resolution, size, lens aperture, shutter, sensitivity, etc, are all technical details about the photo that are incorporated in its file, thus called technical metadata.

Figure 2.11: Apple's iPhoto technical metadata

Descriptive metadata Descriptive metadata differs from technical metadata because while technical metadata does not make any reference to the content itself, descriptive metadata has something to do with the content. A good example is also shown in Figure 2.10, where we can see a label on the person's face with his name. Since this data is describing the content of photo, then it is considered to be descriptive metadata.

2.2.1 Metadata uses for television production

So far we have presented metadata examples from photos since these are the ones most common to firstly understand the concept. But metadata is also widely used in other contents such as video. We will now examine how metadata, both technical and descriptive, enters the workflow of TV production.

2.2.1.1 Capture

Metadata enters the production workflow as soon as the camera starts recording. Typical cameras can record information about day and hour as well as technical information about the camera, recording modes, etc. Newer cameras also include a lot more features such as GPS location and the most advanced professional models may even have a SIM card with a 3G UMTS Internet connection for live connection with other logging systems.

Figure 2.12: Avid's Interplay Assist screenshot with notes on shots

This way, descriptive metadata can be included to mark, for example, good shots, bad shots, close ups (Figure 2.12), among other informations that might be useful in later stages of the process like knowing who was the cameraman or the reporter.

2.2.1.2 Post-production

The post-production phase, specially the editing stage, makes an extensive use of the metadata previously mentioned. This way they can accelerate the editing process since they do not need to analyse all the recorded footage when they are looking for specific things. Imagine a news report where a camera (or maybe more) is following a politician for hours while he campaigns on the street. From those hours of raw footage, there will probably be only a handful of shots that are worth editing for a final piece. Without metadata immediately showing where the interesting images are, the editor would be forced to what the whole video an take a lot of notes so he could find the most interesting moments.

2.2.1.3 Broadcast

Less important for this project, but still worth mentioning is the use of metadata also during broadcast. While most uses of metadata only concern in-house production-related operations, TV programmes are also transmitted with metadata so as to inform the audience what they are watching at the moment or will be able to watch in the near future. Such an example of that use is the common Electronic Programming Guide (EPG) that is present in most digital cable operators as well as digital terrestrial transmissions (Figure 2.13).

Figure 2.13: Screenshot of an EPG example

2.2.1.4 Archive

The archiving phase is the phase where metadata is used more deeply, specially concerning descriptive metadata. After the videos are manipulated, broadcasted and finally reach the archiving stage, there is an organizational need of making them searchable again for future use, such as explained in Chapter 2.1.2.6. This is a very important task, since without metadata, a thousand-hour digital video archive is reduced to a terabyte or greater jumble of bits; with metadata, those thousand hours can become a valuable information source. [WC02]

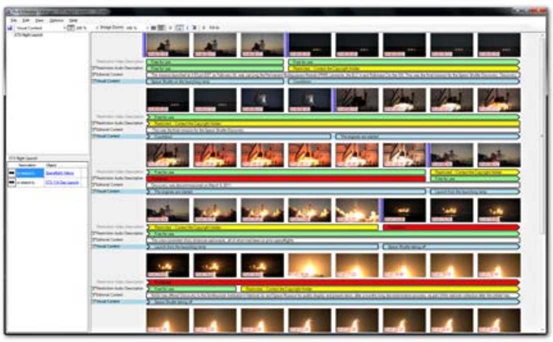


Figure 2.14: Screenshot of Avid's Interplay Cataloger

Present technology cannot yet search for image contents on a text basis, like searching for images of a politician based solely on his name or party. But attaching a text field to a video, or even a specific frame, makes this search an easy job for any computer system. Technical data about the audio and video streams, facts about date, location, authors, conditions for accessing the material - e.g. copyrights -, classification of parental rating and even links to other relevant material are only some of the typical contents of audiovisual metadata 2.14.

2.2.2 Metadata standards

Since metadata is considered essential by all audiovisual-based companies, it is important that this data should be stored in a standardized way so it that can be exchanged between different entities. Several standards exist in the market and we will now present two of the most important ones.

2.2.2.1 Dublin Core

One of the main initiatives for standardization is Dublin Core [Cor12], which provides a set of vocabulary terms which can be used to describe resources such as book, images, videos and even websites. The Simple Dublin Core Metadata Element Set (DCMES) [Cor04] offers a 15-element set of metadata fields to describe:

Title The name given to the resource, usually by the Creator or Publisher.

Creator The person or organization primarily responsible for creating the intellectual content of the resource. For example, authors in the case of written documents, artists, photographers, or illustrators in the case of visual resources.

Subject The topic of the resource. Typically, subject will be expressed as keywords or phrases that describe the subject or content of the resource. The use of controlled vocabularies and formal classification schemas is encouraged.

Description A textual description of the content of the resource, including abstracts in the case of document-like objects or content descriptions in the case of visual resources.

Publisher The entity responsible for making the resource available in its present form, such as a publishing house, a university department, or a corporate entity.

Contributor A person or organization not specified in a Creator element who has made significant intellectual contributions to the resource but whose contribution is secondary to any person or organization specified in a Creator element (for example, editor, transcriber, and illustrator).

Date A date associated with the creation or availability of the resource. Recommended best practice is defined in a profile of ISO 8601 [Con97] that includes (among others) dates of the forms YYYY and YYYY-MM-DD. In this scheme, the date 1994-11-05 corresponds to November 5, 1994.

Type The category of the resource, such as home page, novel, poem, working paper, technical report, essay, dictionary. For the sake of interoperability, Type should be selected from an enumerated list that is under development in the workshop series.

Format The data format and, optionally, dimensions (e.g., size, duration) of the resource. The format is used to identify the software and possibly hardware that might be needed to display or operate the resource. For the sake of interoperability, the format should be selected from an enumerated list that is currently under development in the workshop series.

Identifier A string or number used to uniquely identify the resource. Examples for networked resources include URLs and URNs (when implemented). Other globally-unique identifiers, such as ISBN or other formal names would also be candidates for this element.

Source Information about a second resource from which the present resource is derived. While it is generally recommended that elements contain information about the present resource only, this element may contain metadata for the second resource when it is considered important for discovery of the present resource.

Language The language of the intellectual content of the resource. Recommended best practice is defined in RFC 1766 [Alv95].

Relation An identifier of a second resource and its relationship to the present resource. This element is used to express linkages among related resources. For the sake of interoperability, relationships should be selected from an enumerated list that is currently under development in the workshop series.

Coverage The spatial and/or temporal characteristics of the intellectual content of the resource. Spatial coverage refers to a physical region (e.g., celestial sector) using place names or coordinates (e.g., longitude and latitude). Temporal coverage refers to what the resource is about rather than when it was created or made available (the latter belonging in the Date element). Temporal coverage is typically specified using named time periods (e.g., Neolithic) or the same date/time format [Con97] as recommended for the Date element.

Rights A rights management statement, an identifier that links to a rights management statement, or an identifier that links to a service providing information about rights management for the resource.

All the Dublin Core elements are optional and may be repeated, such as having two Creator tags or several Language tags according to necessity. This provides a helpful basis for metadata logging, but since it was initially thought for document-like objects, it is still incomplete to be applied in moving pictures. Fortunately Dublin Core is extensible and can be further adapted to purpose.

2.2.2.2 MPEG7

MPEG7 [Mar04] is a standard developed by Moving Picture Experts Group that aims to describe audiovisual content with some degree of interpretation of the information. Taking many concepts from Dublin Core and other existent standards, MPEG7 offers more flexible and extensible framework to describe audiovisual metadata. This framework takes account that many descriptive concepts are meaningful in the context of a specific application, so instead of being limited to a particular use or so extensive as to cover all possible needs, MPEG7 tries to be as generic as possible, using XML schemas as extensions to allow adaption to a particular context. Therefore, it includes a set of description tools to allow users to create descriptions of content that can include:

- Information describing the creation and production processes of the content (director, producer, title);
- Information about the usage of the content (copyrights, usage history);
- Information about the storage features of the content (format, encoding);
- Structural information about spatial, temporal and spatial-temporal components of the content (scene cuts, segmentation in regions, encoding);
- Information about low-level features in the content (colors, textures, timbres);
- Conceptual information about the reality captured by the content (objects, events, interactions);
- Information about how to browse content in an efficient way (summaries, variations);
- Information about collections of objects;
- Information about the interaction of the user with the content (user history).

2.2.3 Related technology

There are many technological solutions to deal with the complex details of metadata introduced so far. Since most features are repeated in high-end solutions, in this section we will only present one of the most complete ones.

2.2.3.1 Avid Interplay

Avid Interplay [Avi12a] is a set of software and hardware for the automation of all television processes, from acquisition to distribution (Figure 2.15), that is used by many major TV stations and content production companies.

Avid Interplay allows the insertion of metadata on videos for future metadata-based searches. These features are both offered by its Media Asset Manager component. Interplay Media Asset Manager allows users to insert data to videos on a time-referenced basis. It allows many basic

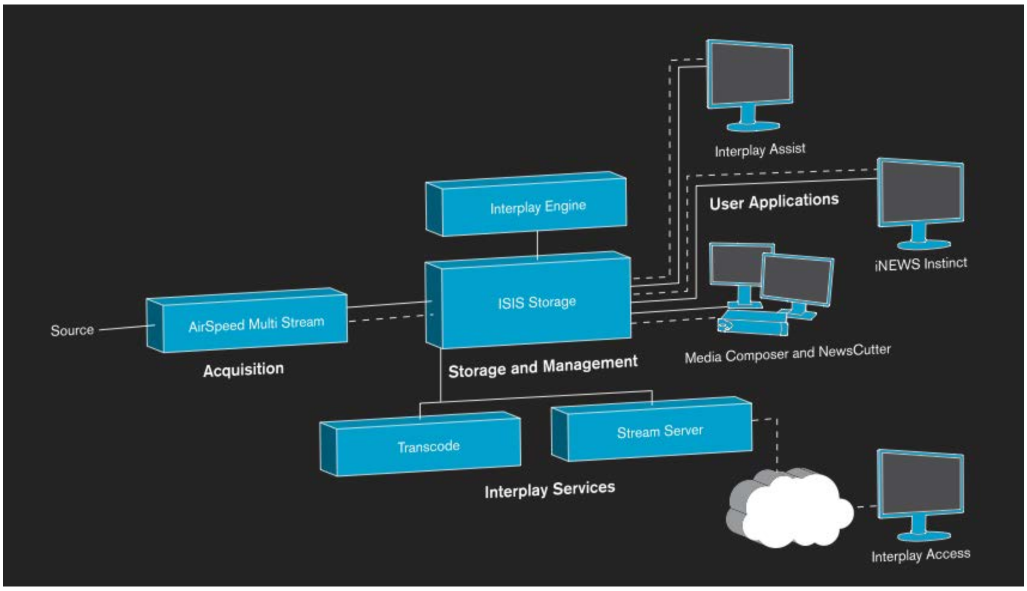


Figure 2.15: Avid Interplay Workflow

different data types to be inserted, like strings, ints, dates, and other base types of any database system.

Although it is clear from the commercial brochures [Avi12b] that metadata is more extensively used throughout Interplay's workflow, this kind of data is not available to the general public.

2.3 Mobile computing

Mobile computing is a human-computer interaction concept that concerns devices with software and hardware that are designed to be transported while in use, unlike the typical desktop environments which are built for a static use - typically composed of a screen, keyboard and mouse on top of a table. But with the recent boom of smartphones and PDAs and the following price reduction, more and more attention is being given to the mobile computing as a serious complement (if not alternative) to traditional desktop systems.

Nowadays there are two main mobile based systems: smartphones and tablets. While smartphones are the evolution of the mobile phone with added capabilities like greater processing power, wi-fi connection and touch screens, tablets are a mix between a smartphone and a laptop computer. Tablets tend to have the same functionality as smartphones, but since smartphones have reduced screen sizes, thus reducing its capabilities for some tasks, tablet computers have wider screens and allow a more diverse set of tasks to be executed.

There are two main operating systems common to these technologies: Android [Goo12] and iOS [App12]. Both have their advantages and disadvantages which are mostly subjective since there has been discussion about them since the beginning of both OSs. More alternatives also

Figure 2.16: Examples of smartphones

Figure 2.17: Examples of tablets

Top Six Smartphone Operating Systems, Shipments, and Market Share, 2012 Q1 (Units in Millions)

Mobile Operating System	1Q12 Unit Shipments	1Q12 Market Share	1Q11 Unit Shipments	1Q11 Market Share	Year-over-Year Change
Android	89.9	59.0%	36.7	36.1%	145.0%
iOS	35.1	23.0%	18.6	18.3%	88.7%
Symbian	10.4	6.8%	26.4	26.0%	-60.6%
BlackBerry OS	9.7	6.4%	13.8	13.6%	-29.7%
Linux	3.5	2.3%	3.2	3.1%	9.4%
Windows Phone 7/Windows Mobile	3.3	2.2%	2.6	2.6%	26.9%
Other	0.4	0.3%	0.3	0.3%	33.3%
Total	152.3	100.0%	101.6	100.0%	49.9%

Figure 2.18: Smartphone OS market share [New12]

exist but while these two are direct competitors with considerable market shares, others tend to have very small rates when compared with the two major ones (Figure 2.18).

2.3.1 Web-based applications

A typical problem when developing applications is compatibility amongst different operating systems. Each operating system has its own APIs and frameworks, resulting in limited access to the chosen one. Therefore, in case we want to develop a multi system application, repetitive work will be needed to adapt each version to each operating system. In mobile application development, that problem grows even bigger, as the two main operating systems - Android and iOS - both have its advantages and disadvantages to such a level that the resulting market is widely spread between both systems.

Fortunately, both systems can be unified by the so well-known web standards, where HTML5 clearly stands out as more than a standard to create web pages, but a real alternative for the development of system-independent applications, that despite running on a browser, offer almost as much functionality as native programs.

Table 2.4: Comparison between HTML5 and native mobile applications

HTML5 application	**Native application**
Slower performance	Better performance
Partial access to API	Full access to API
System independent	Operating System dependent

A first look at Table 2.4 may reveal that a native application would be a better choice, but even though the application's requirements will only be defined later, an evaluation of each field of comparison, when based on the predicted requirements for the core tasks for the application, shows that the native advantages are not that relevant.

Performance, although important, is not a crucial factor for the kind of task we want to preform. Metadata logging and querying demands very little processing effort on the client-side, specially when video playback (the most demanding task) can be GPU-accelerated on most devices nowadays. API access, although limited, is also not essential for this application, as the chosen framework (discussed in the next chapter) offers enough functionality for our requirements.

System independence, however, was the decisive factor for choosing HTML5 as the basis of our application. As explained before, the diversity amongst existent operating systems would demand that we either lose market by limiting the application use to a specific operating system, or lose time by redeveloping more that one version of the system. Considering that this work is being developed with a company, we understand that these two disadvantages are crucial for a business and therefore we chose HTML5 as the technology to be used in this project.

2.3.2 Sencha Touch

Sencha Touch [Sen] is a framework - the first of the kind - that tries to mimic native iOS/Android applications by making full use of web standards like HTML5, CSS3 and JavaScript.

Figure 2.19: Checkout, a Sencha Touch application

Being based mainly on Webkit [Web], Sencha Touch has full support for Webkit-based browsers like Safari and Google Chrome, making it fully compatible with iOS devices, Android devices and PCs. Other browsers do not guarantee full operability, therefore are not recommended to be used.

By using this JavaScript framework we can develop a web application a lot faster as we can use a lot of widgets that are typical amongst mobile applications, saving time by not having to develop them ourselves. Furthermore, its authors even say one doesn't really need to know all about HTML5 to use Sencha Touch, although it's a good idea to get the basics down [GM11].

First of all, Sencha Touch offers us a GUI environment instead of the typical webpage look, dealing with tricks to make the application look full-screen so we don't have to care about the low-level development of the program. Then we can add to the screen GUI controls - like buttons, toolbars, menus and forms - and transitions - made with CSS and AJAX so we never leave the initial "page".

As the framework name reveals, touch event management is already included in this framework, so we can easily identify events like Tap, Double-Tap, Swipe, Drag and even Pinch for multi-touch surfaces, making it easier to assign actions to them.

One may have noticed that the comparison on Table 2.4 did not include anything about offline compatibility. That's because Sencha Touch allows applications to work offline, by making use of the so called Cache Manifest included in HTML5 specification. This is really important for us, as even though an Internet connection is always needed - because video contents are streamed online -, poor network coverage could cause Internet to fail and our work to be lost. This way we can prevent that from happening as we can turn the device off and back on without losing application data [Wik12d].

Last but not least, this framework even allows the possibility of fully logging the activity of an application by recording every step a user executes during the use of the application. Every touch, swipe, pinch, etc is logged and can be reproduced just like a video recording of the execution. This is a very important feature for a User Centered Design as the one we are approaching - we will explain more about UCD later in this document - since we can analyze the way our users interact with the interface, correct eventual mistakes and easily learn what the best practices are.

2.4 Interaction design

Nowadays, developing computer software doesn't only need to have in mind effectiveness at executing required tasks, nor technical issues like performance are the only 'extras' to consider when designing an application. With the massification of computers, a new wider range of people that did not have computer-related backgrounds started using computer software to support their activities, both in professional and personal contexts. This change demanded software developers to start worrying about usability, term that will be explored further into this document.

Preece et al. [PRS02] defined interaction design as "designing interactive products to support people in their everyday and working lives".

To better understand this concept, one should first understand the difference between Interaction Design and Software Engineering, as both lead to the same result as the definition cited above.

While Software Engineering aims on the production of software solutions for given applications, Interaction Design aims on finding solutions to support people, through the design of interfaces that allow humans to communicate and interact with them. To make this distinction even more clear, there's a simple analogy between professions when it comes to building a house. While the architect is mainly concerned with how people interact with the house - considering how spaces relate to people and how people relate to them -, the civil engineer is focused on the technical details of the project - like cost, durability, construction methods, etc. Likewise, Interaction Design is related to Software Engineering in the same way Architecture is related to Civil Engineering [Win97].

Interaction design follows a simple process based on four main cyclical activities which may overlap. Those activities are:

Identifying needs and establishing requirements
This activity concerns the elicitation of requirements for the product we want to develop. In order to make a product that is indeed interactive, the user's input plays a very important part in this activity, fundamental to a user-centered approach, that will be discussed more in detail later in this document.

Developing alternative designs
Divided in two sub-activities - conceptual design and physical design -, this is the phase where, after establishing the requirements, we design both a model that describes how the product should work and a model that shows how the product should look. A very important aspect is that this is not an activity that should be developed just once, but as many times as one can. Just as with brainstorming, quantity generates quality, therefore alternative designs are encouraged.

Building interactive versions of the designs
If we want users to try our designs, it is naturally mandatory for us to build interactive versions of them so they can be tested. But interactive versions don't imply software versions. Although software prototypes give the user the real perspective of what the application will look like, the time and effort needed to implement them make it impossible to develop a series of them. Therefore, very low-fidelity prototyping is encouraged, like a series of paper prototypes allied with some role-playing to simulate interaction.

Evaluating designs
After having our prototypes ready to test, we need to evaluate each and every one of them to see if they match the users' requirements, if they appeal to them, and which are the best characteristics of each. This feedback will be used to restart the process from the second activity, adding the results

of the evaluation to the initial requirements in order to develop a better version of the software. Iteration is one of the main characteristics of interaction design and this process should only stop when the evaluation is widely successful.

2.4.1 Usability

The main focus of interaction design is usability. Usability is the criteria that ensures interactive products to be easy to learn, effective to use and enjoyable from the user's perspective. Some of its main goals are:

- Effectiveness;
- Efficiency;
- Safety;
- Utility;
- Learnability;
- Memorability.

Effectiveness concerns on how good a system is at performing the tasks it is suppose to perform. Efficiency refers to how helpful the system can be at assisting people in their tasks - not to be confused with technical efficiency (e.g., how the system gets the most out of the machine's resources). Safety concerns on how the system is able to avoid and recover from undesirable situations, like deleting a document instead of saving it, or even the common undo button that help us recover from sometimes fatal mistakes. Utility measures how much the system is useful to the user in terms of its functionalities - not only functions can be missing, but useless functions are of no interest to the user. Learnability refers to how easy a system is to learn to use and Memorability relates to the ability of users to memorize and remember how to use the software - referring to not only the software itself, but also its documentation.

To achieve these goals one should follow some usability principles, also known as heuristics, that provide the guidance needed that is needed to have in mind from the conception of the system until its evaluation phase.

2.4.1.1 Nielsen 10 principles

The ten main usability principles, developed by Nielsen [Nie94] are:

1. Visibility of system status - provide appropriate feedback in order to always keep users informed about what is going on;

2. Match between system and the real world - speak the users' language by using concepts familiar to the user instead of system-oriented terms;

3. User control and freedom - always allow “emergency exits” for users to easily escape from places they unexpectedly ran into;

4. Consistency and standards - never use two different names to describe the same action or situation;

5. Help users recognize, diagnose, and recover from errors - describe the nature of the problem and suggest a way of solving it;

6. Error prevention - always try to prevent errors from happening;

7. Recognition rather than recall - make objects, actions and options visible;

8. Flexibility and efficiency of use - provide shortcuts invisible to novice users that allow experienced users to carry out tasks more quickly;

9. Aesthetic and minimalist design - avoid showing irrelevant and unneeded information;

10. Help and documentation - provide easily searchable information to provide help to unexperienced users

2.4.2 User Centered Design

Interaction design is all about users and how they interact with a product. Therefore it is only natural to involve them in the development process from an early start, and User Centered Design is the best approach for that. There are various techniques to involve users in this development, like Ethnography, Coherence, Contextual Design and Participatory Design. In order to select the one that better fits our project, we need to analyze each one of them and compare their pros and cons.

2.4.2.1 Ethnography

Ethnography is a method used in the social sciences to describe the social organization of activities so as to understand work. It literally means “writing the culture” and its activity is based on observing the users’ environment and participating in their day-to-day work [HA83]. The purpose of this is to understand some implicit activities that are so deep inside the users’ everyday lives that cannot be captured by other means since they do not recognize them. Because of its very nature, it is difficult to define what should be documented in this activity, thus making it much more an experience than a data-gathering exercise. Still, documentation is needed in order to share this knowledge with other team members. This documentation can have multiple forms, from notes of your own to pictures or recordings of people’s work. Its purpose is much more detail rather than rationalization, which is a later activity based on the data collected by ethnographers.

2.4.2.2 Coherence

Coherence [VS99] is a combination between ethnography and requirements engineering. It aims to complement the social analysis of ethnography with object-oriented approaches from software engineering. This method provides a set of questions to give some guidance to the observer, named "viewpoints", and a set of goals to help the requirements activity, named "concerns".

2.4.2.3 Contextual Design

Contextual Design is a technique to handle and interpret the data from fieldwork - such as ethnography - in order to produce a software-based product. It is composed of seven parts: Contextual Inquiry, Work Modeling, Consolidation, Work Redesign, User Environment Design, Mockup and Test with Costumers, and Putting It into Practice. Contextual inquiry is an approach to ethnographic study that follows an apprenticeship model where the designer works as an apprentice, with the user trying to teach him about his work. Work Modeling aims to produce a model of the work by making work flow, sequence, artifact, cultural and physical models. Consolidation helps understanding a more general picture of the work from different points of view one may have from different users. The other parts are not relevant to this project's context.

2.4.2.4 Participatory Design

The intention of Participatory Design is to involve users to such an extent that they become equal partners in the design team, designing the product in cooperation with the designers. The main difference of this technique is that it consists primarily in low-fidelity prototyping, mostly paper-based, to allow a deep involvement of the users in all activities, since they don't have the knowledge needed to build a software mockup.

By comparing the pros and cons of the four different techniques on Table 2.5 we are planing to adopt both Ethnography and Coherence, since user involvement is desired but not available to the extent the last two techniques demand.

2.4.2.5 The Star Lifecycle Model

The Star lifecycle model is an alternative to the waterfall lifecycle to support the design of interactive interfaces. This model does not specify the order the activities are suppose to be followed, since we can naturally start with the requirement elicitation or evaluate them from a previous project. As we can see on Figure 2.20, the activities are highly interconnected and to go from any activity to another, evaluation is required. Using this model in a UCD context will highly involve the users in the development process thus guaranteeing their approval during all the project.

Table 2.5: Comparison between UCD techniques

	Ethnography	**Coherence**	**Contextual Design**	**Participatory Design**
Active user involvement	Low	Low	Medium/low	Very high (equal partners)
Role of designer	Reveal findings about work	Present ethnographic data according to the "viewpoints" and "concerns"	Steer discussion and interpret findings	Equal partners
Length of study	Continuous and extensive	N/A	A series of 2-hour interviews	A series of 2-hour design sessions
Benefits	Good understanding of work	Overcomes some disadvantages of ethnographic data	Designed to feed into the design process	Users' sense of ownership is increased and their contact is beneficial for designers
Drawbacks	Difficulties translating findings into design, requires expertise and time	Coverage limited to presenting ethnographic data	Involves many diagrams and notations, making it complicated for users to understand the output	Too much involvement of users can be counter-productive
When to use	Where there is sufficient time and expertise	If an ethnographic study is to be conducted	When a user-centered focus is required	Whenever users are available and willing to become actively involved

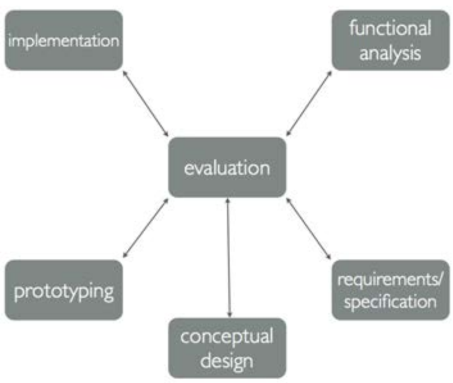


Figure 2.20: The Star lifecycle model

Chapter 3

Case study of RTP Porto

3.1 Introduction

For this project, we successfully approached Rádio e Televisão de Portugal (RTP) for a partnership in which we would be allowed to observe and investigate their work in order to devise a solution that would correct eventual problems that might exist or just improve their efficiency. Having accepted our invitation, we decided to focus on the archiving department, where metadata was mostly used. Since RTP has two main headquarters, one at Lisbon and one at Porto and it would be impossible for us to move to Lisbon for this research, we decided to focus only on the work done at RTP Porto.

At RTP Porto there are two main work divisions: general programming, where some (few) TV shows are produced, and information, where the afternoon news show is produced as well as most of programming for RTP Informação (the old RTPN) channel, RTP's cable news channel. But since most of the work is information-related we will explain it further in detail.

3.2 News production at RTP

News production is not just another activity of RTP. It is the main activity developed by the public TV station and its excellence has given them many awards as well as the leadership in terms of audience. In order for this to happen, RTP's news department is the department that never sleeps, with constant news gathering, production and reporting for its wide public. Consequently, other departments from which the news department depends of, such as archive and broadcasting, also need to have a 24/7 work mode in order to guarantee that the workflow never stops.

3.2.1 News shows

As previously mentioned, RTP Porto is in charge of the production of the afternoon news show. This show is broadcasted live from a virtual studio (Figure 3.1), a room all painted green in order to use chroma key backgrounds.

Figure 3.1: RTP Porto virtual studio

In terms of information systems, journalists build the program line up using Associated Press' ENPS [Pre] software, which allows centralized content creation and distribution as well as many features that help to plan the production of such contents.

3.2.2 News pieces

News pieces are the main content of news shows. These videos are usually 5 to 10 minutes long and report some news-related happening. Their composition is mainly edited raw footage, which was previously captured by a reporter, and archived videos that were previously captured or broadcasted. Journalists add graphics, text and usually a voice soundtrack explaining the contents.

3.2.3 News-related web clips

After edition, all of the clips go to both some TV program and also to RTP's news website [eTdP]. This innovative service allows people to build their own TV shows or just watch every news video content as soon as it was produced and uploaded to their website.

3.2.4 News-related shows

Even though RTP Informação is a news channel, not all off their contents are news shows. Some of them can be news-related shows like political commentary, sports commentary and cultural events reporting. These shows are also filmed in the same location as the news show (the virtual studio), but using different backgrounds in order to provide a completely different scenario.

3.3 Main stakeholders

There are three main stakeholders in the process of news production at RTP Porto: Reporters, Journalists and Archivists (or Documentalists). We will now explain each one of them.

3.3.1 Reporters

Even though the typical description of a reporter is the name given to people who researches, writes, and reports on information to be presented in mass media [Wik], at RTP a reporter is a journalist who goes on the field in order to report something.

Figure 3.2: Cameraman and Reporter from RTP ready to start recording

Reporters have a daily schedule of happenings that they must visit and report. In case of RTP Porto, since their news gathering is all video-based, they make these trips accompanied by a camera operator that captures the images while they perform their journalistic duties such as interviewing and note taking (Figure 3.2). After recording the images, they must immediately go back to the studios in order to deliver the storage media and allow the journalists to continue the work.

3.3.2 Journalists

Journalists are the people who develop an idea into something suitable to the public. This means that they get the raw footage filmed by reporters and turn it into a news clip that explains the news by adding a voice soundtrack and context information like an old related happening as well as pictures from archive to enrich the information of the clip.

What might be a surprise for people who know the usual process of creating a TV content is that the editing part is not done by a professional video editor. With the massification of video-editing technology and the ability to edit a video on a regular personal computer, this activity is currently done solely by the journalists themselves. At RTP Porto they use Quantel's sQ Cut [Qua] to edit those images since this software works remotely and is operated online by a computer connected to a an Ethernet network. This approach has several benefits, like not needing many powerful computers to operate the software and the typical advantages of content centralization, making every content automatically available to every workstation.

3.3.3 Archivists/documentalists

According to the Society Of American Archivists, the primary task of the archivist is to establish and maintain control, both physical and intellectual, over records of enduring value. [Arc] But much like the previous definitions, at RTP this profession does not follow the common designation.

There are two job designations for what is commonly known as an archivist. At RTP Porto, an archivist is the person that takes care of the physical part of the archive. This includes tape management such as identifying tapes with adequate descriptions, storing them on the archive, erase them when space is needed, etc.

On the other hand, there is the job designation of documentalist. Garfield [Gar53] defines documentalism by comparing it to librarianship. “While the wide subject of librarianship is concerned with every aspect of the treatment of books, the business of the documentalist is to make available the original information that has been recorded in articles in periodicals, pamphlets, reports, patent specifications and such records. Because the material, with which it deals is so much the more voluminous, its methods must be so much the more precise. So, by his unostentatious drudgery, the documentalist contributes to the increased production of genius, and becomes, by proxy, the benefactor of mankind.” This somewhat romanticized version of the librarian seems to be shared by archivists.

In RTP’s case, documentalists differentiate themselves from archivists since they are the ones that make the information searchable by adequately describing its contents. This activity is performed by using a Media Asset Manager that allows metadata to be inserted in video contents making the videos searchable in text form. Apart from just searchable text, there is also the option of thesaurus indexation. Thesaurus is a dictionary of synonyms which allows a more structured and organized way of archiving, since posterior searches can simply be selected from a pre-defined list of concepts. This has two main advantages: non-ambiguity between similar names - should one search for coffee house, coffee shop or caffè? - and database efficiency since the space taken is smaller and its indexed nature allows for quicker query answers.

At RTP there are 4 levels of documentation:

Level 1 Basic information: Title, Date, etc. Synthetic resume: a short text description of what the content is about.

Level 2 Analytic resume: a full text description about every content that could be relevant in a search.

Level 3 Basic indexation: thesaurus indexation based on the synthetic resume.

Level 4 Exhaustive indexation: full thesaurus indexation. In this level, the archive process lasts eight hours for each hour of video footage.

Each level accumulates the needs of the previous ones.

It should be clear that while the synthetic resume is a description of what the content is about, the analytic resume refers to the raw insides of the video. This means that the contents of the analytic resume could have nothing to do with the ones from synthetic resume.

Resumo Sintético	Almada, Hospital Garcia de Orta, em Almada, foi obrigado a pedir compressas emprestadas para não adiar cirurgias; fornecedor do hospital suspendeu a entrega do material por falta de pagamento, a dívida ao fabricante é de 260 mil euros.

Figure 3.3: Example of a synthetic resume

Resumo Analítico	Bloco operatório com cirurgia a decorrer, hospital Garcia da Orta; declarações de Hélder Miranda, Diretor Comercial da Albino Dias de Andrade.

Figure 3.4: Example of an analytic resume

A good example is shown in Figures 3.3 and 3.4, a real example taken from RTP's archive. Here we see that while the synthetic resume mentions what the content is about and the news itself, the analytic resume mentions raw details about the clip, like the hospital and the name of the person that appears on the pictures. The happening is irrelevant to the description.

Indexação

Lista de termos

Idx	Termo	Infraconceito
1	BLOCO OPERATÓRIO	
2	CIRURGIA	
3	HOSPITAL GARCIA DA ORTA	EXTERIOR
4	HOSPITAL GARCIA DA ORTA	INTERIOR
5	ALMADA	
6	PORTUGAL	

Figure 3.5: Example of indexing entry

Examples from indexation can be found in Figure 3.5. This is clearly a level 4 description, since it includes indexation from details only described in the analytic resume, such as the name of the hospital, details from every different scenario, etc.

The full archive entry where Figures 3.3, 3.4 and 3.5 were extracted from can be found in Appendix A.2.

3.4 Typical workflows

In order to better understand how all these entities relate to each other, we will present some workflow diagrams illustrating the main activities performed at RTP studios.

3.4.1 General workflow

Firstly we present, at Figure 3.6, a workflow that illustrates only the main activities of RTP, involving our three main entities: reporters, journalists and archivists.

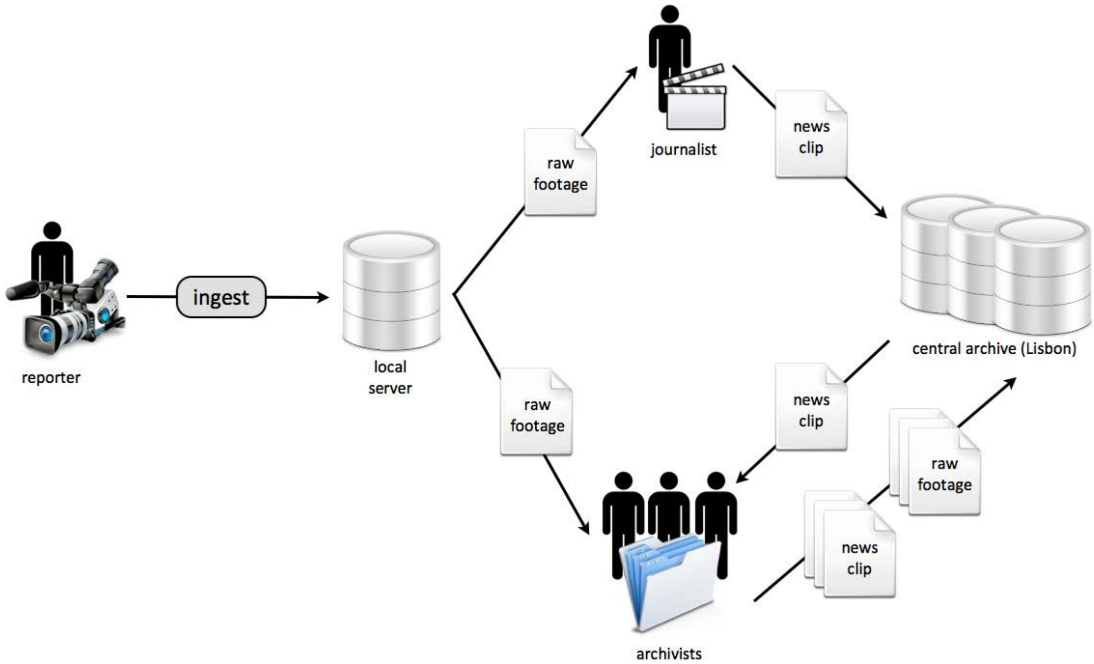


Figure 3.6: General workflow between reporters, journalists and archivists

The workflow begins with the reporter (along with a camera operator) capturing images from some happening. As soon as it is over, the physical media (memory card, tape, DVD, etc.) is immediately delivered to RTP studios for ingesting. The ingest department places the raw footage at a local server, only accessible from inside the building's network.

From now on, the workflow divides in two activities performed in parallel. As soon as the footage starts being ingested, the journalist can automatically start accessing the images even if they aren't fully ingested yet (thus only being able to access the already ingested parts). At this moment, the journalist starts editing the images into a news clip and may ask the archive for images to help enrich the news clip, an activity that will be detailed further in this document at section 3.4.2. As soon as the clip is edited, it is submitted to a central archive server in Lisbon and added to ENPS news show line-up or some other end that the clip may have.

At the same time, as these journalists tasks take place, the archive is also working with the exact same assets and performs three main tasks. At the archive, two of the tasks have one day of delay from real time. This happens because videos keep being delivered during the day at the archive, without predictable quantity or length, thus making it impossible to plan a workday in advance. Delaying the archiving activity by one day doesn't negatively affect the company and allows for a better work planning. Assets archived by journalists are divided in two types: raw footage and news clips/programmes. While journalists get the raw footage and transform it into a clip by trimming most of its contents, the archive department gets the full raw footage and archives what is most relevant, submitting it to the central archive. This activity will be explained more in detail at section 3.4.3. As for news clips, after being edited and submitted to the central archive by journalists, they are received by the archive department to be fully archived with adequate description. This activity will also be explained in section 3.4.4.

3.4.2 Request for archived footage

The third activity performed by the archive has a deep connection with the other two main archiving activities.

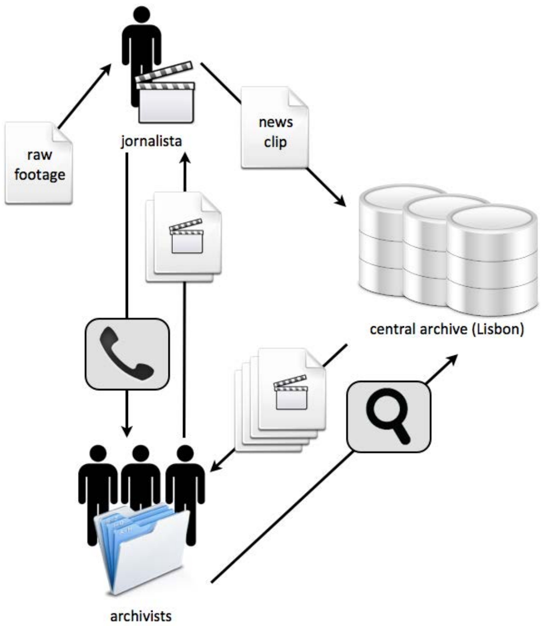


Figure 3.7: Workflow of archived footage request

As mentioned in section 3.4.1, whenever journalists need images from the archive to enrich a news clip, it is the archivists duty to search for those images and send them back to the journalist workstation. For that reason, there is always one archivist detached from the archiving activities to attend requests for archived footage. This archivist stands by the phone waiting for an internal call by some journalist who makes the request. In the case that the designated archivist cannot answer the call at the moment, one of the other archivists will pro-actively answer the telephone. The telephone request is an informal conversation that includes all the requirements from the journalist and can include some advice or questions from the archivist. The only mandatory requirement is that the journalist gives the archivist his employee code so the images can be properly identified and can later be found by the journalist. All these details are written on a paper form and added to one of three FIFO queues with different urgency criteria. There are typically three kinds of urgency:

Urgent Typically, footage requests for news pieces that are about to go on air and need to be processed as soon as they can. Such a request will have maximum priority and in case there is no archivist free to do the job, it is informally decided which one will stop his actual task and work on that request.

Non-urgent This is the typical request by default, usually for a news clip that will air later on. It needs to be concluded quickly but without abnormal urgency. Such a request is added to the queue and processed when its turn is up.

Research This is the less urgent kind of request. It is typically a request for footage to be part of a TV programme such as a documentary that does not have any urgency and should be left for when the other daily tasks are concluded.

After the call is concluded, the archivist takes the most urgent request and opens an EDL [Har95] on Blue Order (an old MAM that was bought by Avid in 2010 and merged into Avid's Interplay). Then he tries to come up with a set of keywords that relate to the request. For every keyword, the archivist searches the database and selects the most important images by trimming each asset and submitting the cuts to the active EDL. As soon as there are enough images, the EDL is submitted and the task is concluded. After the processing is done, the footage will be available on a journalist workstation through the sQ Cut software. The journalist is not notified when the footage is ready, so he needs to manually pool the software for the EDL with his employee code until he finds the requested footage.

3.4.3 Raw footage archiving

One of the main tasks of archivists is to archive raw footage. As previously mentioned in section 3.4.1, this task usually has a 24 hour delay, since the assets captured on one day will only be archived on the next day.

Every day, the designated archivist creates a folder in sQ Cut for the day's workload. Then he browses for new footage in Porto's local server with a creation date previous to the present day and copies it to the desktop. One by one, he analyses the assets in quest for some interesting images for the raw archive, trimming them and saving them in the new folder. Every new generated archive should have a new MOS [Pre12] title that will be manually written in a sheet of paper with a short explanation of what those images are. Here lies one of the biggest problems of the archive department right now. Since these are raw images with no metadata whatsoever, many times the archivists do not know what the images are actually about and either try to guess by its contents or needs to ask for more information about those pictures from the reporter that recorded them. Fortunately, the title of the files includes the initials of the reporter. This way, archivists need to open ENPS, search for the reporter's schedule for that day and there is information about his workday that is usually very helpful to understand the footage. But since there is no way to digitally attach this schedule to the images, the archivists need to print this schedule and attach it to the initial sheet of paper with the codes and descriptions. When all the videos are watched

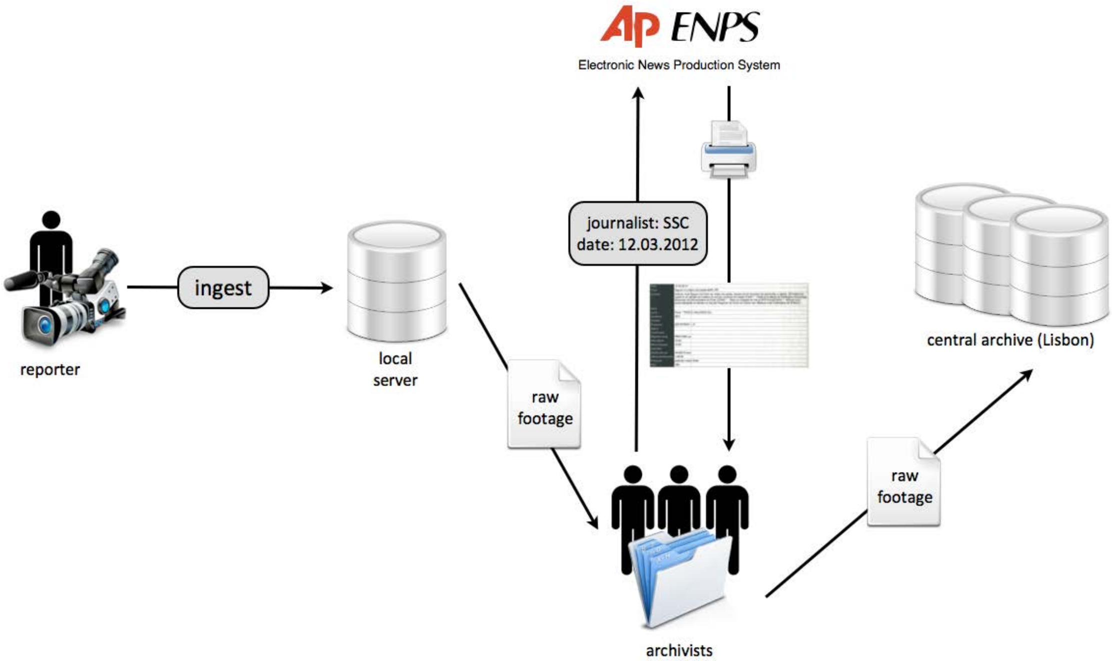


Figure 3.8: Workflow of raw footage archiving

and the information collected, they compile a folder with that day's work and add it to a queue with equivalent folders that need further processing. Concluded this phase, there is another delay for the new trimmed assets to be outgested (see Chapter 2.1.3.1 for a definition of outgest) to the central archive for further description and indexing. But this delay is not relevant since there is always enough older work on the queue to notice the detail. For the final archiving phase of raw footage, an archivists picks up the older folder of the previously mentioned queue with the MOS codes and some descriptions of footage to be properly archived. Entering each MOS code on the Blue Order software, they are able to locate every video and enter the adequate descriptions that were described in Chapter 3.3.3.

3.4.4 News clips archiving

The other main archiving activity is for news clips and TV programmes. We will focus on the first, since these are a bit more complex, but the second one also has a similar process so we will not explain it in detail. Firstly, an archive manager (at Lisbon) distributes the workload among the two archive departments (Porto and Lisbon). As for news clips, the Porto team usually gets the news clips from the afternoon news show while the Lisbon team archives the evening one. They receive this in a paper table with the MOS codes and a short description of the clips. Upon this, they must search each one individually in Blue Order by their MOS code. News clips need level 4 description, which requires full understanding of the content and can sometimes demand for some extra info.

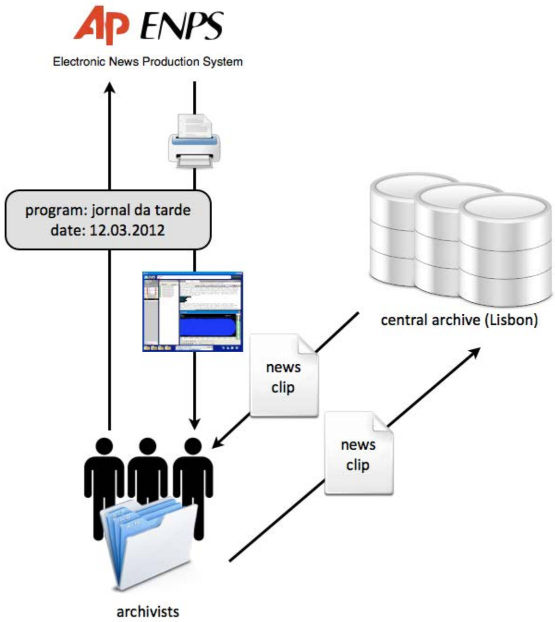


Figure 3.9: Example of indexing entry

When this happens, archivists open ENPS and search for the corresponding clip, also by its MOS title, in order to find many notes about the content, since they can find all the journalistic notes in there that the involved journalists took. But since they do not have two computer monitors and need to have the video open at all times, the notes are printed in order for archivists to consult them while they describe the clip.

When they finish describing, the files are submitted back to the central archive and marked as concluded in the previously mentioned paper table.

3.5 Problems

This investigation of the processes around RTP's news production and archiving allowed us to realize several problems that affect the efficiency and effectiveness of archiving. In this section we aim to sum up some of the problems, its consequences and finally which ones we will try to solve.

3.5.1 Inefficiency

On one hand, it is clear that this process is inefficient. Most of the problems were already presented in the previous descriptions, but we will now summarize them here and add some others that were not clear before.

Table 3.1: Checking ENPS for raw footage information

Title	Checking ENPS for raw footage information
Description	This problem, already described in Chapter 3.4.3, is that since there is no information link between the reporters and the archivists, the archivists get the raw footage without any information about them and typically do not understand its contents. When this happens, it is required for them to see the reporter's ENPS schedule for that date and understand where were those images filmed.
Consequences	Huge amounts of time are lost just searching for information that should already be there in the first place.
Observations	One of the many times this happened during our presence, the archivist in charge, trying to save time by not consulting ENPS, actually spent more time trying to figure out the location of one of the videos than the time he would take with that extra task.

Table 3.2: Checking ENPS for news clip information

Title	Checking ENPS for news clip information
Description	This problem, already described in section 3.4.4, is that since there is no information link between the journalist and the archivists, the archivists get the news clips without any information about them. Since these clips must be described with the deepest detail possible, it is important for archivists to gather as much information as they can. Therefore, archivists tend to open ENPS and search for journalistic notes of each clip, since important information can arise from there.
Consequences	Time is lost searching for information that should already be included in the video assets.
Observations	

Table 3.3: Repeated description work

Title	Repeated description work
Description	Description work is repeated every time indexation is needed since the indexed terms are always repeated in the text-based resumes.
Consequences	Time is lost since work is doubled and sometimes trebled.
Observations	By looking at Figures 3.3, 3.4 and 3.5 we can see that "hospital garcia de horta", the name of the hospital where the footage was shot, is repeated three times.

Table 3.4: Lack of time-based descriptions

Title	Lack of time-based descriptions
Description	Time-based descriptions are a common activity in archived footage. But since there is no functionality on their software to facilitate such descriptions, time-based descriptions need to be manually inserted in text form.
Consequences	Time is lost since instead of clicking a button that would automatically set the actual time and would just require additional description, the video needs to be stopped so the archivist can copy the time and write the description, resuming the video only when the task is completed.
Observations	As shown in Figure A.3, the analytic resume of a TV programme is full of time-based notes. Some of them even refer to other news clips with known MOS codes, but all this information needs to be manually inserted.

Table 3.5: Repetitive events are not taken advantage of

Title	Repetitive events are not taken advantage of
Description	There are many repetitive events that have to be manually inserted many times throughout an asset description. We included an example of such an asset in Appendix A.2 Figure A.4 where we can see many repetitive events that had to be constantly time-noted and re-written, such as goals offsides, penalty cards, etc.
Consequences	Time is lost since work could be accelerated and although repetitive events could have a fun/easier approach, they are actually seen as the most boring ones, since most of the time is spend rewriting such repetitive words.
Observations	Indexation, if properly used, could be an advantage since we can see that it is possible to index a goal from a specific player. But they are still required, for no reason at all, to re-write the event in the analytic resume.

Table 3.6: Unwatchable video previews

Title	Unwatchable video previews
Description	Since describing or searching for a video, demands that the video is watched, Blue Order includes a preview module that allows instantaneous video preview. It's quality is good enough for searching, but description demands a better quality version since the extremely low-resolution of these videos don't allow archivists to understand important details.
Consequences	Gigantic time loss every time this happens. When archivists cannot understand some detail, they will request for a full-resolution version of the video by simulating a request for images (Chapter 3.4.2). Then they need to go to the only computer of the department that is able to run sQ Cut and watch the full resolution to clarify what previously was not clear in the images.
Observations	This typically happens in news clips where names of people are written in a graphic caption that cannot be read in low-resolution video and such information is also not written on the news clip's ENPS notes.

3.5.2 Indexation problems

Table 3.7: Untouchable thesaurus

Title	Untouchable thesaurus
Description	As previously explained in Chapter 3.3.3, indexation is based in terms from a concept list called thesaurus. The problem is that this thesaurus is practically untouchable, since the introduction of new terms can only be made by the chief archivist in Lisbon. This requires a formal request for a new thesaurus term, which has to be asked by the chief archivist of Porto. These requests are usually considered non-important and therefore denied.
Consequences	Indexation, which would be far superior in efficiency, is limited to a restricted set of terms.
Observations	There is an example of a gap existent in RTP's thesaurus, that is almost seen as a joke among the archive workers: since coffee shop and coffee (the roasted bean) have the same name in portuguese - Café -, Porto's archive suggested that there should be two entries, one for each, since they are both very popular among requests. After making the proper request, this was denied and they were asked to describe a coffee shop as a shop. Nowadays it is almost impossible to quickly find a coffee shop in RTP's archive, since a query for shops will mostly return other shops like clothes shops or grocery stores.

Table 3.8: Non-used indexation

Title	Non-used indexation
Description	Since indexation is a source of frustration for the archivists, they tend not to use it al all. In fact, many of them already forgot how to use index-based search.
Consequences	Indexation, which would be far superior in efficiency, is negatively seen as useless work.
Observations	

3.5.3 Information loss

On the other hand, footage that was supposed to be archived is sometimes lost. This is an even worse problem, since captured video and audio are the main assets of a content producer and when these are lost, money is also lost.

Table 3.9: Quick-and-dirty raw footage selection

Title	Quick-and-dirty raw footage selection
Description	As described in Table 3.1, raw footage selection usually demands for ENPS consultation. But even when this happens, sometimes ENPS is not used. Frequently, archivists try to save time by selecting to archive the footage that they understand and the ones that would demand more information are just discarded.
Consequences	Money is lost since assets that should be archived are deleted instead.
Observations	

Table 3.10: Lack of communication between reporters and archivists

Title	Lack of communication between reporters and archivists
Description	Since there is zero communication between reporters and archivists, archivists happen to receive footage with special details that are lost since they can be considered irrelevant.
Consequences	Money is lost since assets are deleted when they shouldn't.
Observations	This problem was realized when we were observing the activity of raw footage selection and noticed a close-up of a famous Portuguese composer in a political congress. Upon realizing that the archivist ignored that footage, we asked him why wouldn't he archive that footage. He answered he was not sure who that was and since it was not written anywhere, they could not be sure. Days later, as our investigation moved on to journalists, we found the journalist who was in charge of the news clip that came up from that raw footage. After telling him what happened, he could not believe those images were deleted, since they were filmed for a reason and, in his opinion, that close-up seemed obvious enough that it was for archive purposes. Because of this lack of communication, if images from that composer are needed in order to connect him to the political party in question, they will never be found even though they were filmed.

3.5.4 Conclusions

In conclusion, we found many problems that create delays and losses in the process. Some of these caused by lack of software functionality, others by bad habits and wrong work approaches.

In the next chapter we will explore a solution that should solve some of these problems.

Chapter 4

Proposed solution

In the last chapter, we reported our investigation at the RTP studios, expecting to learn more about the work developed there and in quest for problems to solve around the theme of metadata and touchable interfaces.

In this chapter, we will discuss our approach to the problem and explain the solution we came up with. Furthermore, we will explain the methodology used, based on user centered design, as explained in section 2.4.2.

4.1 User Centered Design

If user-centered design has clear advantages in designing any system to be used by human beings, a system like the one we are aiming to develop is a perfect match for such a methodology. Our view is that although most problems come from lack of functionality and lack of connection between systems, one of the biggest problems will be to convince three different kinds of users - namely, reporters, journalists and archivists - into changing their actual working ways and probably add work to reporters and journalists in order to relieve the work overload of archivists. This change on traditional ways and the addition of a new task can only work with a system in which the advantages are clearly perceived by all users. By involving the future users in all the phases of the development process should cause the idea that this is their product, that they helped to develop, thus having a clear advantage in terms of acceptance.

After the literature review phase that originated the state of the art chapter, our approach was develop an in-site study of the work developed at RTP. Based on the studied approaches for UCD techniques, we adopted a mixture between Ethnography and Coherence, by performing an ethnographic study, but having in mind the work we wanted to develop, although not defining clear “viewpoints” and “concerns”, since we had in mind the development of a solution that better fitted their needs by taking advantage of new technologies, but not a clear viewpoint of what exactly should be improved.

This ethnographic study was taken in two days, 2012/04/11 and 2012/04/12, spent in the RTP studios of Porto. The first day was spent on a morning in the archive followed by an afternoon in the news department. The next day was fully spent in the archive department. During both studies, we fist asked them to explain their work, taking notes of their words, and then tried to interfere the least possible as we observed their work and logged as many details as we could.

After each session, the hand written notes were "translated" into real sentences and diagrams so as to consolidate our observations and guarantee they would not be forgotten. This allowed a very smooth transition from observation into requirements elicitation. After defining the requirements of our solution, we validated them first with MOG's representatives, since they would always have the final vote on the developed solution. We decided not to validate them immediately with RTP since the proposed solution had a strong interaction between three entities - reporters, journalists and archivists - that do not know enough about each other's jobs - specially between reporters/journalists and archivists - to validate only textual requirements. It was decided that we would only validate the solution only after building some prototype that would be better understandable by users.

After validating the requirements, we started designing an interface for the solution we had in mind. This was firstly done using paper prototypes, which allowed more freedom to explore different solutions and approaches. After many experiments, with the help of other software engineers at MOG we selected one of the solutions and re-draw it so we could do a heuristic evaluation on paper. After some input from our coworkers, we implemented a medium-fidelity prototype using the open-source Pencil [Evo12] software. This mockup had the ability of being exported to an interactive PDF file that, when loaded in a compatible tablet computer, could emulate our intended solution. This allowed us to take the mockup to RTP and test it with users, validating both the requirements, the concept of the solution and the user interface at the same time. With the additional requirements and input on the interface, we redesigned the solution and re-implemented another set of mockups. Upon another test at RTP, we validated the requirements and the solution, having only minor interface details that did not demand for another cycle of medium-fidelity prototyping and allowed us to go on to implement the solution.

4.2 Requirements

Based on the problems we saw at RTP, we established some simple requirements for our solution:

Completely unify the work of three entities In order to never lose any information, our solution should have a centralized approach so there is never the risk of information lost.

Allow video-wide metadata Almost all current descriptions concern the whole video, so video-wide metadata was mandatory.

Help data insertion with typically used formats Data insertion has many common formats, like numbers, text or dates and the input forms can be personalized so as to accelerate the noting activity.

Allow time-based metadata This kind of metadata did not exist on the actual systems but was manually introduced. Our system should have a personalized way of adding such metadata.

Log as you watch The logging activity was previously separated from the watching activity and both were never done at the same time. Our system should find a way of allowing real-time logging.

Mobility Reporters cannot have a computer in their hands at all times. A mobile approach is the solution to that problem.

Take advantage of repetitive events As we could see, there are several contents composed of repetitive events like the example of a football match. Our system should take advantage of such repetitions to allow faster logging.

4.2.1 Speed Marking

Speed Marking is a concept we came up with to solve the last of the requirements previously presented. It is based on an idea from a feature of MOG's mxfSPEEDRAIL S1000 where an user can set the F1 to F10 keyboard keys to log specific text strings associated with the actual timecode they are pressed at. These so called "locators" allow metadata to be inserted into an SDI stream in real time. We took that concept and extended it into personalizable lists of concepts for different kinds of happenings such as football matches or political debates, where there can be a pre-defined list of all possible events and allow real-time logging without losing any detail. Even though this concept seems quite simple, it was never deployed into market as a general solution.

Figure 4.1: Screenshot of Cinergy Media Desktop

The only existent solutions are personalized software/hardware for specific uses, like the one shown in Figure 4.1, containing several buttons for cinematic events and actors from the TV series Pride and Prejudice, that helped to achieve a faster logging. By using tablet computers we can further improve this concept by using a touchable interface where the archivist simply touches the desired event. Furthermore, we also allow infra-concepts, making it possible, for example, to not only mark goals in a football match, but also who scored it.

4.2.2 Integration with mxfSPEEDRAIL

Another requirement was the integration of our solution with MOG's mxfSPEEDRAIL. This requirement could be easily fulfilled, since our proposed solution would always need an ingest system that centralized the videos in a networked location accessible by a mobile device. mxfSPEEDRAIL turned out to be a perfect match, since, as well as it performs the ingesting task, it also includes a web server where we can host our service and even generates MP4 proxy videos that are compatible with the latest HTML5 standards.

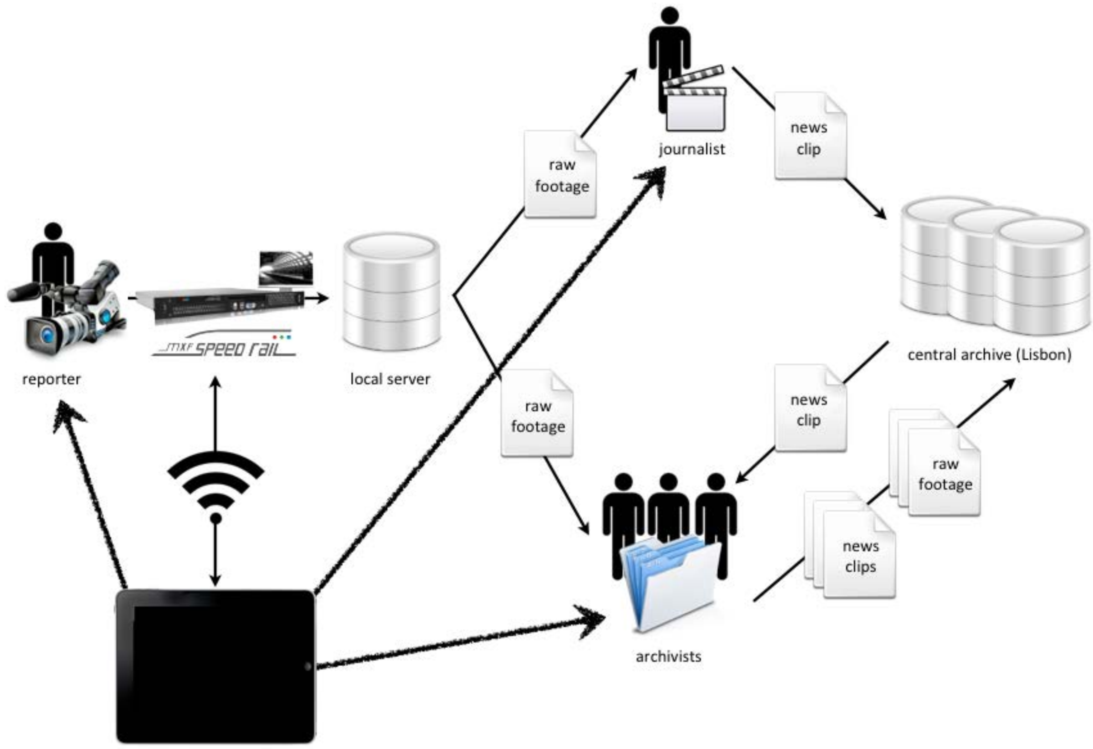


Figure 4.2: New workflow integrating mxfSPEEDRAIL and our solution

This way, as soon as videos are ingested using S1000 or F1000, all users can access them via a tablet computer and add as much information as they want. This new workflow is illustrated at Figure 4.2. This should be done firstly by the reporters, right after ingesting, then by the journalists after reviewing the footage and signal what they believe is important and finally by the archivists for the final archiving stage. These last ones, instead of receiving an asset with no description at all, should now receive an asset filled with metadata that would have to be inserted by them.

4.3 Low-fidelity prototyping

As soon as we got our first requirements and solution overview approved by MOG's representatives, we began to sketch out the interface items that this system would need.

4.3.1 Asset list

As soon as the user launched our application, our initial idea was to show all the ingested assets in the S/F1000 they are connected to. We also provided the ability to filter the results based on some typical criteria such as asset title, file name, creation date, descriptions, etc.

As illustrated in Figure 4.3, each asset would be shown as a composition of a preview picture taken from the video as well as it duration, its title (or file name if there was no title defined yet), creation date and a snippet of the description.

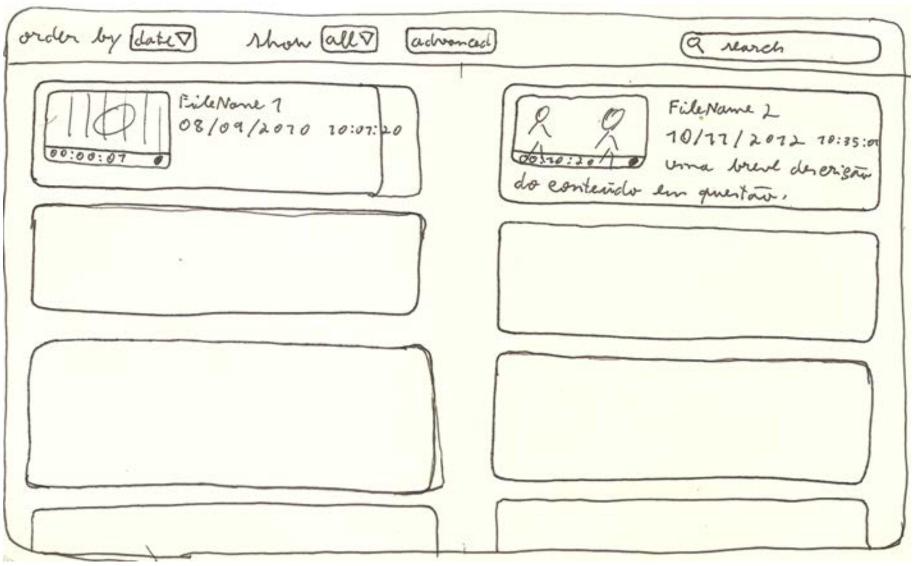


Figure 4.3: First mockups of the initial asset list

4.3.2 Video preview

When the user tapped one of the assets, the two-columned list would become single columned on the right side of the screen and from the left side would appear an area with the selected video, as shown in Figure 4.4.

This seekable video would allow the user to see its contents as well as metadata details. The metadata was separated in two different areas, video-wide and time-related. The first we called just metadata, since this is the kind of information that users typically identify by that name. The second we called markers, since they mark only some part of the video and not the whole asset. This information would be displayed in the right half of the screen, where the assets previously were. If a marker was tapped, the video would go to its starting point and start playing from there.

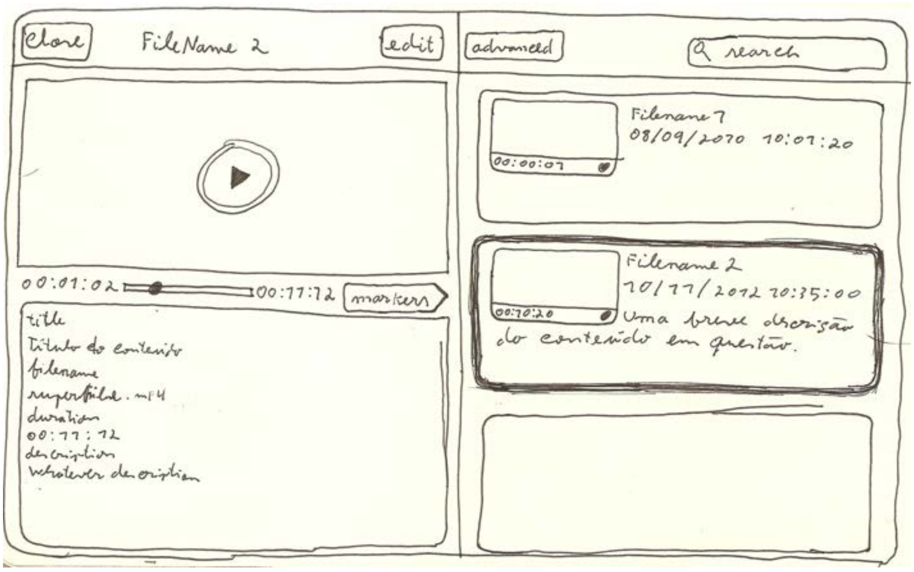


Figure 4.4: First mockups of the asset list with the video preview

4.3.3 Edit mode

We decided it was important to separate the editing mode from the browsing/preview mode. This way we could allow users to do anything in the initial mode without having the risk of doing any mistake, since they were just consulting existent information. Edit mode was seen as the core of our application, since this was where metadata would be inserted and therefore our biggest challenge. This editing mode was itself composed of three different modules:

4.3.3.1 General metadata

Our first module is the one we called "general" for general metadata. It includes the type of fields that already existed, but with two advantages. One of them is the ability of adding new fields: if one wants to add the Country field, that did not exist in RTP's solution, clicking on an empty field would ask what does the user want to do with it - this will be better illustrated in the medium-fidelity prototypes. The other advantage is to have different input ways for different input types. For example, if one is supposed to insert a date, a calendar will pop out and help its insertion, instead of just asking for a simple string of text.

As we can see in Figure 4.7, a video preview would still be available on the left in a smaller window where also the technical (thus unchangeable) metadata would be displayed.

4.3.3.2 Markers

The second module was named "markers" and this is where we create markers that can make a piece of text refer to an instant of the video or an area of the video from a timecode to a later one.

A list of the already created markers would be displayed on the left and if one of them was tapped, it would be disclosed below the video, as illustrated in Figure 4.6. Our initial idea was that

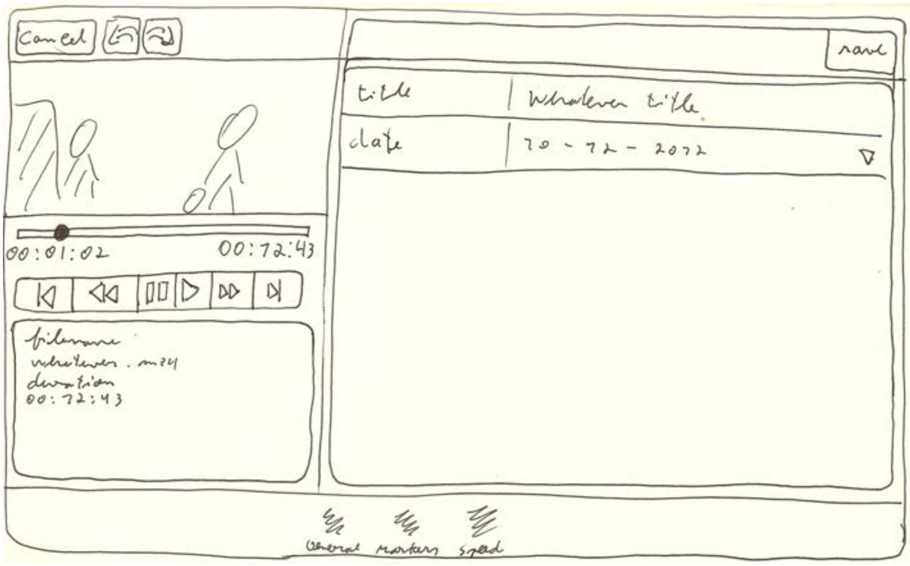


Figure 4.5: First mockups of the general metadata edit mode

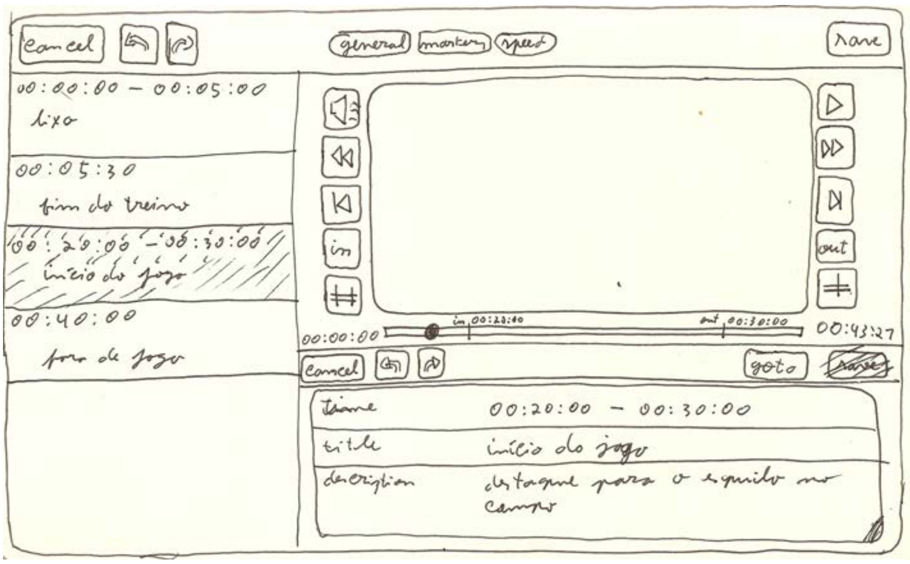


Figure 4.6: First mockups of the markers edit mode

the video should be placed on the right since this was always the “main” area of our application and video watching would be essential in this module since we need to locate the markers to certain moments of the video.

4.3.3.3 Speed Markers

The last module was called “speed” due to our coined term of “speed marking”.

Our idea was that it was needed to have a way of logging metadata for repetitive events very efficiently and in Figure 4.7 we show how a yellow/red card could be logged, by just clicking

Figure 4.7: First mockups of the speed markers edit mode

“cartões” (“cards” in Portuguese) and selecting either “vermelho” or “amarelo” (red or yellow) which would create a marker in the moment they were selected.

4.3.4 In-house heuristic testing

As initially planned, we developed a heuristic testing session with some of MOG’s software engineers. These tests were individual and performed informally since all of the evaluators, although not specialists, had some HCI background thus being able to quickly give good advice without the need of instructions. Since we did not have a design team to discuss the problems that were noticed, every time a problem came up, we would discuss how to solve it with the person that noticed it. By using paper prototypes on this phase, we redesigned every interface that had a problem right at the moment and presented the new sketch to the next evaluator. This way we quickly advanced many iterations in just one afternoon, since a design iteration would take just as much as to sketch a new interface on a piece of paper, meaning just a few minutes.

4.4 Medium-fidelity prototyping

After validating our interface sketches, we moved on to medium fidelity prototyping. Since the implemented prototypes are very similar to the paper ones, we will not illustrate them all. Instead we will only present the changes caused by the user tests performed at RTP.

4.4.1 User testing and redesign

As previously mentioned, these prototypes were implemented in Evolus Pencil and exported to an interactive PDF that was loaded into a tablet in order to allow interactivity between the user and the system. This allowed us to evaluate the user interface’s usability by asking RTP workers to perform a set of tasks that we defined in a script. Comparing their solutions with our ’ideal’ solutions, we were able to detect many usability mistakes. Their spoken input was very helpful as well, since the users felt happy for being involved and wanted to participate in the design themselves.

4.4.2 Login

Starting by the initial screen, this was ironically one of the features we did not think it should be included. In fact, we thought about it and saw no advantages in having a login system, since the application would need a more secure authentication system such as a VPN making the login only another burden for the user.

But as we discussed the asset list with the users, we found out that different entities - reporters, journalists and archivists - have different views on what should be immediately available as soon as they enter the system. The reporters would like the videos that were captured by them. The journalists would like to see their news clips and the raw videos from where those clips were built from. Finally, the archivists would like to see the work they need to do, ordered by urgency. This reason clearly justified the existence of a login phase (Figure 4.8) where the authentication would differentiate the three entities, personalizing the asset list for each.

4.4.3 Asset list

The asset list, as previously explained, also had some major changes, specially concerning its contents, since the assets presented in here were initially thought to be the assets contained in the machine they were connected to (Figure 4.9).

After discussing with all the workers and realizing that the ideal asset list would be the one trimmed to each worker's requirements, we redesigned the asset list. Implementing a system of different filters selectable on the upper left corner of the screen (Figure 4.10), these filters can be customized in case the user wants to have access to some other content different content. This is important in order to allow work flexibility - it is not very dangerous to let a user access other user's assets, but it would be very harmful for the organization if a worker called in sick and no one else could easily access his assets.

Furthermore, another small change was the small coloured dots next to the asset duration (Figure 4.9) which we thought about making them symbolize whether an asset was still empty in terms of metadata (red), modified but not concluded (yellow) or ready to be archived (green). Apparently this caused much confusion to the archivists, since they were accustomed to think about those three colors in terms of rights-management. Since the general opinion was that they would not need to know information about the metadata state of an asset, we just removed that feature since in order to create less noise on the screen.

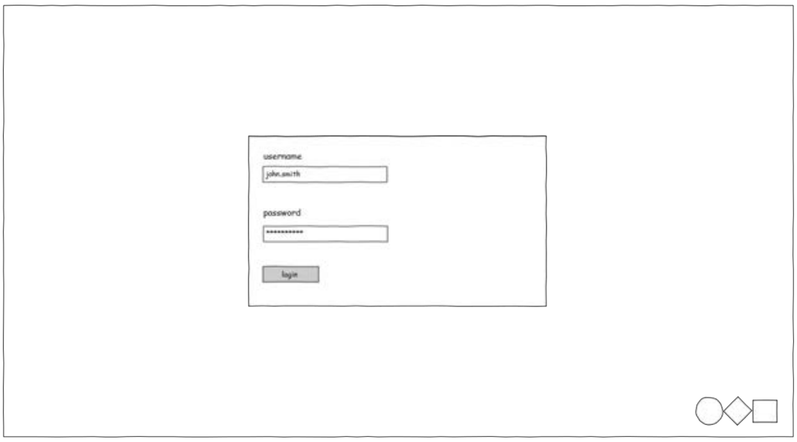


Figure 4.8: Mid-fidelity prototype of login screen

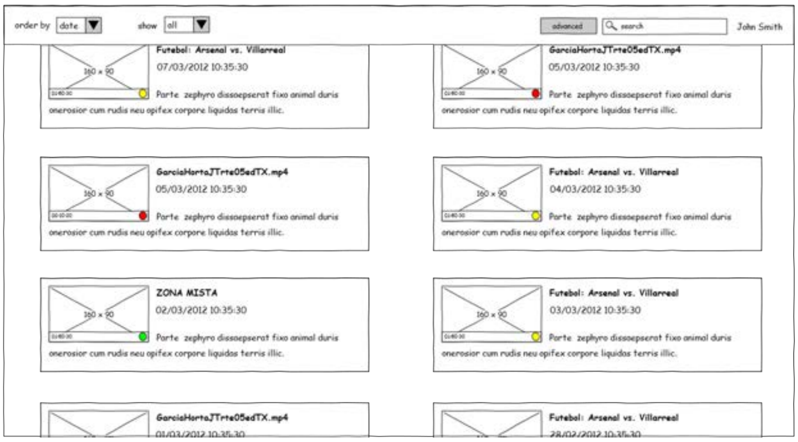


Figure 4.9: Mid-fidelity prototype of asset list before user testing

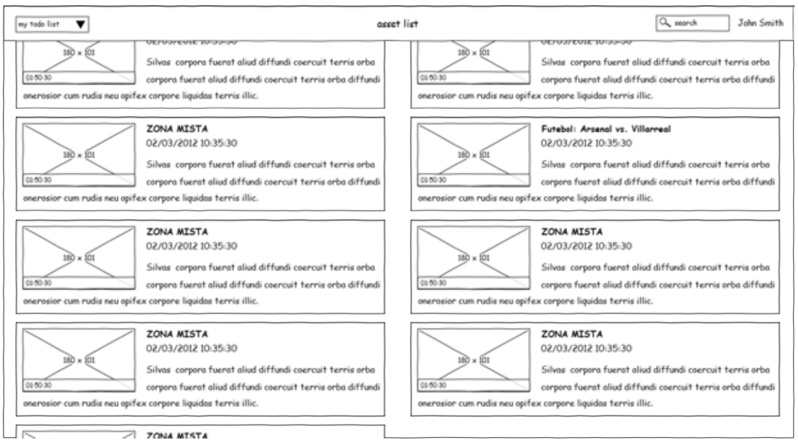


Figure 4.10: Mid-fidelity prototype of asset list after user testing

4.4.4 Preview

The preview screen suffered two main changes. The first one happened with the transition between the low-fidelity prototype of Figure 4.4 into its medium-fidelity version of Figure 4.12. The button that discloses the markers on the preview screen was changed from the right of the seek bar into the left of the seek bar. Although this change may seem irrelevant, it had a major impact in the rest of the design.

When we were showing the prototype among MOG's engineers, they would automatically try to use it in search for problems, while still standing up. But because the usual way of handling a wide-screen tablet is with both hands on the back of the screen, leaving just the thumbs for screen tapping (as shown in Figure 4.11), we noticed that users could never reach the buttons that were located in the middle of the screen. In our case, users would either re-grab the laptop in another way, using just one hand or just land the laptop on a table in order to tap the button. Neither situation is desirable for our system, specially since reporters should be able to use this in rough conditions where they may have no table and may even require both hands to handle the device with improved stability. After this problem, we made sure this error was not repeated by adding this rule to our now personalized heuristic tests.

Another change that happened after user testing was the addition of a label containing information about who was the last person to edit that asset as well as its last modified date. This was requested by both journalists, in order to know if someone else "touched" their assets as well as archivists to be able to track down eventual mistakes.

4.4.4.1 Metadata and markers preview

The other major change happened between iterations of the medium-fidelity prototyping, differences that can be easily noticed by comparing Figure 4.12 with Figure 4.13. First, the size of the video window was changed. This happened because we were informed that almost all videos shot at RTP have a 4:3 width/height ratio instead of the 16:9 ratio that we initially supposed. This change demanded the video window to take more vertical space, causing the metadata information box to disappear. This gave space to another user's request, better video controls, now including frame by frame, rewind and fast forward buttons.

In terms of metadata visualization, the previously mentioned box disappeared as well as the markers display (Figure 4.14). They were both replaced by two distinct displays on the right, one for general metadata that used to be on the removed box (Figure 4.15) and another one for markers (Figure 4.16). Tapping a marker would cause the video to go to the marker's start point.

Figure 4.11: Person handling a tablet with both hands

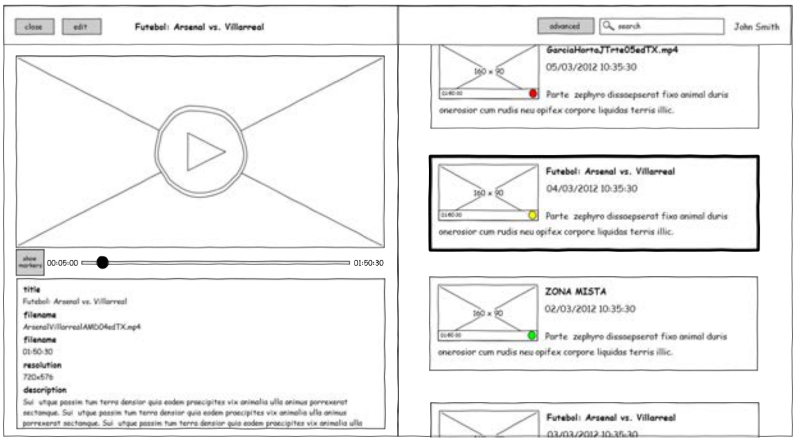


Figure 4.12: Mid-fidelity prototype of asset list with preview before user testing

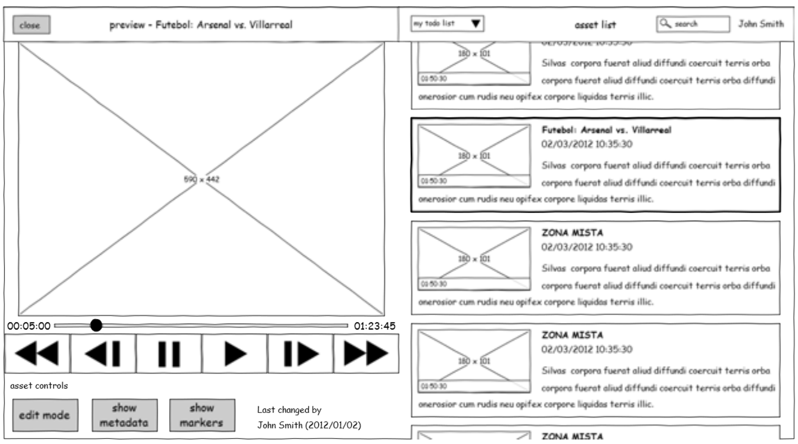


Figure 4.13: Mid-fidelity prototype of asset list with preview after user testing

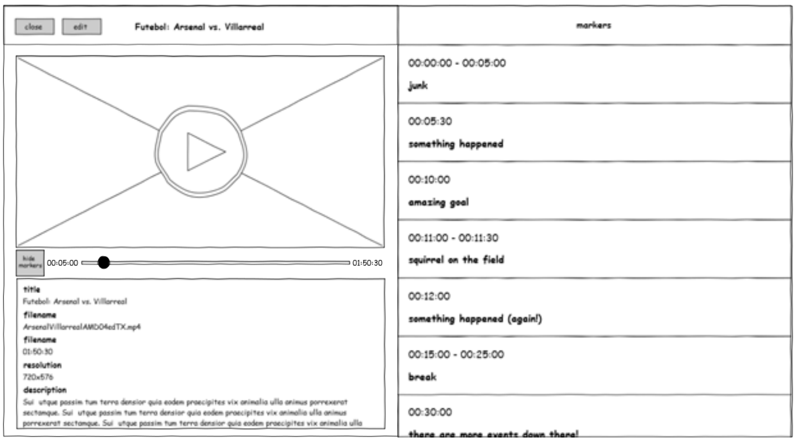


Figure 4.14: Mid-fidelity prototype of video preview showing markers before user testing

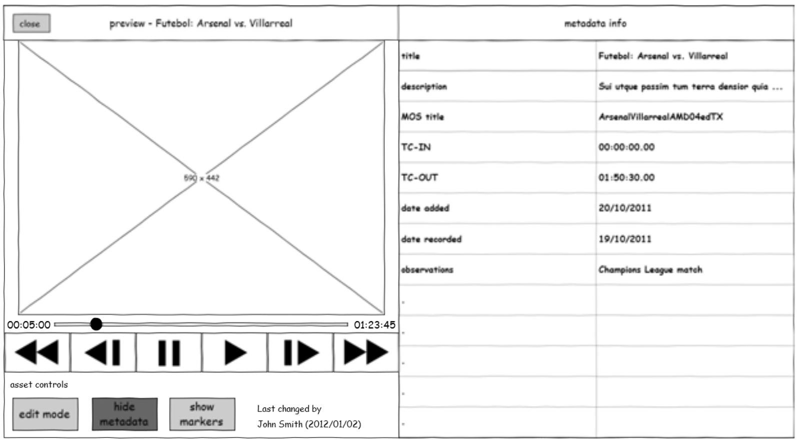


Figure 4.15: Mid-fidelity prototype of video preview showing metadata after user testing

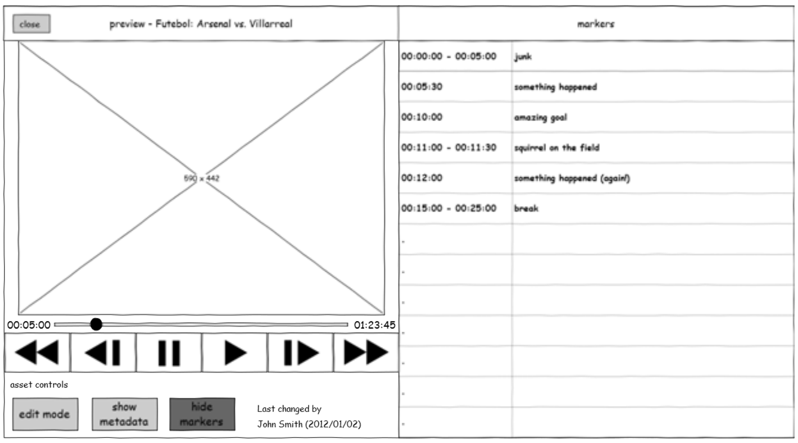


Figure 4.16: Mid-fidelity prototype of video preview showing markers after user testing

4.4.5 Edit mode - General metadata

Entering the edit mode, we start the application with the section of general metadata, where we can add different fields of text, numbers or text, as well as edit or delete existent ones.

We also made some major changes in this interface after user testing. The first one, which is probably the most noticeable, is that the left part of the screen in Figure 4.18 is now equivalent to the previous preview screen. This was a major change throughout all the application. Users felt very uncomfortable with the constant changes in video size and location, advising us that the video should always be in the same place. Therefore, we adopted a different user experience strategy. Now, as soon as a video is selected from the asset list, that left part of the screen will be static until the video is closed again. Apart from the lower row of action buttons on the left, it is now the right half of the screen that can be modified. Another function that clearly failed our tests was how to add a new metadata field. Users could never figure out that the plus-signed button on the top-right corner of the screen (Figure 4.17 was the one they should click. Instead, they always tried to click the empty fields in order to add a new one. Since this idea made much more sense that our initial design, we implemented this new way of adding metadata. We also added a button with the same functionality on the left lower row, because if the video was full of metadata fields, the last (thus "empty") metadata field would not be viewable at all times, justifying this redundancy. The last change was the technical metadata box that disappeared from the bottom-left corner. We thought this information could be useful, but every tested worker told us otherwise. Once again, information on the screen that is not needed is considered noise and should be removed.

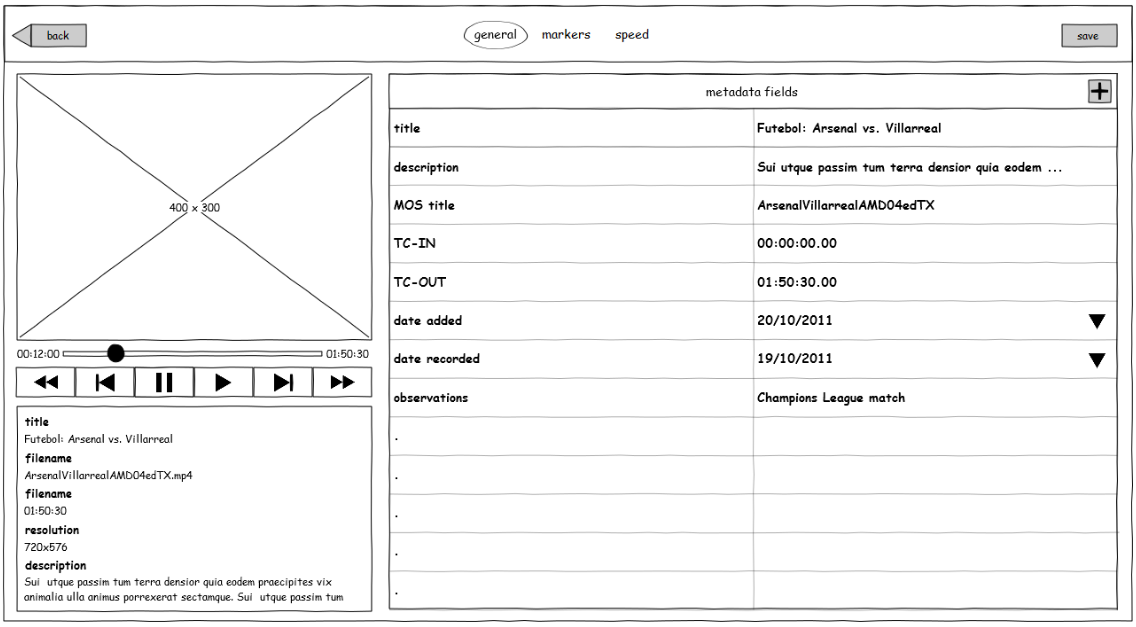


Figure 4.17: Mid-fidelity prototype of general metadata editing before user testing

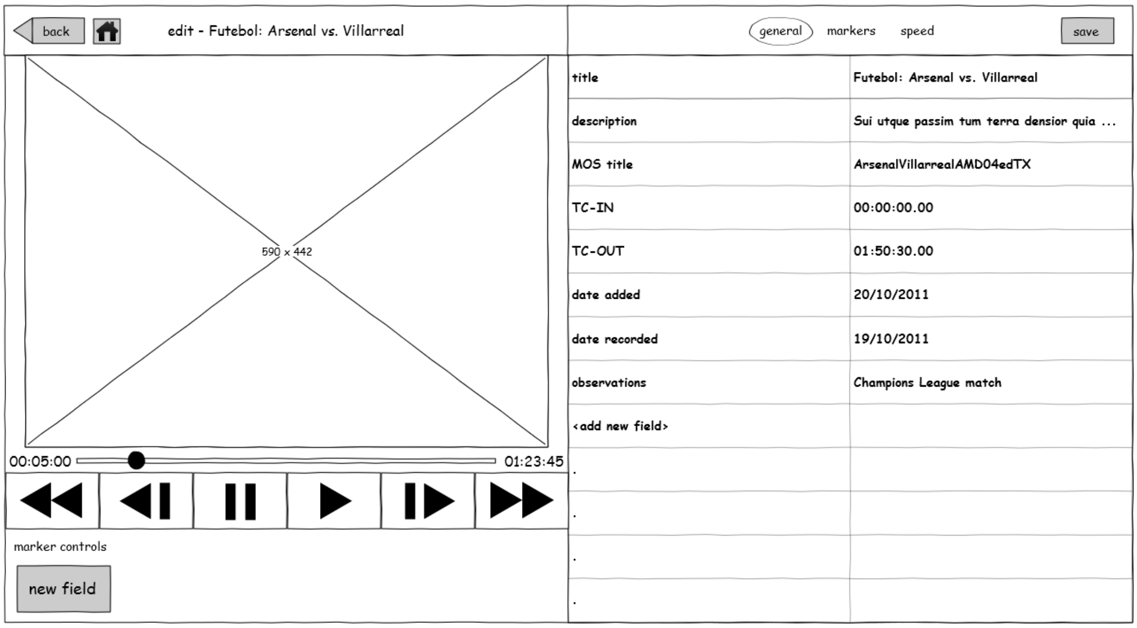


Figure 4.18: Mid-fidelity prototype of general metadata editing after user testing

4.4.6 Edit mode - Markers

Selecting "markers" on the tab selector in the top bar of the screen takes us to our marker editor. Once again, there were several changes to the UI. The first problem was the screen location, that was moved from the left side of the screen to the right side of the screen. Users immediately felt uncomfortable and asked why would we do that. The reason was simple - we were trying to concentrate the main area always on the right. While the main area for general metadata were the input fields, to insert markers it is more important to actually be looking at the video, rather than the forms, so as to better choose the exact moment where a marker should be located (Figure 4.19). But since this strategy failed with everyone without a detailed explanation, we decided not to adopt it, in favour of the fixed left side video window (Figure 4.20).

Another problem was our idea of filling the two columns at both sides of the video with buttons related to video control. Since there is a typical association of left/right with backward/forward, we thought it would be a nice idea to associate the two sides with corresponding buttons like pause/play, frame back/forward, rewind/fast forward, etc. But this concept did not turn out to be clear and was disapproved by most users. We adopted the video controls from previous screens and just added the bottom buttons, divided in two clusters. The first button generates an empty marker at the current time code. The other three work together to create a marker between two times. The user should now first tap the in button which assumes the starting time code, then the out button when he arrives the end time code, submitting the dual marker by tapping the middle button "mark from in ... out". All the timecodes are shown at the list in the right and tapping one of them reveals a small window from the bottom-right corner of the screen to modify its content.

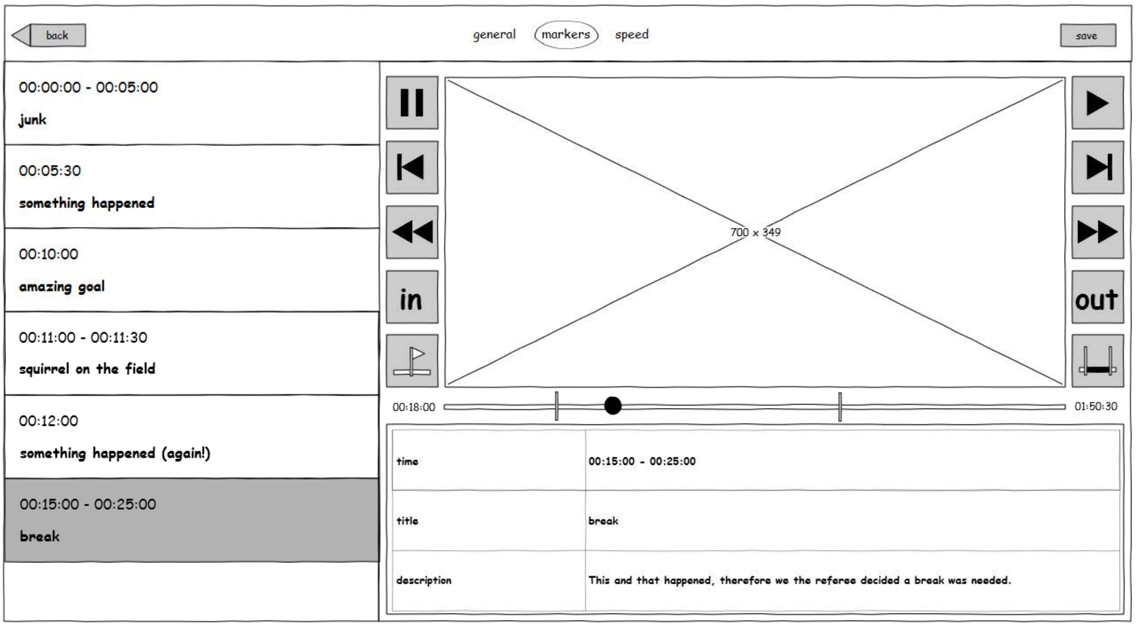


Figure 4.19: Mid-fidelity prototype of markers editing before user testing

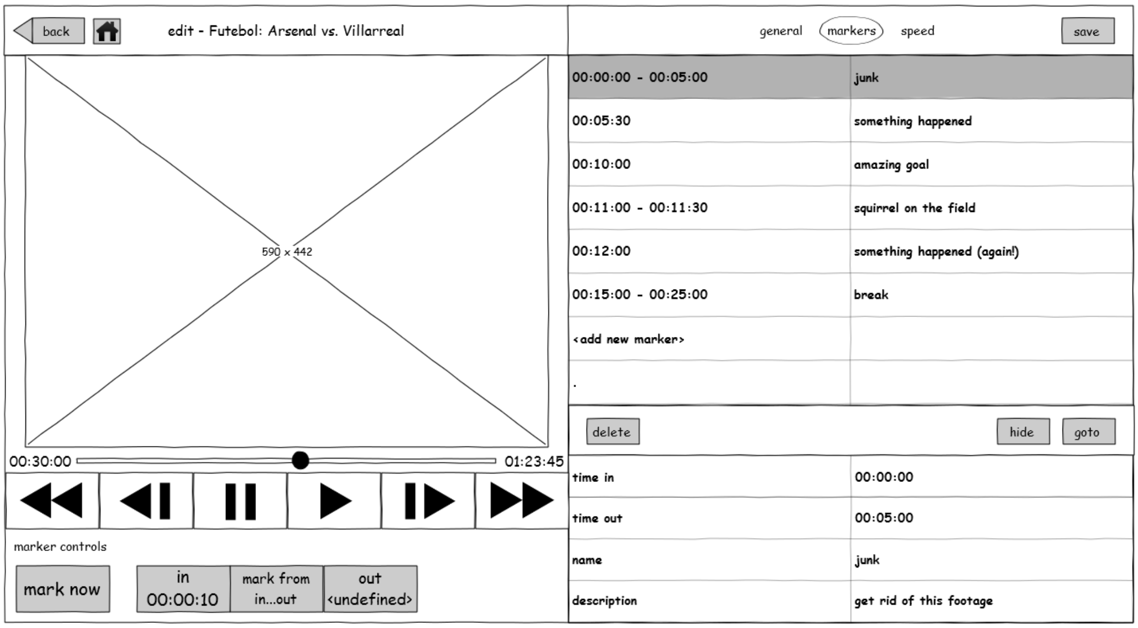


Figure 4.20: Mid-fidelity prototype of markers editing after user testing

4.4.7 Edit mode - Speed Markers

Our concept of Speed Markers, as explained before, was the most complex of the system since it is also the most innovative of them. Since we did not have many other systems to base our interface study, we decided that:

- There should be a way of building lists of concepts that would help the logging activity to be accomplished in a much faster way;
- These lists should be reusable and customized;
- The concepts should be interlinked according to necessities (e.g. to have a player linked to a goal);
- Due to the complexity of the previous requirement, these concepts would not need to be created in the tablet computer.

These simple requirements served as a basis for the development of our solution. We will now present our first approach followed by its reformulation based on the user testing.

4.4.7.1 Before user testing

Since we defined that users could either build a personalized list or reuse a previously built list, we first ask users if they want to browse existent lists or build a new one (Figure 4.21). Selecting the "browse lists" option takes the user to a screen where he can browse for lists on the left panel seeing its contents on the right panel (Figure 4.22).

In other hand, if the user chooses to build a new list, he will see the whole library of concepts on the left side and add them to the new list, shown at the right side of the screen. The library is represented as a nested list which, like a tree data structure, has nodes and leafs. Adding a leaf is just like dragging the concept from the left list to the right list, but adding a node is also possible. As we can see in Figure 4.23, when we tap the add button for a node which is not a leaf, the user will be asked how should those leafs inside the node be added. In Figure 4.24 we can see both examples. The different icon for the first item on the right panel shows us that this is not only one concept but a collection of them, since when the "Referees" node was added, the "collection" option was chosen. Choosing the "one by one" option makes each leaf be individually added, like we can see from Figures 4.23 and 4.24, where "Penalty Cards" were added "one by one" and therefore we see two new concepts on the list: "Red Card" and "Yellow Card".

After the list is ready or has been chosen from the list browser, we can begin the logging activity. Tapping a speed marker (the name given to a concept on the list), the application gets the actual video time code and creates an instance of that concept as a marker with that time code. The only different scenario is if a collection is tapped instead. In this case (Figure 4.25), a bubble list pops up from the collection and shows us the leafs inside that collection. Tapping one will cause a new marker to be created and in the bottom-right corner a window appears for further detailing.

This information can be edited at the moment or later by selecting that marker in the bottom-left marker list.

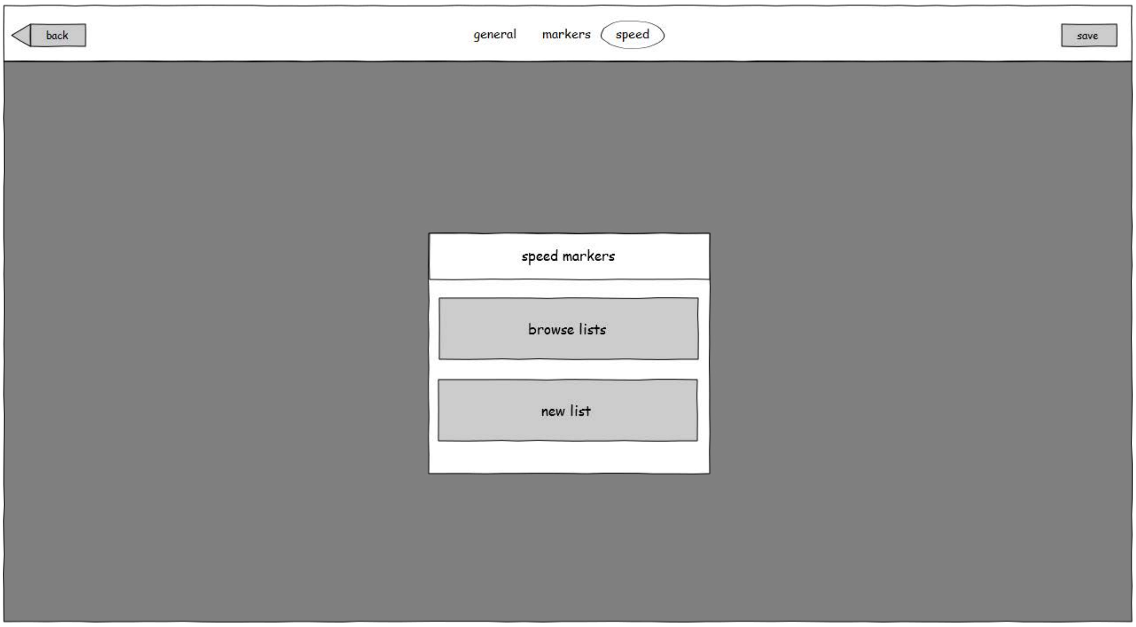


Figure 4.21: Mid-fidelity prototype of initial speed marking screen before user testing

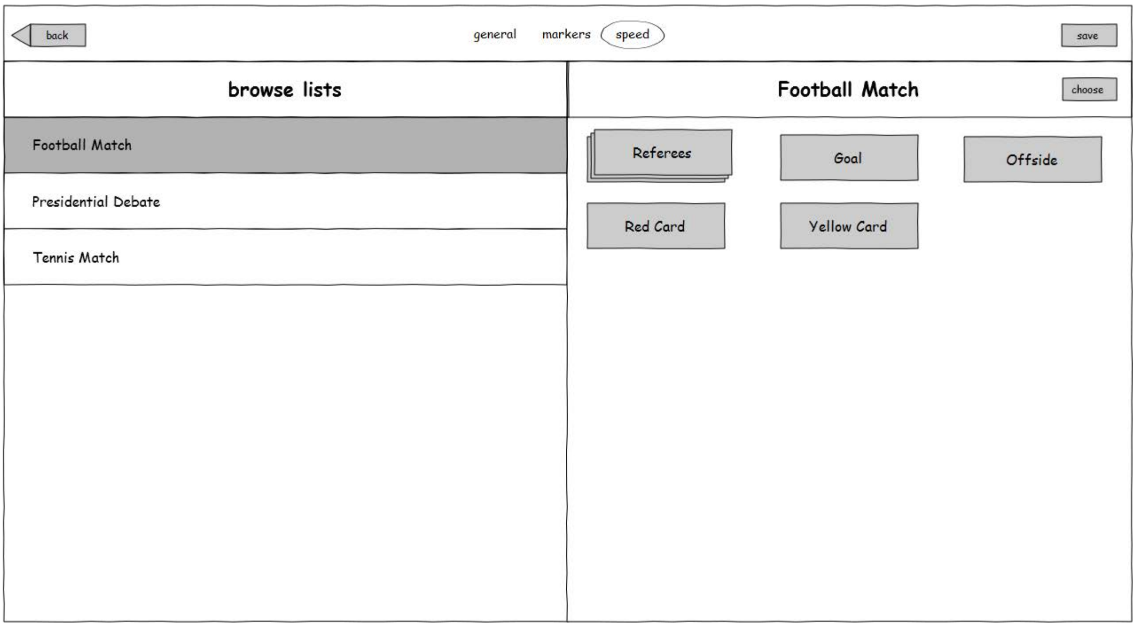


Figure 4.22: Mid-fidelity prototype of speed marker list builder before user testing

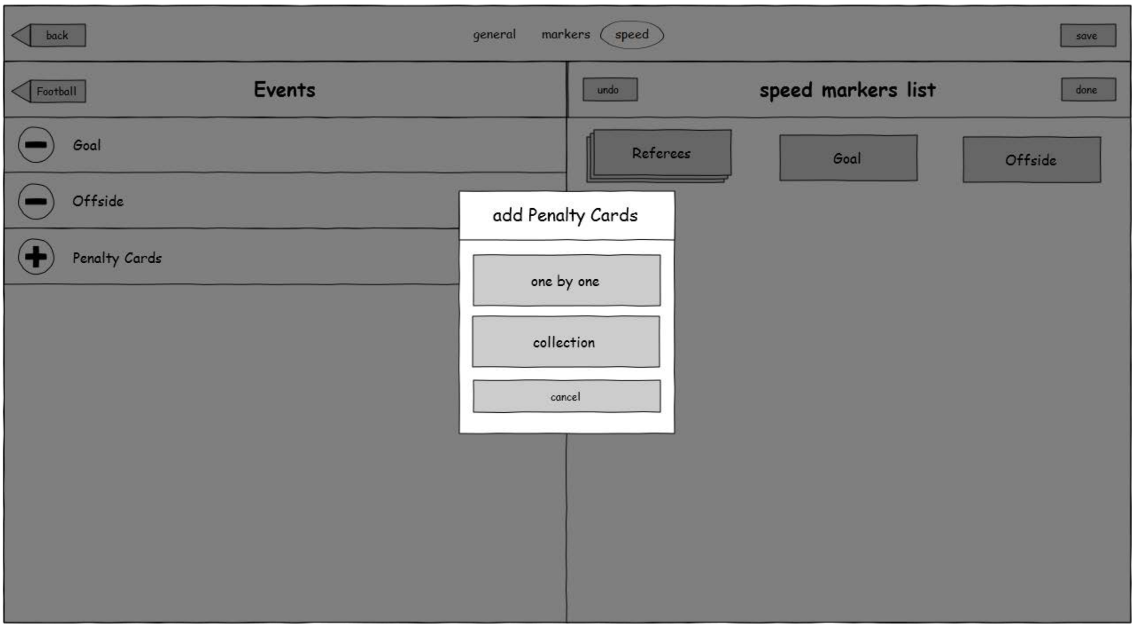


Figure 4.23: Mid-fidelity prototype of speed marker list builder adding a collection of markers before user testing

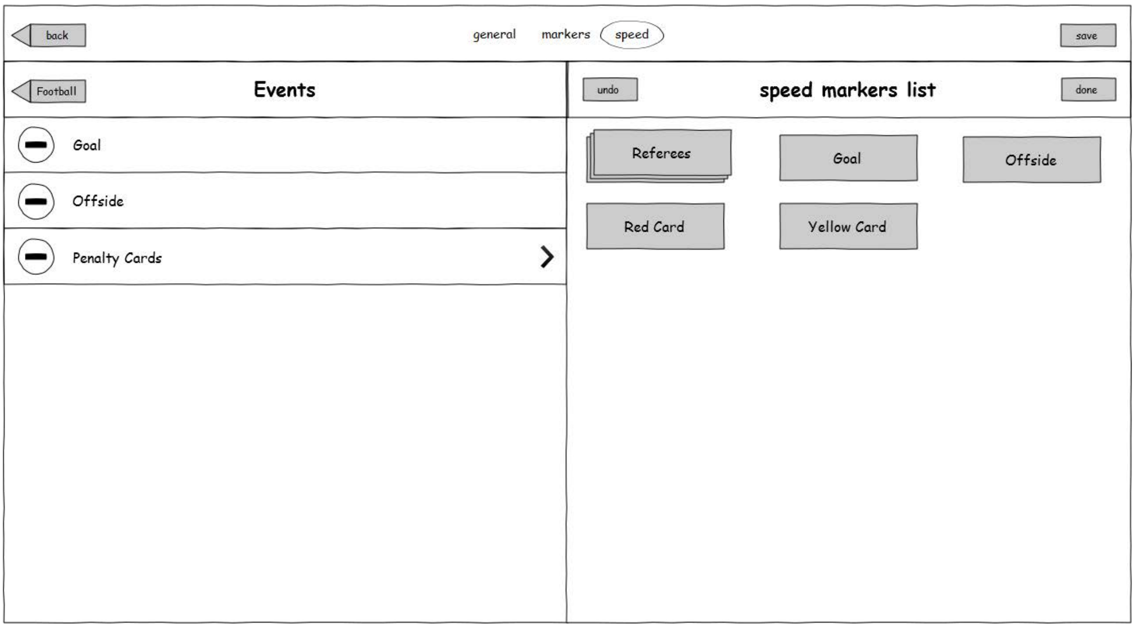


Figure 4.24: Mid-fidelity prototype of speed marker list builder after adding a collection of markers, one by one, before user testing

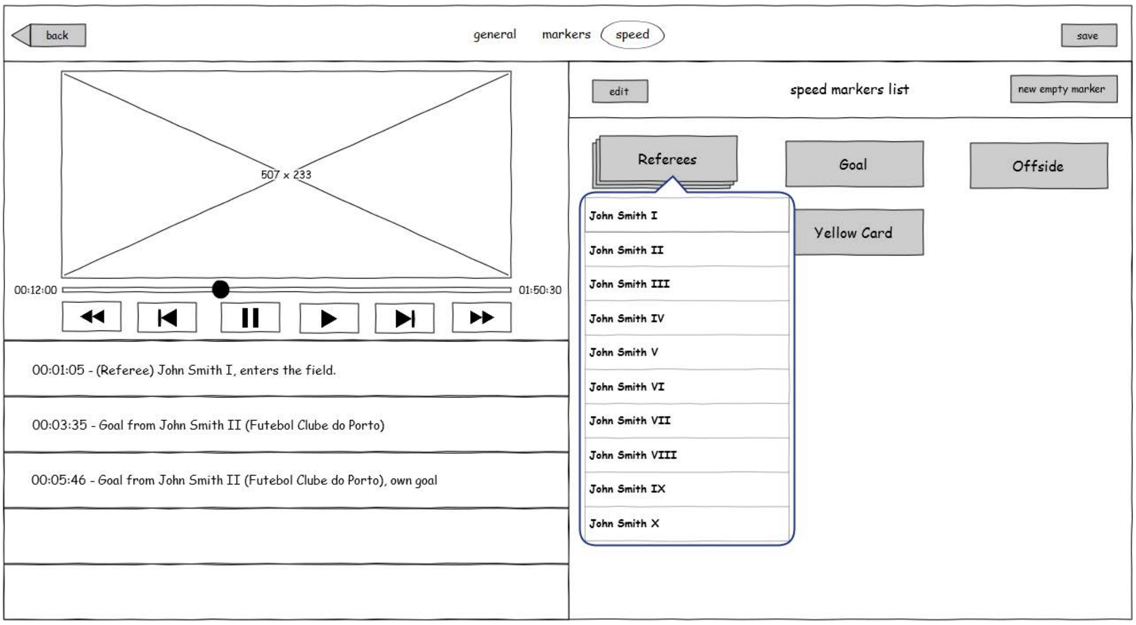


Figure 4.25: Mid-fidelity prototype of speed marker disclosure before user testing

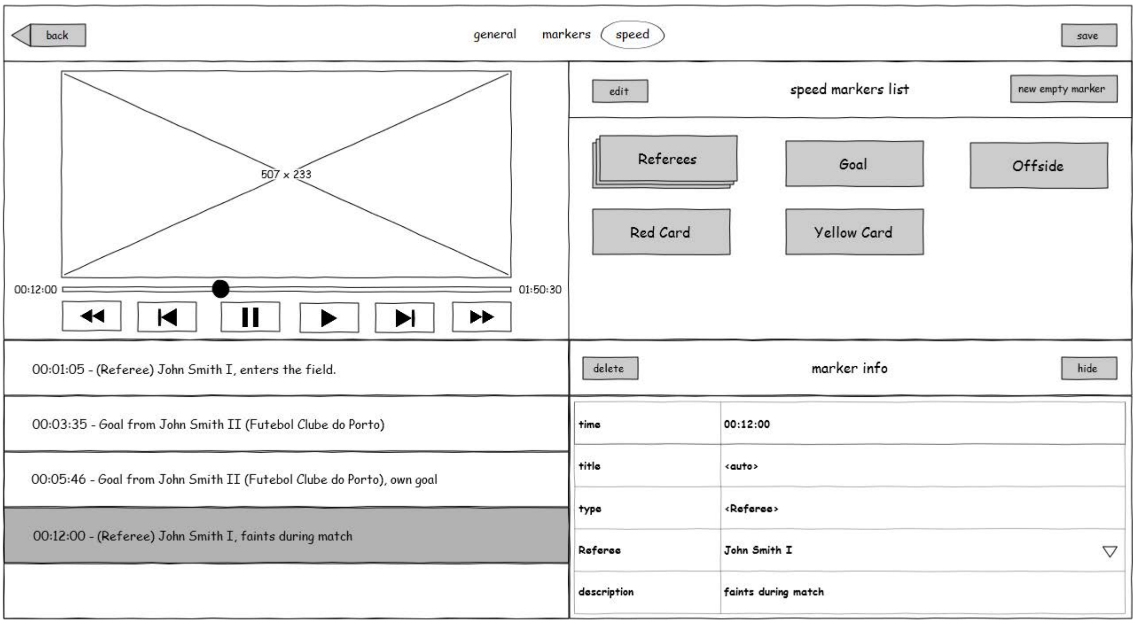


Figure 4.26: Mid-fidelity prototype of speed marker list before user testing

4.4.7.2 After user testing

After testing the previous mockup with RTP users - especially archivists since those would be its main users -, many issues came up and we redesigned our solution.

The initial screen was kept the same, but we incorporated the system-wide change of the static video left window. Since the left space cannot be used for list browsing, we divided the space on the right for list browsing on the bottom and its contents on the top. Selecting a different list in the bottom will cause the upper contents to change. On the left panel we included options for list editing, so as to select a list as a basis but still be able to add or remove elements.

List editing is just like creating a list but we added a new concept: folders. Folders work just like collections, but we can now manually create a folder (Figure 4.23) and add the desired contents inside it. Figure 4.24 shows the example of the football player "Danilo" being added to the folder "Porto Players", by first tapping the player and then tapping the folder. If the user tapped anything else but a folder, it would be added to the list as another speed marker outside any folder.

After the list is ready and we want to start the logging activity, tapping the "start marking" button will take us to the screen on Figure 4.31. On the previous set of mockups (Chapter 4.4.7.1) we were not clear how we would create the interlinking between concepts, such as a goal from a specific player. After discussing the idea with the users in the first iteration, we designed the following interaction.

Taking as an example the list previously edited and selected, Figure 4.31 shows us the screen that would appear when the system is ready to start logging. To create a new marker, the user just taps one of the speed markers or the "new empty marker" button on the bottom-left corner of the screen. By choosing "Goal", it is natural that we would want to immediately choose which player scored it. Tapping the "Goal" speed marker, a window pops up asking us who scored the goal. As we can see in Figure 4.32, there are several options one can choose. Firstly, there is a list of players automatically available. The players automatically added are the players that were selected to the speed marker list, inside or outside any folders. If the player was not added, we can still choose it from the full library or insert its name manually.

This creates a marker like the one we can see on Figure 4.33, where all the details are disclosed and can be modified. Pressing the "back" button of that window takes us to the full list of speed markers (Figure 4.34), where tapping one would again disclose its information.

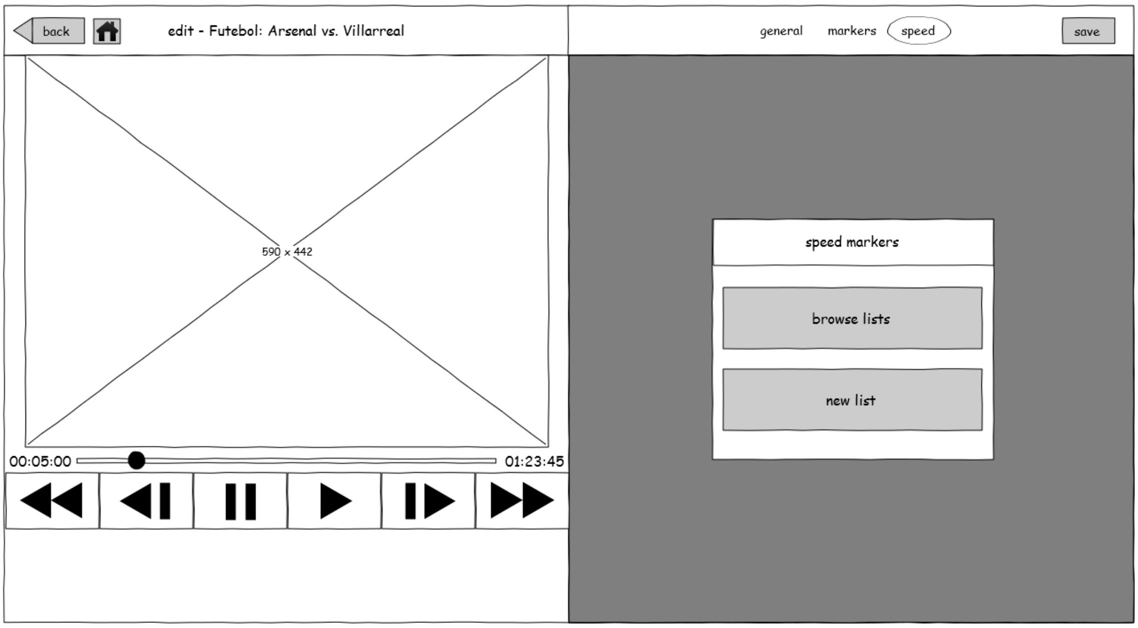


Figure 4.27: Mid-fidelity prototype of initial speed marking screen after user testing

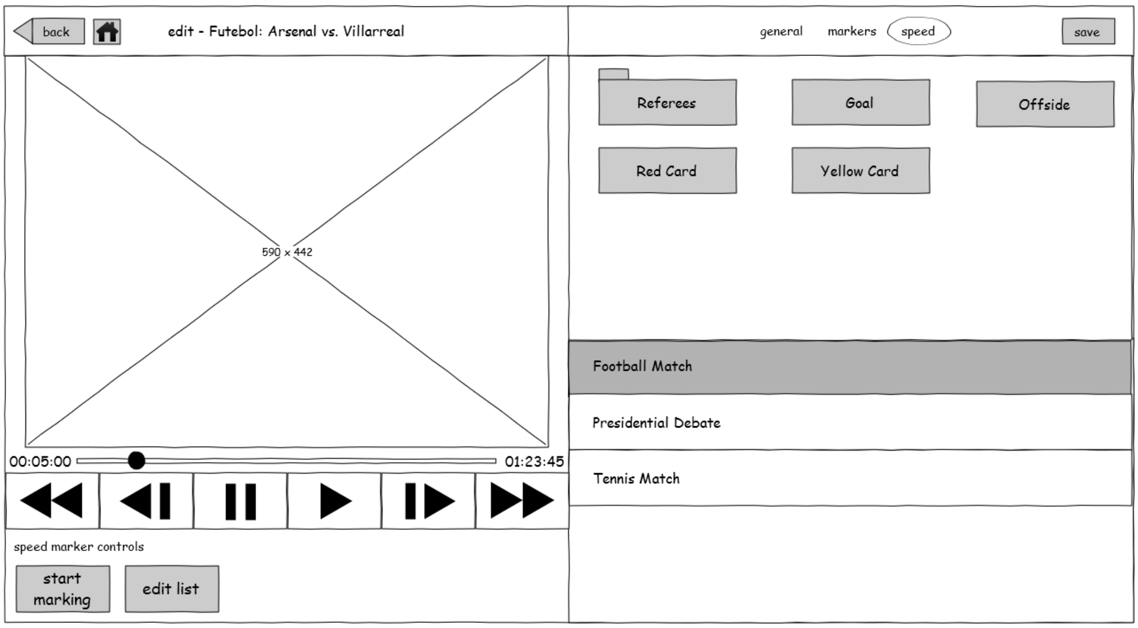


Figure 4.28: Mid-fidelity prototype of speed list browsing after user testing

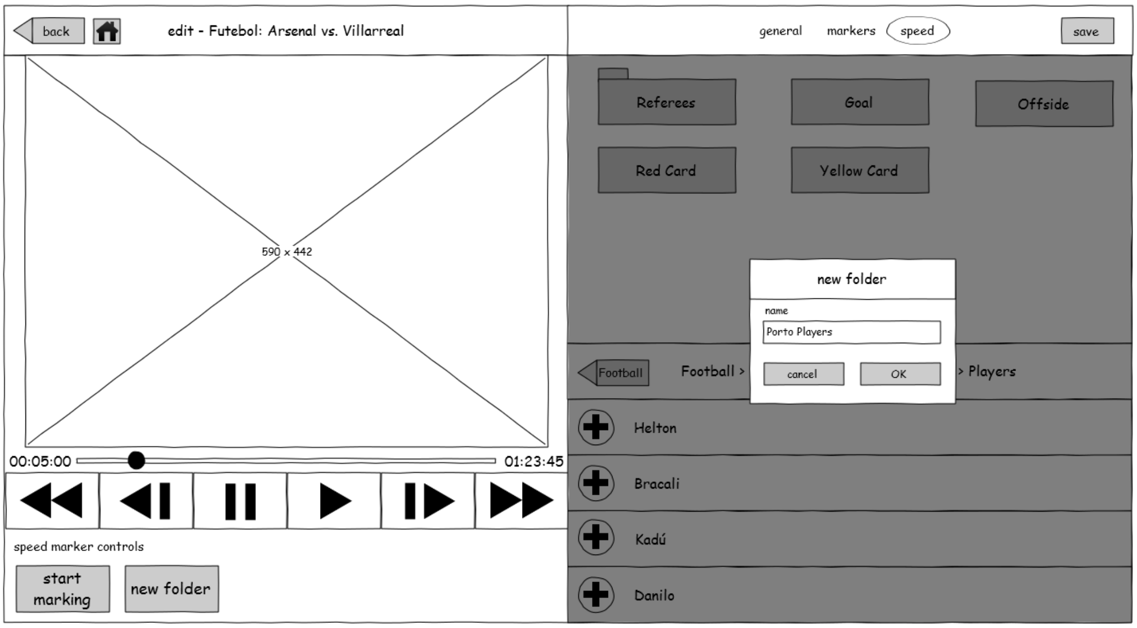


Figure 4.29: Mid-fidelity prototype of speed folder creation after user testing

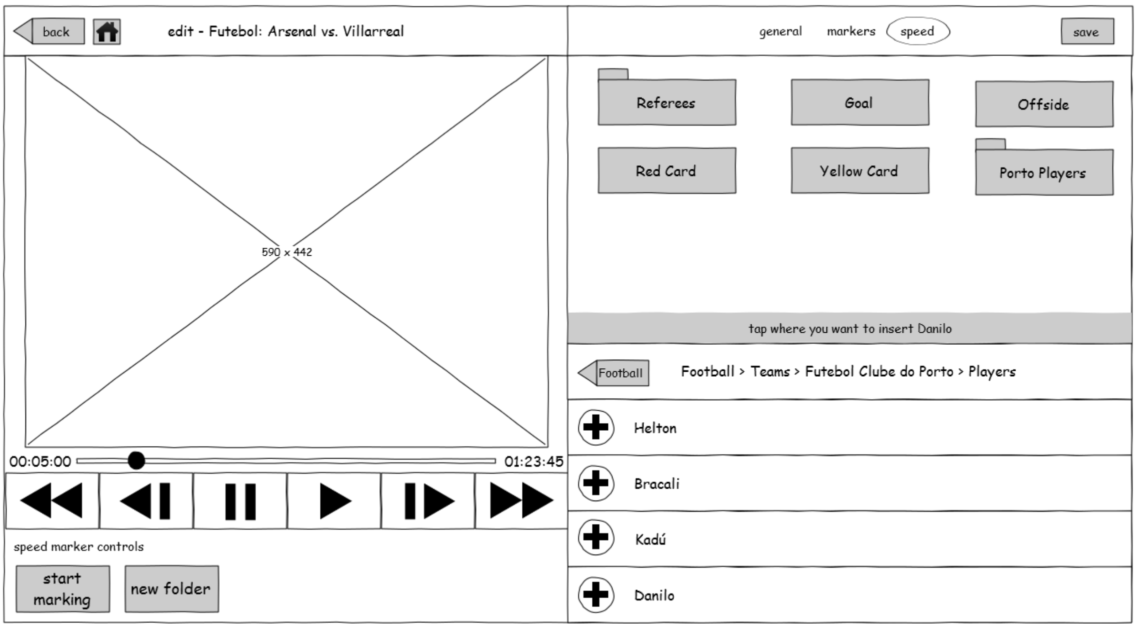


Figure 4.30: Mid-fidelity prototype of speed list creation after user testing

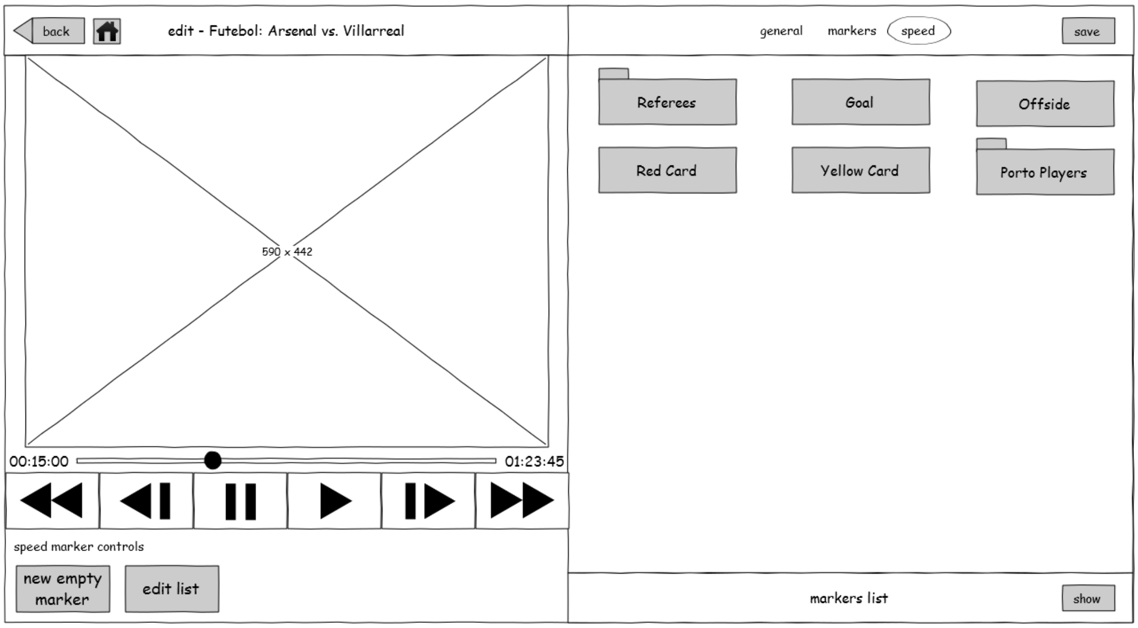


Figure 4.31: Mid-fidelity prototype of speed marker ready to start marking after user testing

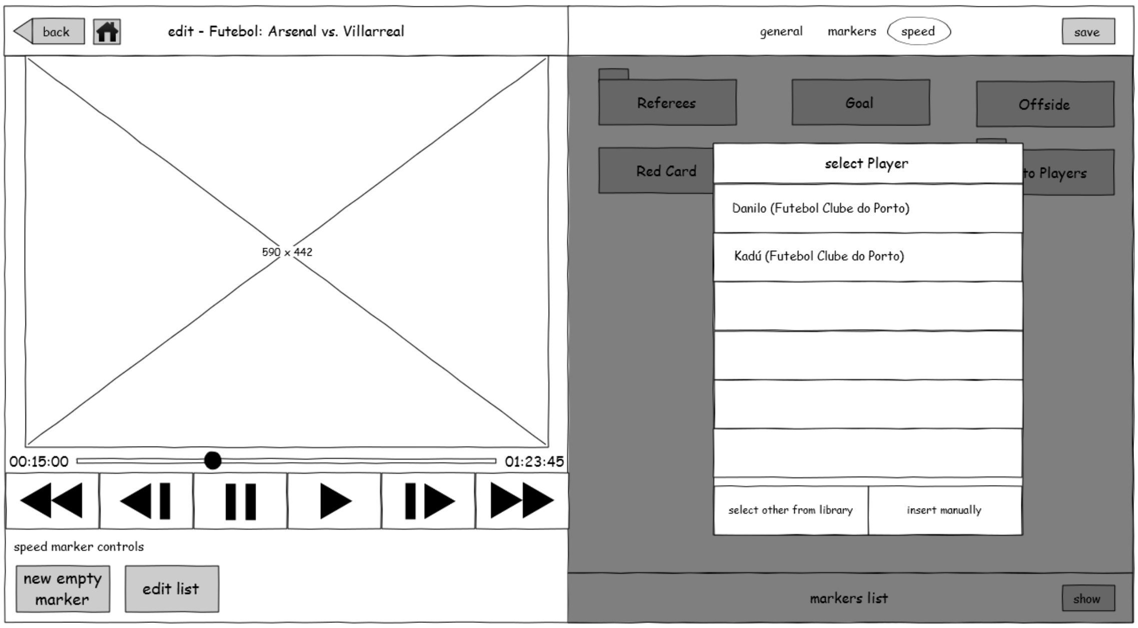


Figure 4.32: Mid-fidelity prototype of speed marker disclosure after user testing

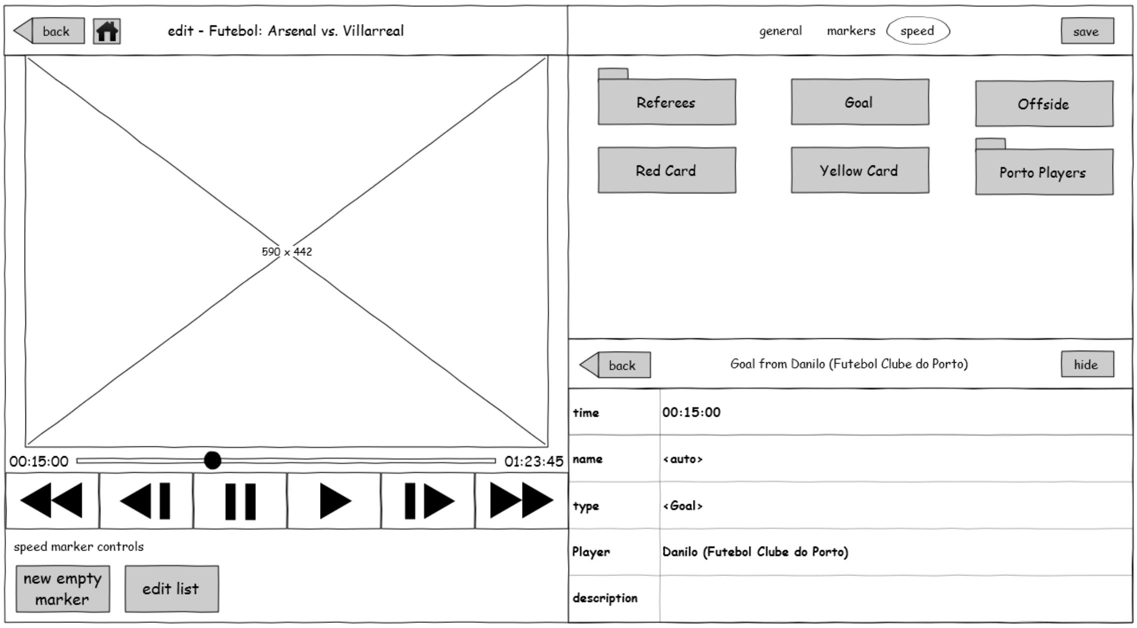


Figure 4.33: Mid-fidelity prototype of speed marker editing after user testing

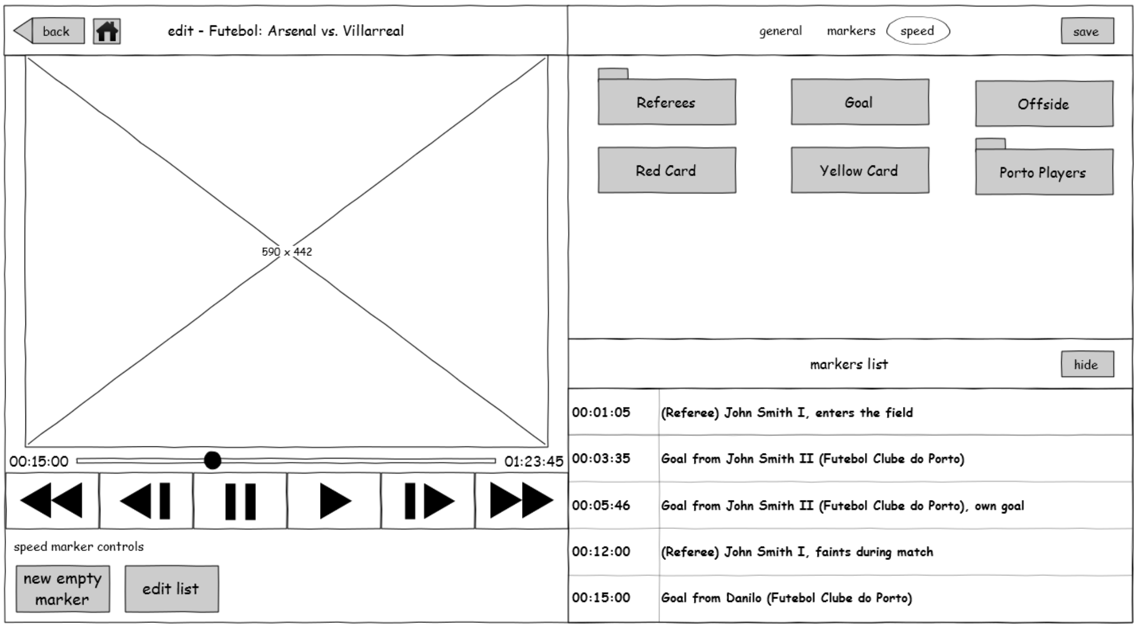


Figure 4.34: Mid-fidelity prototype of speed marker list after user testing

4.5 Conclusions

We have previously mentioned on Chapter 2.4 the advantages of using user-centered design methodologies and prototyping as a way to effective reduce costs by strongly involving users in the design process. This project and the success of the study we made so far was a proof of that theory. Although we still lack user testing in a hi-fidelity prototype or the real implementation, it is a fact that if the solution was developed as initially planned, it would fail, no matter how good its programmers were.

We only discussed the user evaluations from the first to the second medium-fidelity iterations of prototyping, but the second set of mockups was also tested and validated. Some small details about the design were pointed, but nothing that would justify another set of mockups, and we decided to directly implement them.

Chapter 5

Development

5.1 Introduction

Even though the aim of this project was not to fully develop a functional application rather than designing a way of accelerating the process of metadata logging, we were able to start developing a hi-fidelity prototype that could easily be evolved into a functional application through the development of some middleware to connect the prototype to other machines in order to work with real data.

In this chapter we will first explain some details from the development phase of this project, such as the architectural design of the application, as well as how the final solution and the hi-fidelity prototypes turned out to be.

We will conclude this chapter reporting some technical problems we ran into while developing this solution.

5.2 System architecture

We previously described very briefly the system's architecture, in section 4.2.2, when we discussed the requirement of being integrated with MOG's mxfSPEEDRAIL system.

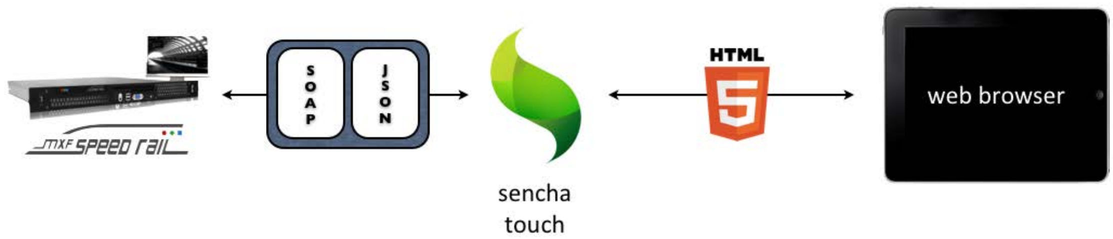


Figure 5.1: System architecture illustration

For the mobile development of our system's application, we used Sencha Touch, the HTML5 mobile framework. This framework works entirely in JavaScript and needs to be hosted in a server so that its code can be downloaded by the machine that wants to use it, accessing it via browser with the server's address.

But since the mxfSPEEDRAIL internal server will just be hosting our application's files that will be downloaded by the external machine that connects to it, we need to find a way of communicating between mxfSPEEDRAIL and our application. Fortunately, Sencha Touch implements JSON [JSO12] to communicate with external entities, offering many advantages in using this data-interchange language, such as already built functions for most communication needs.

In other hand, mxfSPEEDRAIL has a web-service interface based in SOAP that allows full interaction with the system. But since we have a JSON communicating system in one side and a SOAP communication system in the other, a middleware layer needs to be built in order to connect these two systems.

Because this task would require additional work (thus time) to be developed, we decided to leave it out of this project, since our available time was limited and the task was out of our main focus.

5.3 Software architecture

For the development of this application, we adopted a model-view-controller (MVC) architectural design, since separating these three application layers would allow:

- Easy conversion of the hi-fidelity prototype into a fully capable software through adaptation of the Model layer into allowing connection with mxfSPEEDRAIL by developping a middleware layer between JSON and SOAP.
- Easily changeable user interface by replacing the View layer, allowing adaptation and personalization for multiple devices.
- Reusable logic layer (Controller) allowing for other SOA based systems to communicate with ours.

Fortunately, Sencha Touch already implements many functions that help achieving such architecture by providing extensible JavaScript classes to seamless integrate models, views and controllers into an application.

5.4 Implementation

The implementation of this project was, as previously mentioned, with the unique goal of achieving a hi-fidelity prototype to be further developed into a real functional application. Therefore, even though there is no connection between our prototype and other systems, we try to simulate as much interaction as possible by implementing most of the user interface for testing purposes.

We will now present how the final version looks by showing some screenshots of the UI as well as differences from previous mockup versions, also based on user input from the last mid-fidelity prototype user testing.

Figure 5.2: Hi-fidelity prototype of login screen

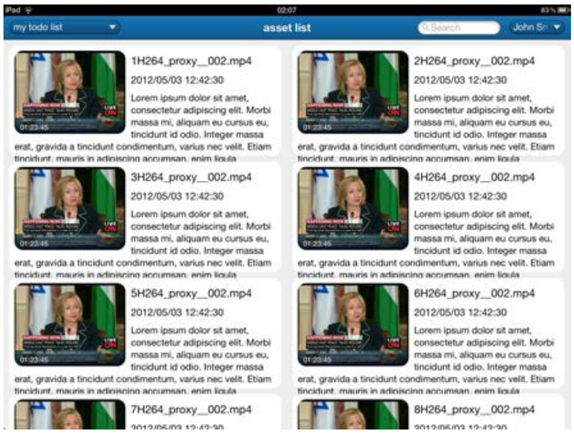


Figure 5.3: Hi-fidelity prototype of asset list

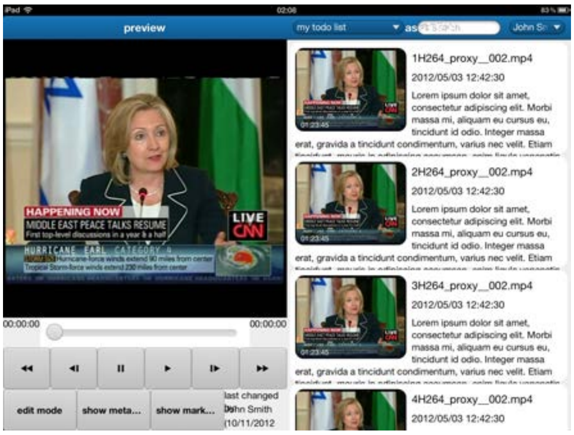


Figure 5.4: Hi-fidelity prototype of asset list filtering

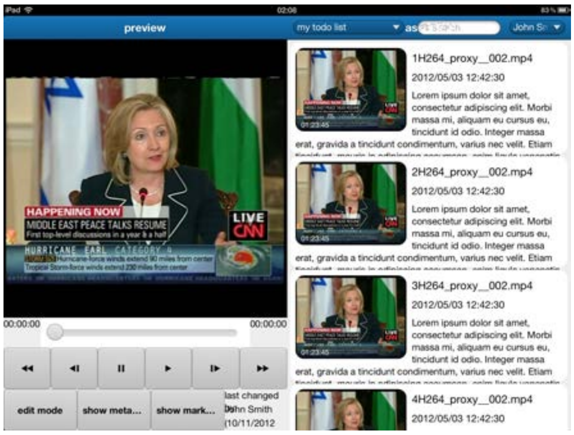


Figure 5.5: Hi-fidelity prototype of video preview

The login screen (Figure 5.2) as well as the asset list (Figure 5.3) were implemented just as initially designed. The only small change on the asset list was the disclosure of the options/logout menu by clicking on the users name in the top-right corner that was not obvious for any of the users. By using the framework's native drop down widget we made it more obvious into understanding that the name tag can be disclosed for further options. The ability to change a list filter, as illustrated in Figure 5.4, is also achieved in the same way but on the top-left corner of the screen.

By tapping a video on the asset list, as illustrated in Figure 5.5 the size of the asset list is halved and moved into the right part of the screen. On the left we can see the video as well as the buttons and controls that were initially designed.

Video-wide metadata (Figure 5.6) and markers (Figure 5.7) were also implemented as planned and tapping a marker will indeed make the video change its position to the marker's starting time.

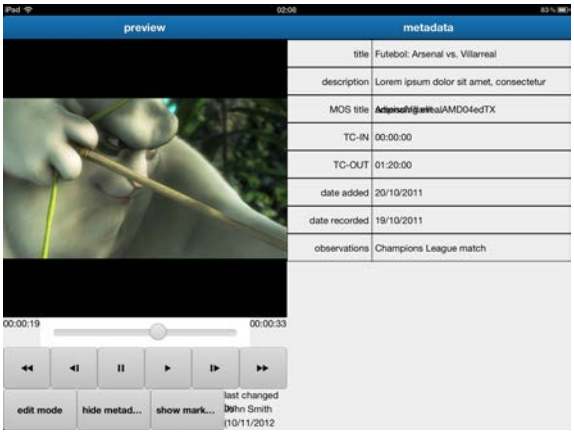


Figure 5.6: Hi-fidelity prototype of video preview showing metadata

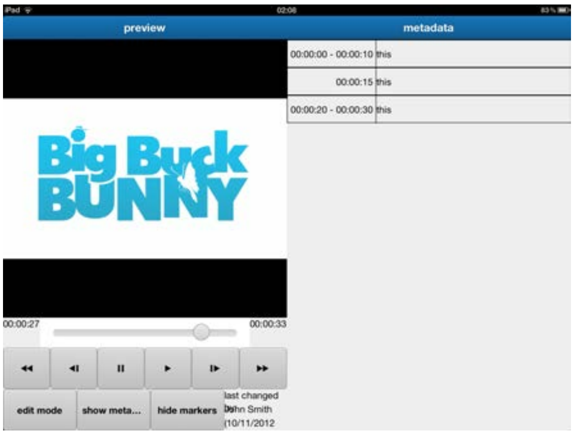


Figure 5.7: Hi-fidelity prototype of video preview showing markers

Entering the edit mode (Figure 5.8), we can see an implementation also just as planned, but adding a new metadata field is now more obvious. By tapping the "add new field" button at the end of the list, a tabbed window pops up on the right. In here we may choose from a set of starred fields or a list of all the fields stored in our system where we can star them in order to appear on the previous list (Figure 5.9). In addition, we may also create a completely new field by choosing its type (so we can improve the input method), name and value as well as some other options (Figure 5.10).

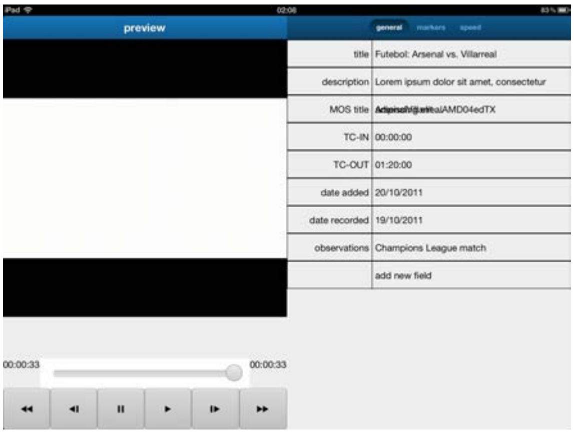


Figure 5.8: Hi-fidelity prototype of edit mode in general tab

Figure 5.9: Hi-fidelity prototype of edit mode adding a metadata field from the internal repository

We also had some minor improvements in the markers tab of editing mode. Users thought that the "in ... out" idea was a good idea but that between that and total manual edition in case of marker editing, there should be a way of editing a marker with the help of the video seek bar. Now, when we disclose a marker from the list, in panel that slides from the bottom of the screen there is a bar where we are able to set the time in and time out properties by tapping those buttons that will define the markers times according to the actual video time code. Also noticeable is how

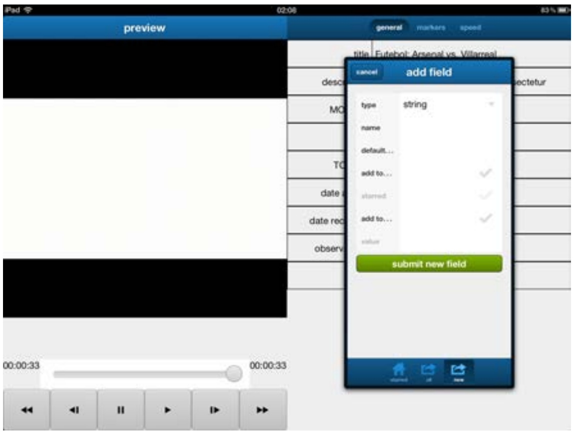


Figure 5.10: Hi-fidelity prototype of edit mode adding a new metadata field manually

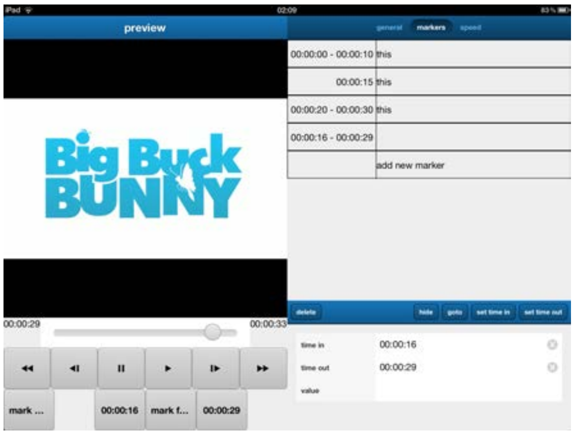


Figure 5.11: Hi-fidelity prototype of edit mode in markers tab

the delete button was packed to the left, the middle of the screen, in order to be less accessible, thus less prone to user mistakes by accidentally tapping.

5.4.1 Problems

During the implementation of this hi-fidelity prototype, we ran across multiple problems. In this section we will report the most important ones.

5.4.2 HTML5 <video> tag

The <video> tag [Con12b] was one of the most awaited tags of the new HTML version and probably the most popular one, since it allows video playback without the need of external plugins such as Adobe's Flash [Ado12] or other specific plugin capable of playing the video format the webmaster chose to include in his page from the immense list of video codecs available for use. Another great advantage in this standardization is that video playback on mobile-oriented websites no longer needs a specific workaround for each device's model, browser or operating system, since all of these plugins are platform-specific and most of them have very limited compatibility, therefore not able to reach the whole market.

But although this functionality is being put in use by many popular websites, like Youtube.com [You12], there are still many issues concerning the specification by W3C [Con12a] and its actual implementation.

From the problems we ran into, we selected two that limited our implementation and had no feasible solution in sight:

5.4.2.1 Rewind/fast-forward

According to the standard specification, there is an option in the <video> tag that allows speed playback to be changed. Its default value is 1.0. If, for example, we changed this value to 2.0 or 3.0, the video would play twice or triple as fast, respectively. Unfortunately, for reasons we could not find an explanation, this functionality was not implemented in any of the tablet computers we tested our prototype - Apple's iPad (3rd generation) and Asus Transformer Prime (Android 4.0).

Likewise, a value of -1.0 or -2.0 should mean that the video would play backwards at a normal speed or rewind double as fast. This functionality does not seem to be implemented in most browsers. From our experiments, the only successful implementation was Apple's Safari.

This limitation was unable to be fixed without changing the technology in use - HTML5 - or adding external plugins that still would make it unusable for most tablet computers. It was included in the prototype anyway, since we think it should be only a matter of time until browser developers implement this feature.

5.4.2.2 Frame-by-frame playback

Another feature according to the specification is the ability to change the video position by changing the current time property. In theory, calculating the amount of time per frame, we should be able to advance the video frame-by-frame. Unfortunately this is not true in every browser. Google Chrome for Microsoft Windows, for example, is perfectly able to play a video frame-by-frame, but Android's Google Chrome cannot. After much experiment, we found that the Android version would only advance between I-frames, a kind of frame specific from the video codec in use [Wik12a].

This was another problem that we also could not solve, since neither there is enough metadata available in the HTML DOM to locate the I-frames, nor there it is feasible to encode a video only with I-frames without considerably increasing its size. Since this is an application to use mostly on tablet computers, we decided to make the frame-by-frame buttons variable, so the user could decide in the settings how much should that button advance in seconds or if it should try to do frame-by-frame anyway.

5.4.2.3 Hardware acceleration for HTML rendering

Another problem that we found concerns the way the elements are rendered in the browser. While all Apple's tablets use GPU as well as CPU in order to render HTML elements and CSS3 animations, Android-based tablets rely on CPU for all those tasks.

While this could be seen as just a small issue, but the truth is that the difference is more than noticeable. While we can get almost real-time response using Apple's iPad (both 2nd and 3rd generation), Android's Asus Transformer Prime needs at least one or two seconds or more to process some UI changes ans skips most of the CSS3 animations. This is a problem when we consider the price of each device and compare their characteristics, since the iPad has a double-core CPU while the Android has a quad-core, but this optimization makes the first one much faster than the latter.

Chapter 6

Conclusions

Metadata logging is not a straightforward activity. On one hand, since there are activities such as archiving and archive search that are deeply dependent on the available metadata, metadata logging is an unavoidable task. But metadata logging is also a cost. This is a very time-consuming activity that currently demands dedicated people so it can be performed correctly.

Furthermore, the ambiguous nature of descriptive metadata and the unpredictability of knowing when an asset, or a specific information of an asset will be worth the cost of logging it, makes it very difficult to make business decisions and decide where time should be spent and where money should be saved.

With our system, two of these problems are solved. First, we distribute work around professionals. The work that was previously relying exclusively on archivists, is now distributed between reporters, journalists and archivists. Second, since it is not feasible to calculate what the return on investment of logging an asset will be, the best thing that can be done is reducing the amount of time the logging activity. We also achieve that reduction in many cases.

6.1 Accomplished goals

Most of our initially planned goals were accomplished successfully.

Our investigation phase was developed in the best possible conditions, since we had access to one of the main Portuguese TV stations and its facilities where we were able to observe, question and work with professionals of various fields. From there, we extracted both problems and solutions to them. We consider this was one of the most interesting parts from our work.

One of the main reasons this collaboration was very interesting was that without it, it would be impossible to clearly understand the methods professionals really use. This goal was achieved with a greater fidelity than only literature reviews would allow us to achieve.

The solution could also be designed and it even exceeded our initial expectations, since we ended up designing a solution for a specific company but that can be applied to a wider range of TV studios and media production companies.

As for the implementation, despite not having concluded our hi-fidelity prototype, we were happy with the result achieved for the time we had available to implement it. With a limited amount of time imposed for this dissertation, increasing the amount of time dedicated to implementation would mean decreasing the amount of time dedicated to our investigation, solution design and user involvement. This scenario was seen as scientifically inferior and therefore discarded.

6.2 Future work

Since this work was more about investigation than implementation, it is natural that there is future work ahead. Firstly, the hi-fidelity prototype needs to be finished in order to be further tested at the RTP studios. The user input will probably add one or more iterations to the development cycle until the final result is validated by the users. Only then should the middleware layer that connects JSON to SOAP be developed and fully tested with mxfSPEEDRAIL.

Some other ideas came up while developing this system and the next two chapters explain them.

6.2.1 Desktop application for modifying SpeedMarker library

Data entry on SpeedMarker's library is not a job to be performed on a tablet computer. Since it requires a lot of text input, a keyboard is still the best option available. Modelling how the entities relate to each other, like a goal needing a player, is also a job that ideally requires a lot of screen space to see all available entities. Probably a mouse would also be desirable in this situation, since touchscreens of a considerable size are too expensive and bring no advantages to this task.

For this reason, it was considered that a desktop application would be the ideal solution for this task. Ideally, it would export a file to be imported by the SpeedMarker application.

6.2.2 Allow offline marking

Our application has a limitation: metadata can only be added to assets after they are ingested. We never considered live-logging to be feasible because we always thought that the video stream would be needed on screen. Since sending a live video signal from a camera to a tablet would be almost impossible without hardware changes, we initially discarded this idea and assumed such issue to be a natural constraint due to technological limitations.

But then an idea came up of how that could be implemented in a simple manner. The main problem to be solved was how to synchronize a set of time-related metadata fields with a video timecode that would constantly be changed in an unpredictable manner, every time a camera operator pressed the recording/stop recording button of the camera. What wasn't realized was that there is a common time-related property between most professional cameras and any tablet computer

- the time of day property. If instead of saving a timecode we could save instances of the actual time, two fairly synchronized devices would be enough to sync the metadata to the timecodes of the assets.

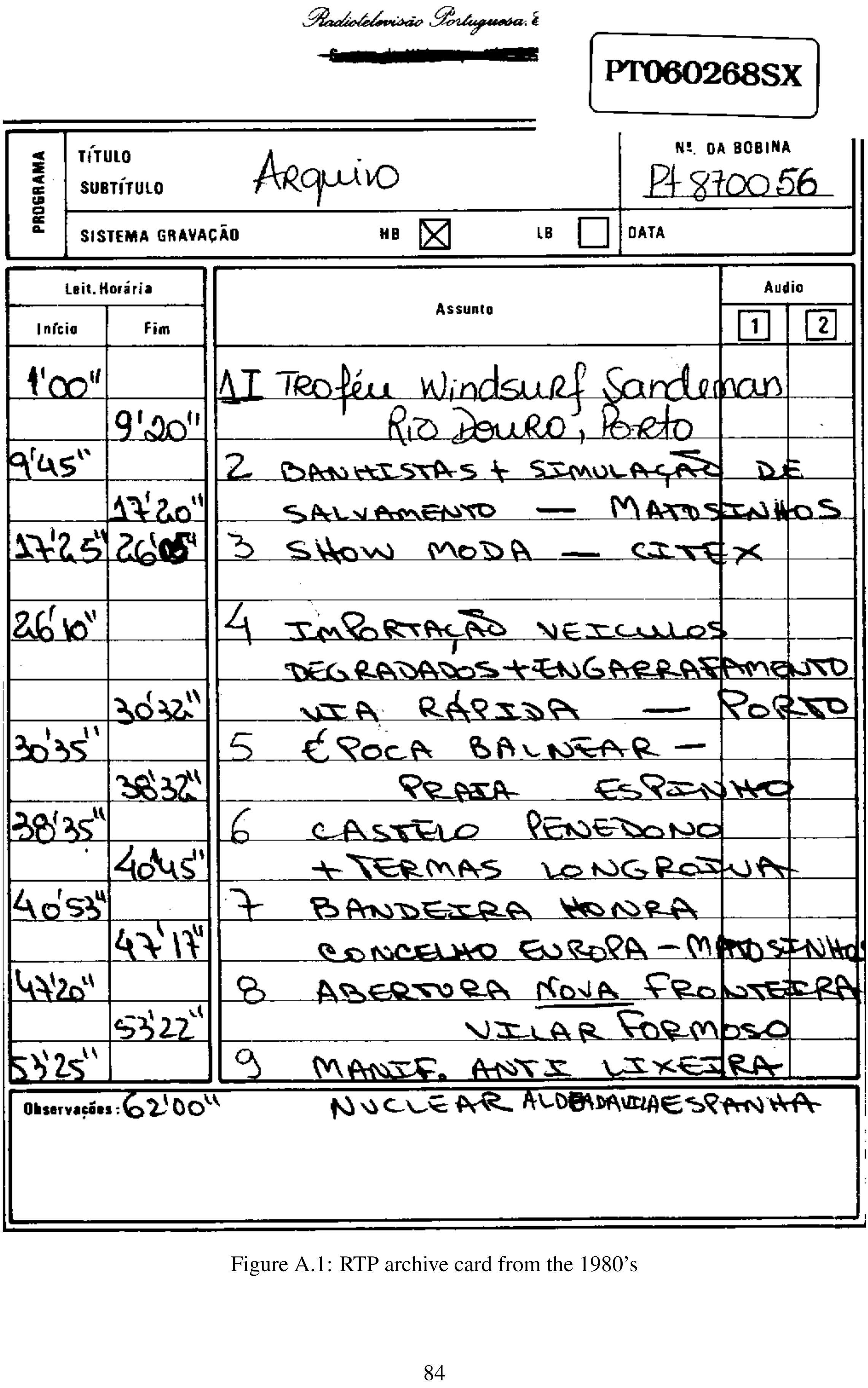
Fine tuning could easily be achieved with a kind of virtual clapper-board system, where we could insert a dummy metadata event in the application at the same time we start a dummy video in the camera. This would be enough the calculate the amount of sewing needed to sync the two devices. Manual tuning, if needed, would be a very simple task due to the sync proximity that the data would already have.

Appendix A

RTP archive entries

A.1 Old fully-manual archive cards

A.2 Snapshots from Blue Order entries



Radiotelevisão Portuguesa, E.P.
[redacted]

PT060268SX

PROGRAMA

TÍTULO / SUBTÍTULO: Arquivo

Nº. DA BOBINA: PT 870056

SISTEMA GRAVAÇÃO: HB ☒ LB ☐

DATA

Leit. Horária Início	Leit. Horária Fim	Assunto	Audio 1	Audio 2
1'00''		1 I Troféu Windsurf Sandeman		
	9'20''	Rio Douro, Porto		
9'45''		2 Banhistas + Simulação de		
	17'20''	Salvamento — Matosinhos		
17'25''	26'05''	3 Show Moda — Citex		
26'10''		4 Importação Veiculos		
		Degradados + Engarrafamento		
	30'32''	Via Rápida — Porto		
30'35''		5 Época Balnear —		
	38'32''	Praia Espinho		
38'35''		6 Castelo Penedono		
	40'45''	+ Termas Longroiva		
40'53''		7 Bandeira Honra		
	47'17''	Concelho Europa — Matosinhos		
47'20''		8 Abertura Nova Fronteira		
	53'22''	Vilar Formoso		
53'25''		9 Manif. Anti Lixeira		

Observações: 62'00'' Nuclear Aldeadávila Espanha

Figure A.1: RTP archive card from the 1980's

Clip objecto

Identificação

Nome do Ficheiro	GarciaHortaJTrte 05edTX	Título MOS	GarciaHortaJTrte 05edTX	Criado em	5/3/2012
Suporte de arquivo	LX120722XD	Suporte de ingest		Arquivado em	5/3/2012
TC-IN	00:00:00.00	TC-OUT	00:02:07.07	Duração	00:02:07.07
Arquivo Permanente	1	Informação diária	1	Originais	0

Contexto

Data	5/3/2012	Canal	RTP 1	Versão	FINAL LIMPO
Estado	Catalogado				

Conteúdo

Descrição	Hospital Garcia da Orta
Resumo Sintético	Almada, Hospital Garcia de Orta, em Almada, foi obrigado a pedir compressas emprestadas para não adiar cirurgias; fornecedor do hospital suspendeu a entrega do material por falta de pagamento, a dívida ao fabricante é de 260 mil euros.
Resumo Analítico	Bloco operatório com cirurgia a decorrer, hospital Garcia da Orta; declarações de Hélder Miranda, Diretor Comercial da Albino Dias de Andrade.

Indexação

Lista de termos

Idx	Termo	Infraconceito
1	BLOCO OPERATÓRIO	
2	CIRURGIA	
3	HOSPITAL GARCIA DA ORTA	EXTERIOR
4	HOSPITAL GARCIA DA ORTA	INTERIOR
5	ALMADA	
6	PORTUGAL	

Condições de acesso e de utilização

Direitos: Sem restrições

Fundamentação

Idx	Tipo	Observações

Características técnicas

Formato imagem	4:3 (Default)	Omnibus ID	\\ARQ_LX\Omn\D\S\63\31

Notas

Observações: Jornalista: Lavínia Leal

Última actualização

Alterado em	9/3/2012 09:13:40	Alterado por	WorkflowEngineWS

Figure A.2: RTP archive entry from Blue Order

Clip objecto

Identificação

Nome do Ficheiro	P012002422NSM	Título MOS		Criado em	19/2/2012
Suporte de arquivo	LX120316XD	Suporte de ingest	LX12000316XD	Arquivado em	19/2/2012
TC-IN	22:26:10.00	TC-OUT	23:52:40.00	Duração	01:26:30.00
Arquivo Permanente	0	Informação diária	0	Originais	0

Contexto

Data	14/1/2012	Canal	RTP Informação	Versão	FINAL LIMPO
Estado	Catalogado				

Conteúdo

Descrição: ZONA MISTA

Resumo Sintético: Programa de debate desportivo moderado pelo jornalista Hugo Gilberto, com os comentadores João Gobern e Bruno Prata, fazem a análise dos jogos de futebol da 15ª jornada do campeonato nacional da Liga Zon Sagres, Futebol Clube do Porto vs Rio Ave,(2/0), Sport Lisboa e Benfica vs Vitória de Setúbal (4/1) e o Sporting Clube de Portugal que prepara o jogo frente ao Sporting de Braga.

Resumo Analítico: 22h46m43: Estádio do Dragão, direto da conferência de imprensa de Carlos Brito, treinador do Rio Ave, após a derrota com o FCP. 22h50m02: Estádio do Dragão, direto da conferência de imprensa de Vítor Pereira, treinador do FCP. 23h03m17: Peça tratada com o título mos: ReaxLuzMhe 14EdTX. 23h21m45: Peça tratada com o título mos: SportingNovaINHco 14edTX. 23h23m48: Peça tratada com o título mos: BragaJTMsc 14edTX. 23h32m01: Peça tratada com o título mos: Couceiro 14edTX.

Indexação

Lista de termos

Idx	Termo	Infraconceito
1	PEREIRA, Vítor (1968-)	
2	RIO AVE FUTEBOL CLUBE	
3	BRITO, Carlos	
4	FUTEBOL CLUBE DO PORTO	
5	FUTEBOL	

Condições de acesso e de utilização

Direitos: Sem restrições

Fundamentação

Idx	Tipo	Observações

Características técnicas

Formato imagem	4:3 (Default)	Omnibus ID	\\ARQ_LX\Omn\D\R\74\54

Notas

Observações:

Última actualização

Alterado em	29/2/2012 15:59:51	Alterado por	Ludovina Conceicao Nunes Andrade

Figure A.3: RTP archive entry from Blue Order

Clip objecto

Identificação

Nome do Ficheiro	N004000072NSD	Título MOS		Criado em	4/7/2007
Suporte de arquivo	LX04009256XD	Suporte de ingest	LX04009256XD	Arquivado em	4/7/2007
TC-IN	00:32:48.10	TC-OUT	00:48:30.24	Duração	00:15:42.14
Arquivo Permanente	1	Informação diária	1	Originais	0

Contexto

Data	8/12/2004	Canal		Versão	EMISSÃO
Estado	Catalogado				

Conteúdo

Descrição: Futebol: Liga dos Campeões Roma vs Real Madrid (0-3) 2ª parte

Resumo Sintético: Itália, Estádio Olimpico de Roma 6ª jornada da Liga dos Campeões, resumo alargado do jogo entre a Roma, 0 e o Real Madrid, 3. O jogo decorreu à porta fechada sem público.

Resumo Analítico: Início da 2ª parte. 37m33: Grande penalidade sofrida por Ronaldo (Real Madrid), Golo de Luís Figo (Real Madrid) de penalty. 43m30: Livre directo marcado por Mancini (Roma) defesa de Casillas. 44m48: Golo de Corvia da Roma anulado pelo árbitro por fora de jogo; Plano de Luigi Del Neri. 46m50: Golo de Luís Figo (Real Madrid) 47m31: Substituição no Real Madrid saida de Luís Figo entrada de Pavón. 48m21: Fim do jogo.

Indexação

Lista de termos

Idx	Termo	Infraconceito
1	LIGA DOS CAMPEÕES	
2	FUTEBOL	
3	ROMA (CLUBE)	
4	REAL MADRID	
5	FIGO, Luís (19721104)	GOLO
6	ITÁLIA	

Condições de acesso e de utilização

Direitos: Sem restrições

Fundamentação

Idx	Tipo	Observações

Características técnicas

Formato imagem	4:3 (Default)	Omnibus ID	\\ARQ_LX\Omn\A\F\02\49

Notas

Observações:

Última actualização

Alterado em	14/2/2012 00:16:32	Alterado por	WorkflowEngineWS

Figure A.4: RTP archive entry from Blue Order

RTP archive entries

References

[Ado12] Adobe. Flash player 11, June 2012. http://www.adobe.com/products/flashplayer.html.

[Alv95] Harald Tveit Alvestrand. Tags for the identification of languages. Technical report, Internet Engineering Task Force, 1995.

[App12] Apple. ios, 06 2012.

[Arc] Society Of American Archivists. So you want to be an archivist: An overview of the archives profession. http://www2.archivists.org/profession.

[Avi12a] Avid. Avid interplay, 02 2012.

[Avi12b] Avid. Avid interplay mam 4 features, 06 2012.

[Con97] World Wide Web Consortium. Date and time formats. Technical report, World Wide Web Consortium, 1997.

[Con12a] World Wide Web Consortium. About w3c, June 2012. http://www.w3.org/Consortium/.

[Con12b] World Wide Web Consortium. Html5 working draft - the video element, March 2012. http://www.w3.org/TR/html5/the-video-element.html#video.

[Cor04] Dublin Core. Dublin core metadata element set, version 1.1: Reference description. Technical report, Dublin Core, 2004.

[Cor12] Dublin Core. Dublin core, 02 2012.

[Dev02] Bruce Devlin. Mxf - the material exchange format. Technical report, European Broadcasting Union, 2002.

[DWBT06] Bruce Devlin, Jim Wilkinson, Matt Beard, and Phil Tudor. *The MXF Book: An Introduction to the Material eXchange Format*. Focal Press, 2006.

[eTdP] Rádio e Televisão de Portugal. Notícias - vídeos. http://www.rtp.pt/noticias/index.php?tm=&headline=206&visual=60.

[Evo12] Evolus. Pencil, June 2012. http://pencil.evolus.vn/.

[Gar53] Eugene Garfield. Librarian versus documentalist. *Special Libraries*, 1953. http://www.garfield.library.upenn.edu/papers/librarianvsdocumentalisty1953.html.

[GM11] Jesus Garcia and Anthony De Moss. *Sencha Touch in Action*. Manning Publications Co., 2011.

[Goo12] Google. Android, 06 2012.

[HA83] Martyn Hammersley and Paul Atkinson. *Ethnography: Principles in Practice*. Routledge, 1983.

[Har95] Brooks Harris. Guide to edl management, 1995. http://www.edlmax.com/EdlMaxHelp/Edl/maxguide.html.

[JSO12] JSON. Javascript object notation, June 2012. http://www.json.org/.

[Mar04] José M. Martínez. Mpeg-7 overview, 10 2004.

[New12] GPS Business News. Android and ios: 82

[Nie94] Jakob Nielsen. Heuristic evaluation. In J. Nielsen and R.L. Mack, editors, *Usability Inspection Methods*. John Wiley and Sons, 1994.

[Pre] Associated Press. Enps. http://www.enps.com/.

[Pre12] Associated Press. Enps mos integration, June 2012. http://www.enps.com/pages/enps_integration/what_is_mos.

[PRS02] Jennifer Preece, Yvonne Rogers, and Helen Sharp. *Interaction Design: Beyond Human-Computer Interaction*. John Wiley and Sons, Inc., 2002.

[Qua] Quantel. sq cut. http://www.quantel.com/page.php?u=de7fa7245fa0e03b1d51ed71c0e1dbab.

[Sen] Sencha. Sencha touch. http://www.sencha.com/products/touch.

[Tec12a] MOG Technologies. Mog technologies, June 2012. http://www.mog-technologies.com.

[Tec12b] MOG Technologies. Mog technologies - mxfspeedrail, June 2012. http://www.mog-technologies.com/products.

[Tec12c] MOG Technologies. Mog technologies - mxfspeedrail f1000, June 2012. http://www.mog-technologies.com/products/file-based/product-overview/.

[Tec12d] MOG Technologies. Mog technologies - mxfspeedrail o1000, June 2012. http://www.mog-technologies.com/products/digitaldelivery/product-overview/.

[Tec12e] MOG Technologies. Mog technologies - mxfspeedrail p1000, June 2012. http://www.mog-technologies.com/products/realtimeplayback/product-overview/.

[Tec12f] MOG Technologies. Mog technologies - mxfspeedrail s1000, June 2012. http://www.mog-technologies.com/products/sdi-recorder/product-overview/.

[VS99] Stephen Viller and Ian Sommerville. Coherence: an approach to representing ethnographic analyses in systems design. *Human-Computer Interaction*, 1999.

[WC02] Howard D. Wactlar and Michael G. Christel. *Building a National Strategy for Digital Preservation: Issues in Digital Media Archiving*, chapter Digital Video Archives: Managing Through Metadata, pages 80–95. Council on Library and Information Resources Washington, D.C. and Library of Congress, April 2002.

[Web] Webkit. Webkit. http://www.webkit.org/.

[Wik] Wikipedia. Jornalist and reporter definition. http://en.wikipedia.org/wiki/Journalist.

[Wik12a] Wikipedia. Group of pictures, June 2012. http://en.wikipedia.org/wiki/Group_of_pictures.

[Wik12b] Wikipedia. Linear video editing, May 2012. http://en.wikipedia.org/wiki/Linear_video_editing.

[Wik12c] Wikipedia. Playout, May 2012. http://en.wikipedia.org/wiki/Playout.

[Wik12d] Wikipedia. Sencha touch, January 2012. http://en.wikipedia.org/wiki/Sencha_Touch#Offline_applications.

[Wik12e] Wikipedia. Videotape, May 2012. http://en.wikipedia.org/wiki/Videotape.

[Win97] Terry Winograd. From computing machinery to interaction design. In P. Denning and R. Metcalfe, editors, *Beyond Calculation: the Next Fifty Years of Computing*, pages 149–162. Springer-Verlag, 1997.

[You12] Youtube. Html5 video player, June 2012. http://www.youtube.com/html5.