

Hildeberto Mendonça Filho

Multi-purpose and Extensible Framework for Multimedia Content Description

October 2010

UCL - Université catholique de Louvain

Université catholique de Louvain
Ecole Polytechnique de Louvain
Laboratoire de Télécommunications et de Télédétection

Multi-purpose and Extensible Framework for Multimedia Content Description

Hildeberto Mendonça Filho

Thesis Presented for the Ph.D. degree in Applied Sciences

Jury

Professeur Benoit MACQ (UCL) (Promoteur)
Professeur Christophe DE VLEESCHOUWER (UCL) (Prsident)
Professeur Jean VANDERDONCKT (UCL)
Professeur Thierry DUTOIT (Facult Polytechnique de Mons, Belgique)
Professeur Philippe PALANQUE (Universit Paul Sabatier, France)

October 2010

To my love Kênia

Abstract

Multimedia content is very rich in terms of meanings and archiving systems have to be improved to consider such richness. This thesis defines a multi-purpose, extensible and scalable framework for multimedia archiving to empower the capacity of content description. It is compatible and embeddable in a pragmatic formal process to integrate different stages of multimedia processing. Its extensibility and multi-purpose design allows the application on conceptually distinct problems. These improvements consider a set of techniques to segment and annotate different kinds of media. To validate this framework, we have applied it in two distinct case studies. The first one presents a technique for segmentation and annotation applied on the description of user interfaces, whose generated meta-data may support several approaches for user assistance, observation and user interface evaluation. The second case study aims to propose a multimodal fusion framework for high-level data fusion between two or more modalities. It takes as input media streams from different system devices, analyzes and identifies intrinsic meanings in these data. The fusion combines modalities to capture a context and enhance the user experience. Both experiments show the robustness of the framework to deal with different and complex problems.

Acknowledgements

Some years ago, I have defined this thesis as a major goal in my education, confident about its constructive impact on my carrier. Today, I'm convinced that I could not achieve this goal without the help and influence of many people who pushed me to persist and never give up this milestone. My sincere gratitude to all those special people who have help me directly and indirectly to make it real.

First of all, I would like to thank my parents, Hildeberto Mendonça and Terezinha Menezes, who heavily invested on my education, since I was 3 years old in a small town, Juazeiro do Norte, in the country side of one of the poorest states of the federation, Cear, Brazil. In a country where quality education is accessible, in the 21st century, only for a small portion of the population, they had spent 15 years paying for expensive schools and for the university without asking anything in return. Just moved by love. Still at my childhood's home, I thank my sisters Valeska Mendonça and Helaine Mendonça for helping me with my homework while my parents were very busy administrating a hotel and a restaurant, and Nitalma Mendonça for hosting me during my college's years in Fortaleza and managing me during my teenage years.

After becoming adult, I have met Kenia Sousa at University of Fortaleza. She have helped me a lot during the computer science course and we got married right after getting the undergraduate degree. We have agreed to live abroad and face new challenges, going to Belgium for graduate studies at Université catholique de Louvain (UCL). Since the beginning we have been a great team, helping each other to achieve the maximum of ourselves. I would like to thank her for the long hours she spent reading and correcting the spelling and grammar of this thesis and for her understanding when we were not able to be together while my thesis was consuming my attention.

Already in Belgium, I'm really thankful for the invitation of Jean Vanderdonckt to start my studies at UCL. He has guided my studies during my

first two years and we have discussed about so many great ideas at his office during long scientific talks. It has been a great pleasure working with the faculty, staff, and students at UCL, during my journey as a doctoral student. This work would never have been possible if it were not for the freedom I was given to pursue my own research interests, thanks in large part to the kindness and considerable mentoring provided by Benoit Macq. He has sponsored me during this whole journey in Belgium, allocating me in valuable scientific projects from where I got all the inspiration and resources to develop this thesis.

I'd also like to thank all of the people at TELE (Laboratoire de Tlcommunications et Tldtection) who have made it such a great place to work. Particular thanks go to Lionel Lawson for his support and collaboration with the OpenInterface Platform, Olga Vybornova for the fruitful collaboration we had on the multimodal fusion research, Christian Van Brussels for his understanding when I was supposed to contribute for his project, but I was busy writing the thesis, and Vincent Nicolas for his help on the improvement of this thesis in the last days. Thanks also to Jean Deschuyter and Patricia Focant for their permanent and unconditional support with normal bureaucracy at work.

Many thanks to the members of my jury, specially to Christophe De Vleeschouwer, who closely followed my research in TELE as member of the follow-up committee; Thierry Dutoit, who had the courage to create eNTERFACE, which is a real workshop, where part of this thesis was developed; and Philippe Palanque for his collaboration, motivation and support during my journey as researcher. Thanks also for all questions and reflections that helped me to improve this work.

I must not forget my dear sister-in-law Natalia Soares, who has incredible art talent and did a great job conceiving and designing Yasmim's logo. She also has motivated me together with her parents, Isabel Soares and Kerginaldo Sousa Jr., to go forward in this achievement. Isabel and Kerginaldo also gave me my lovely and loved wife. Thank you both so much!

At last, but not least, I would like to thank the Wallon Region, which has funded those two research projects that I was directly involved; all partners and members of the IRMA project, for their constructive feedback that contributed to improve Yasmim; and 3DMedia for helping to extend the framework to consider 3D contents.

Contents

1	Introduction	1
1.1	Describing Media	2
1.2	Research Motivation	5
1.3	Thesis	6
1.3.1	Thesis Statement	7
1.4	Scope of The Research	7
1.4.1	Target Audience	8
1.5	Reading Map	8
2	State of The Art	11
2.1	Introduction	11
2.2	Requirements	13
2.3	Standards	14
2.4	Multimedia Archiving Systems	15
2.5	Support for Segmentation and Annotation	19
2.6	Multi-Domain	23
2.7	Comparative Analysis	24
2.8	Conclusion	25
3	Multimedia Archiving Framework	27
3.1	Introduction	27
3.2	Application Modules	28

3.3	Access Management	29
3.4	Resources	30
3.4.1	Image	32
3.4.2	Audio	33
3.4.3	Video And Animation	34
3.4.4	Tridimensional	35
3.5	Segmentation	35
3.5.1	Spatial	36
3.5.2	Temporal	37
3.5.3	Spatio-Temporal	37
3.5.4	Segmentation Model	38
3.6	Link Between Segments	39
3.7	Annotation of Segments	41
3.7.1	Property	42
3.7.2	Tagging	43
3.7.3	Transcription	43
3.7.4	Description	43
3.7.5	AdHoc	43
3.7.6	Domain Concepts	44
3.8	Compliance with MPEG-7 and MPEG-21	46
3.9	Multimedia Archiving Workflow	48
3.10	Implementation	53
3.11	Client Implementations	56
3.11.1	Reference Client Implementation	57
3.11.2	IRMA Project Clients	57
3.11.3	MedicalStudio	61
3.11.4	Multimodal Clients	63
3.12	Conclusion	64
4	Segmentation and Annotation of User Interfaces	67
4.1	Introduction	67
4.2	User Feedback Steps	69

4.3	The Abstract User Interface Model	71
4.4	Why a Model-Driven Approach	74
4.5	Applying the Multimedia Archiving Framework	77
4.6	Links Between UI Components	80
4.7	Conclusion	82
5	High Level Multimodal Fusion	85
5.1	Introduction	85
5.2	Fusion Essential Requirements	86
5.3	System Design	88
5.4	Modality Processing Pipeline	88
5.4.1	Recognition	89
5.4.2	Segmentation	89
5.4.3	Meanings Extraction	90
5.4.4	Annotation	90
5.5	Autonomous Fusion Agents	91
5.6	User Interaction	92
5.7	Case Study	93
5.8	Speech Analysis	94
5.8.1	Speech Data	95
5.8.2	Speech Recognition	95
5.8.3	Speaker Identification	97
5.8.4	System integration of the speech components	98
5.8.5	Syntactic and Semantic Analysis	98
5.9	Video Scene Analysis	99
5.10	Ontology Design and Reasoning	101
5.11	Results	102
5.12	Conclusion	104
6	Conclusion	107
6.1	Summary of Contributions	107
6.2	Research Impact	109
6.3	Future Works	110

6.3.1 Short Term	110
6.3.2 Long Term	112
References	113
List of Publications	119
Access to Yasmim's Source Code	123
B.1 Requirements	123
B.2 Getting a Working Copy of the Project	124
B.2.1 Adding New Resources	124
B.2.2 Committing	124
B.3 Project Structure	125
Yasmim Deployment	127
C.1 Requirements	127
C.2 Creating the Database	128
C.3 Configuring Connection Pool	129
C.4 Configuring Authentication Realm	130
C.4.1 JDBC Realm	130
C.4.2 Application Configuration	133
C.4.3 Implementation of the Authentication Page	135
C.5 Configuring JavaMail™	136
C.6 Deploying the Application	138

List of Figures

1.1	LabelMe Website	3
1.2	Indirect indexation example in a blog post	4
1.3	Ambiguity in retrieved media in Google Images	5
2.1	MMDB audio recognition and blob-based image retrieval	17
2.2	SemaPlorer application showing street art in Berlin	18
2.3	IBM Multimedia Analysis and Retrieval System	20
2.4	Content segmentation and annotation	21
2.5	Process of segmentation and annotation	22
2.6	Image annotation tool	23
2.7	M-OntoMat-Annotizer user interface	24
2.8	Comparison of systems according to requirements	25
3.1	Component Diagram showing dependency between modules	29
3.2	Resource Model	31
3.3	Segmentation Model	39
3.4	Links represent the relationship between segments	40
3.5	Annotating using (a) cooking and (b) kitchen ontologies	45
3.6	Annotation Model	45
3.7	MPEG-7 main elements	48
3.8	Multimedia archiving workflow	51
3.9	Multimedia archiving workflow	52

3.10	Multimedia archiving workflow	52
3.11	Yasmim's administration user interface	54
3.12	Architecture	55
3.13	Architecture	55
3.14	Yasmim reference client implementation	57
3.15	IRMA query interface	58
3.16	IRMA retrieval interface	60
3.17	IRMA annotation interface	60
3.18	Speech Transcription Tool	61
3.19	MedicalStudio running components for mammography	62
3.20	Comparison of approaches considering Yasmim	65
4.1	Screen shot of a report annotated by the end-user	69
4.2	Screen shot of a form annotated by the end-user	70
4.3	Segmentation and annotation of a web form UI components	73
4.4	Abstraction levels of the Cameleon Reference Framework	77
4.5	Impact Analysis Visualization	79
4.6	Segmentation and Annotation of Screen shots	79
4.7	Floating component	82
4.8	Elastic component	83
5.1	The fusion system life cycle	91
5.2	A screenshot of the fusion system	92
5.3	The OI pipeline of the system	93
5.4	The system architecture	94
5.5	Results of the computer vision system	104
A.1	Reference Map	119
C.1	Database model of users and groups	131

List of Tables

3.1	Compatibility between segments and media types	35
3.2	Classification of supported annotations	42
4.1	Decomposition of User Interfaces	74
4.2	Representation and definition for Task Relationships	81
5.1	Fusion Mechanism Components	93
5.2	Word accuracy and correctness	97
5.3	Concept accuracy and correctness	97
5.4	Speaker identification accuracy	98

Acronyms

ASF	Advanced Systems Format
AVI	Audio Visual Interleave
BMP	Bitmap
DRM	Digital Right Management
GIF	Graphics Interchange Format
HCI	Human-Computer Interaction
IT	Information Technology
JDK	Java TM Development Kit
JPA	Java TM Persistence API
JPEG	Joint Photographic Experts Group
JSF	Java TM Server Faces
JSP	Java TM Server Pages
JRE	Java TM Runtime Environment
JVM	Java TM Virtual Machine
MPEG	Motion Picture Expert Group
OMG	Object Management Group
PNG	Portable Network Graphics
PPI	Pixels Per Inch
REST	Representational State Transfer
UI	User Interface
UML	Unified Modeling Language
URI	Uniform Resource Identifier

USI4BIZ Usability for Business

YASMIM Yet Another System for Multimedia Information Management

WMA Windows Media Audio

WMV Windows Media Video

Chapter 1

Introduction

The Internet became a widely accessible network with a distributed infrastructure that allowed it to scale exponentially. This scalability has enabled the availability of several on-line services that produced substantial impact on people's daily lives. A study made by IDC [36] shows that 487 billions of gigabytes were the amount of data added to the digital universe in 2008. By 2012, this number will be 5 times bigger. 70% of this number was produced by individuals. In 2007, the amount of digital information created in a year surpassed, for the first time, the amount of storage to deal with it. A lot of data were simply discarded.

It is, in part, due to the fact that content providers, strongly supported by service providers, have driven a wave of creativity all over the web, captivating other people's interest and needs. Social networks have influenced this process, by giving visibility to users' content, mainly for those with strong links with friends and followers. These are seen as propellants of content from those they consider reliable or attractive sources. For them, an important factor of attractiveness is the way and the format in which contents are conceived. At this point, multimedia contents play a relevant role on the general understanding of the message by enhancing people's expressiveness. Another important factor is the affordable price of broadband services, empowering the access and availability of large size contents, such as videos on demand, live shows video streams, music, podcasts, live radio streams, graphics, photos, vectorial 2D and 3D rendering, and so on.

On top of all the technology available to provide such a variety of media, there is the content produced by people all over the web. This content is very rich in terms of meaning, subjectivity, expressiveness, and other influences. Some messages are understood by everyone while others are understood by target groups. The production does not stop and an avalanche of new contents are published every day on web services, like youtube.com, vimeo.com, flickr.com and several other press websites, such as nytimes.com, lemonde.fr, globo.com and so on. Actually, media companies have been dealing with large

amount of data for years, and the Internet just made it a global challenge and everyone's concern.

The same revolution is also happening inside organizations, where multimedia has been applied to solve several real world problems. In hospitals, imaging technology has been used for diagnosis, follow-up of treatments, record of procedures, patient entertainment and so on. Industries have used images to monitor the quality of products, subway and train systems have successfully applied surveillance systems to detect unexpected situations, such as panic, emergency, fire, breaks, medical emergencies, etc. Innovative startups have created multimedia systems for mobile devices that analyze pictures and give information about the content that could be touristic information, price and specification of products, nutrition facts of foods, and many other creative solutions.

1.1 Describing Media

To go beyond the traditional media production and consumption, media contents have also to be described in a lower and higher level, allowing more efficient indexation, organization, distribution, summarization, and other benefits. At the same time, the richness of media contents and the complexity to process this kind of data are factors that hampers a precise content description. Media description is mostly based on surrounding text, some meta-data that can be easily extracted (such as date, size, resolution, etc), automatic detection of text and objects, tracking of objects in movement, and so on. However, these data usually drive to ambiguity, lack of representation, incomplete description, and so on.

While solutions on automatic analysis of media contents are evolving towards precision and autonomy, other initiatives have considered the work of humans on the description of contents, which seems to be a clever way to achieve good results when organizing and localizing media, due to the use of high cognitive capabilities, natural for human beings. Some critics would say that any manual process would cost too high and take too long to be done, but it is not completely true on the age of Internet where plenty of people are willing to contribute all over the world in pro of a cause. Even a business model can be built on top of this new trend, as we can see in Amazon Mechanical Turk¹.

To verify this trend and to improve recognition techniques, the MediaLab at MIT has created the project LabelMe [90], illustrated in Figure 1.1. It has collected 80 millions of tiny images on the web, using a web crawler,

¹ <http://www.mturk.com>

and requested the contribution of people through the Internet to manually segment and annotate objects in those images. Then, they could use this dataset to train the system and improve object detection and recognition research. Doing so, they could decrease in 15% the time needed to recognize objects in images and extend the recognition to multiple application domains, as an unprecedented contribution.

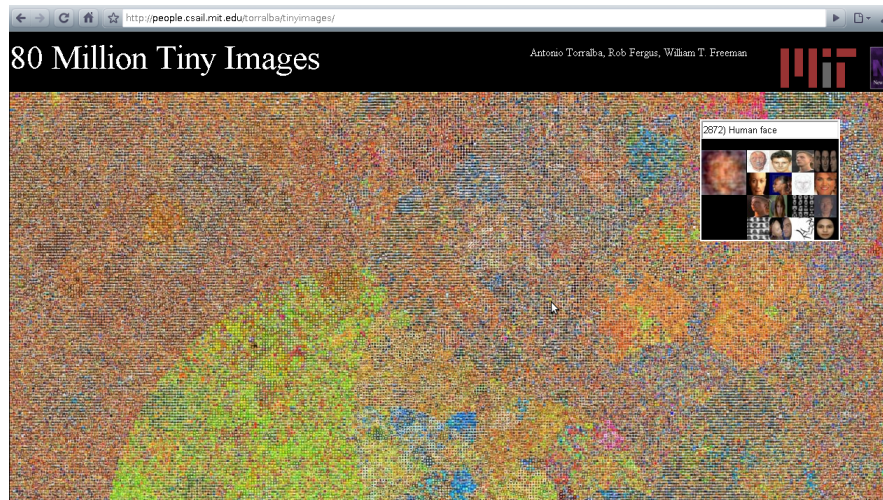


Fig. 1.1 LabelMe Website: <http://labelme.csail.mit.edu>

This experiment shows the relevance of describing media nowadays and how it can be achieved in a large scale, even if it demands a global effort while specialists keep evolving the precision of automatic initiatives. The success on multimedia description research would address problems where users do not describe the media content intentionally and search engines should do what they can to index the media considering other data sources, such as the name of the file, the text around it, the link, etc. Eventually, these data may not necessarily describe the media as expected. On the contrary, it might describe something that is perfect intelligible for humans, but totally confusing for algorithms. For example, the user is looking for figures on Google Images² to illustrate the term “creativity blockers”. The first media retrieved is a cartoon telling an entertaining fact that can happen in the office, as depicted in Figure 1.2, but it is not clearly related to the search term. In fact, this term is more related to the text around the image than to the image in itself. The author intended to illustrate the text with an image that connotes a scenario where some kind of blocking occurs, but not necessarily about creativity.

² <http://www.google.com/images>

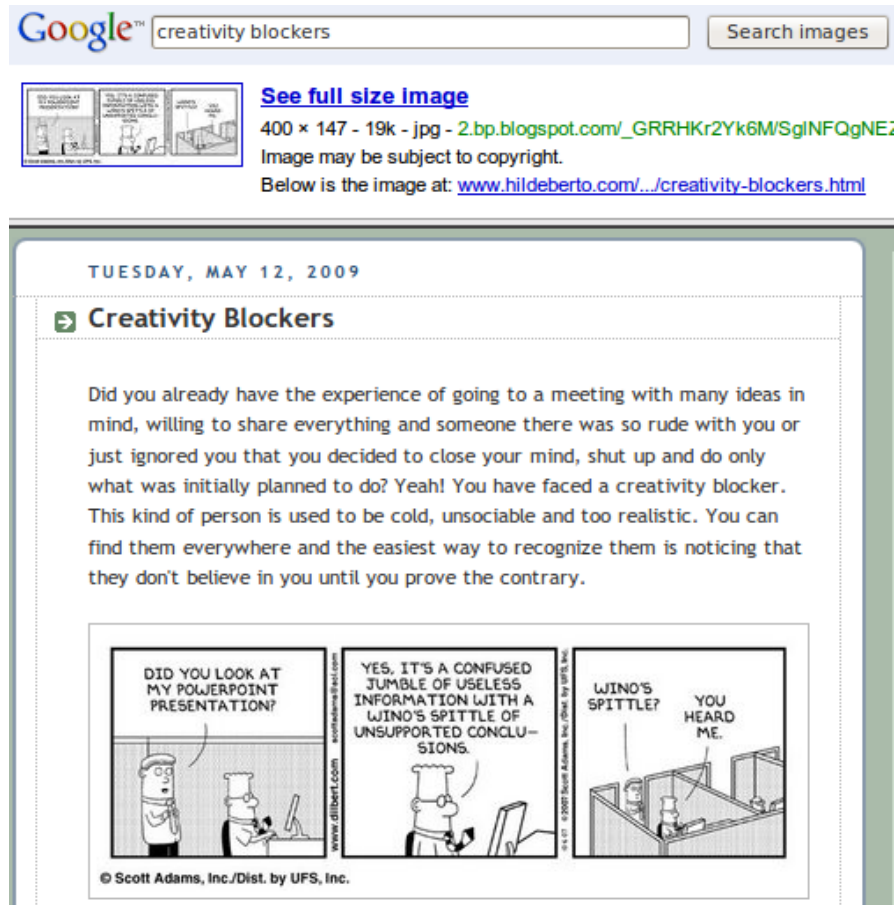


Fig. 1.2 Indirect indexation example in a blog post, from [64]

Another example is shown in Figure 1.3. It illustrates a typical case of ambiguity. A weather specialist intended to search for the term “cloud research” in order to find illustrations that better represent concepts of the literature on clouds, such as dimensions, altitude, density, atmosphere, etc. For his/her surprise, the retrieved search results presented several images related to information technology (IT), not purely about the atmospheric phenomenon. He did not know that IT specialists were using the term “cloud” as a metaphor for highly scalable computing environments. This ambiguity problem would be addressed by semantic annotations considering the domain of knowledge, which may help developers to present search results in a categorized way, grouping results by domain in case of ambiguity. It is a concerning problem for multimedia contents because they are essentially binary data and

meanings represented by 0s and 1s are not sufficiently evident for detection, interpretation and, consequently, indexation.

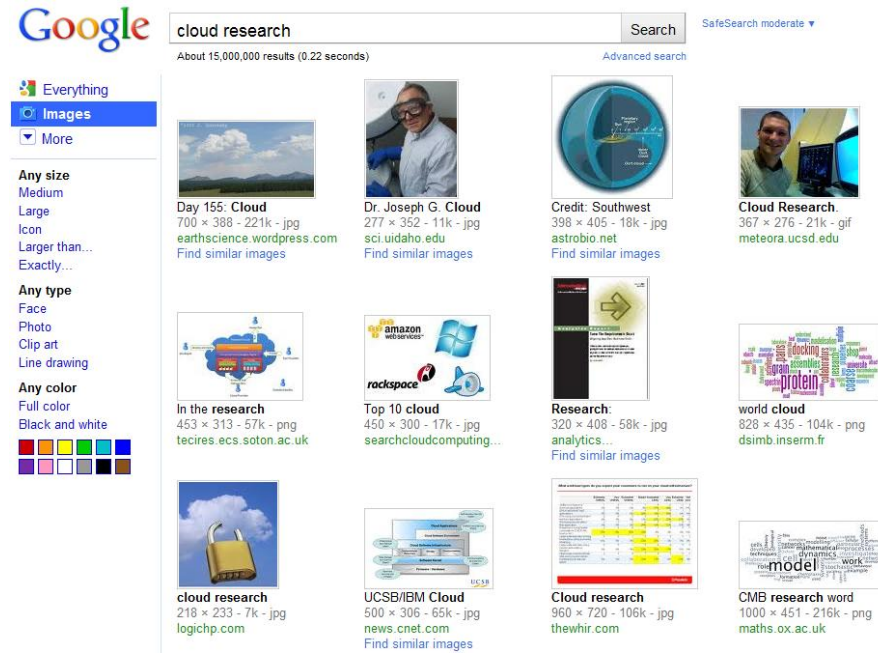


Fig. 1.3 Ambiguity in retrieved media in Google Images, from [42]

1.2 Research Motivation

The question that remains is: *since the description of media is so relevant and impacting on people's daily lives, how can it be done in a large scale, in order to immediately start a revolution?*

The main challenge for a broad adoption of multimedia description is related to the deployment of the idea. Something should be done in order to help developers, designers, researchers, programmers, content producers and providers on the description of the media they are producing and distributing. The production of multimedia contents has been increasingly simplified and the dissemination channels, now widely accessible to ordinary people, has contributed to the creation of an unprecedented amount of media and an audience anxious for original contents.

This problem has been explored before, as we can see in the state-of-the-art, and numerous systems for multimedia archiving and retrieval were created, both commercial and academic, targeting an increasing market. Then, we decided to join this trend and we have created our own framework, which may not be seen as a reasonable starting of an initiative from scratch, but the research on this field is far from being conclusive because supporting technologies are still under development, such as image recognition, speech and sound recognition, compression, languages for description, annotation, querying, etc. Therefore, every initiative is an opportunity to rethink the existing directions and try alternative implementations.

The fact that most works so far haven been limited in terms of supported features and, in some cases, exploring only one specific domain, made us think that just offering a more complete set of features would not be a relevant contribution, because this simple decision was not made before. However, the point is not just to make a larger set of features, but to give the opportunity for the user to be expressive when transmitting his/her interpretation of the media content and allow developers to decide improving their existing applications or create new ones but demanding less effort.

1.3 Thesis

This work provides the foundation through a set of services that enables people to describe multimedia content. We adopted an approach centered on the use of segments to delimit the content and annotations to describe it. We support several types of segments and annotations, giving a rich spectrum of possibilities to individuals who aim to describe multimedia contents. However, we do not specify the “how to” of combining segments and annotations, which is open to interpretation by individuals depending on each scenario. The interpretation of what is or not object of segmentation is also a responsibility of individuals, and even those segments automatically recognized were done in a mathematical and logical implementation. Therefore, it characterizes this work as positivist.

Following a positivist philosophy, this research focuses on the observable and assumes that reality is objectively given and neutral. From this perspective, theories and propositions are analyzed according to their correspondence with the facts seen in an objective world through empirical experiences [10]. Common criticism to positivism is due to the search for causal relationships, thus leading to propositions becoming too narrow, which does not reflect the entire context of the situation under analysis. We see researchers as outsiders, independent of what is being observed, thus being able to objectively examine reality.

1.3.1 Thesis Statement

This thesis defines a multi-purpose, extensible and scalable framework for multimedia archiving to empower the capacity of content description. It aims to simplify the work of developers when improving existing applications or creating new multimedia applications designed to describe multimedia contents. This way, the framework has as mission to increase the number of available described media in order to make them better processed by computers.

1.4 Scope of The Research

This research presents a multimedia archiving framework which has as main characteristic a rich support for segmentation and annotation of multimedia content. This framework is validated in two different scenarios to demonstrate its reuse in distinct situations, its completeness to cover most of the demanded requirements, its accessibility and compatibility to several platforms and its ability to scale to a constantly increasing volume of data.

In order to balance effort and results, some research directions have been interrupted or ignored up to now. In terms of technical aspects, despite the importance of system performance, no optimization and performance tests were implemented so far because more priority was given to the research context's requirements. The short effort on the implementation of User Interfaces (UI) and lack of time on the project did not allow to carry out enough experiments with end-users. The tests were performed in the architectural and implementation levels. Even knowing the impact on performance, the communication is exclusively made over the Internet and no low level communication was implemented. We are using some recognition algorithms, but they are not contributions of this work, but reused from other projects. All implementations were made in JavaTM to profit from its multi-platform support, but we do not guarantee that the framework will run in any other platform, but the Ubuntu Linux Operating System³, where all development was hosted.

³ <http://www.ubuntu.com>

1.4.1 Target Audience

The multimedia archiving framework, described in chapter 3, was developed to simplify the work of developers when improving existing applications or creating new multimedia applications designed to describe multimedia contents. It may also help researchers on the investigation of problems related to the processing of large multimedia datasets. It has been used as a back-end for applications that perform recognition on media files and need to store the extracted data. At the same time, the support for manual segmentation and annotation is needed when recognition is not present or not robust enough for the case. A front-end was developed to support these manual operations, which, consequently, expands the interest to other audiences besides researchers.

In the context of the IRMA Project [59] the multimedia archiving framework was adapted to become a multimedia library for audiovisual companies, such as television channels, radio stations, surveillance companies and others. Another project, 3D Media [70], got profit from this and sponsored the addition of 3D content as a new kind of media, besides video, audio and image. It extended the audience to animation companies, engineering and entertainment industry.

1.5 Reading Map

Chapter 2 explores the state of the art on multimedia archiving systems and their main purposes. Within those systems, we investigate their techniques for segmenting and annotating media and how those techniques influence the achievement of their main purposes.

Chapter 3 describes the multimedia archiving framework developed in the context of this research. It gives an overview of the supported media types, explains how media are archived, indexed and retrieved, formalizes a workflow for the media processing and goes into details on the architecture and implementation, from the user interface to the media storage.

The following two chapters describe two case studies where the media archiving framework was applied. Chapter 4 presents a new approach of user interface modeling based on the analysis of screenshots. This analysis produces segments and annotations to represent user interaction models. These models help to describe the elements of the user interface, the navigation between screens, windows or pages, and the flow of interaction through UI elements. This kind of analysis, using image processing on screenshots, seems to be more robust than existing practices of user interface analysis through

the source code. It uses as an example a user feedback application, where end-users give feedback drawing segments right on the screenshot to annotate problems, suggestions and tips. Chapter 5 describes a multimodal high level fusion system that analyzes a typical office scenario where two co-workers had a conversation and, based on their behavior and speech, we realized their intentions and managed to give advices in order to help the execution of their tasks. Each modality is analyzed separately and their results are combined afterwards to deduce people's intentions. The framework was used to manage a long-term memory, preserving what was learned from the modalities.

We conclude the work in chapter 6, summarizing the contributions made during this research experience, discussing the results and presenting our short and long term vision of the future.

Chapter 2

State of The Art

The state of the art is divided in three main parts: 1) exploration of existing multimedia archiving systems in order to identify relevant features and the direction of their evolution; 2) support for segmentation and annotation and how rich they are in terms of representativity and adoption of standards; and 3) applicability of multimedia storage and retrieving on different problems beyond the traditional media library to organize collections.

2.1 Introduction

The definition of multimedia extracted from the Oxford English Dictionary is the “use of more than one medium of expression or communication”, covering, as an example, from the mixture of dance, music and interpretation to the mixture of digital media, such as audio, video, image and text, which makes it have a different meaning to different people. A more objective concept was expressed by the Merriam-Webster dictionary, which states that multimedia is “a technique (as combining of sound, video, and text) for expressing ideas (as in communication, entertainment, or art) in which several media are employed”. In general, the existing concepts of multimedia are very similar, which shows a common understanding of what it is without visible contradictions. It brings us to the assumption that a fundamental requirement of a multimedia system is to support several media forms.

Multimedia systems are used for different purposes including industrial, medical, biological, entertaining, security, engineering and many other applications. These systems are complex because they have to deal with processing of large binary data, organized in several formats, encoded and decoded for efficient distribution. In addition to that, there are also many restrictions related to copyrights, that may vary according to the country of production

and the country of consumption. Other restrictions, such as age, available reproduction standards, and legal copy acquisition are also frequent.

The availability of the Internet and the facilities to access network infrastructures influence multimedia systems to become distributed. It implies the need to provide availability of media resources to several users, locally and geographically distributed; the access control to these resources; among other requirements. Without distribution, multimedia systems cannot enable collaborative work, consumption of media in large scale, sharing of resources and information about media. These are highly demanded requirements nowadays, besides the fact that only distribution allows scalability to manage an increasing volume of multimedia data.

As technology evolves, multimedia systems implement expert algorithms focused on the recognition of patterns in media contents. It assists users by automating tasks, such as localizing and organizing subjects within media files. Recognized elements are delimited by segments that visually represent recognized patterns to the user and can be stored for posterior retrieval and comparisons. When stored, annotations are used to identify and describe segments. Feature recognition has gained attention and popularized by generic tools, such as Google Picasa, which uses faces recognition to classify pictures, and specialized ones, such as Medical Studio [109] that can describe contents of radiographies and 3D models.

Distribution leads to more users accessing media resources and the presence of meaningful elements in media contents leads to more annotations made by different people, with different expertise. Each individual thinks in a very particular way because his/her reasoning depends on their education and experience. Consequently, there is the risk of inconsistency, ambiguity, conflicts and other problems when users are annotating media. To address these issues, some multimedia systems added support for multi-domains in a formal way. This term refers to multiple domain knowledge, which is the knowledge used to refer to an area of human expertise or a specialized discipline. A domain is composed of a set of concepts and the relationships between them [39]. Using standardized meta-languages, such as OWL and RDF, we can make domain concepts readable by computers. Using these concepts to annotate media, users can be restricted to their domain of expertise, which may reduce the problems previously cited.

One last important trend is the personalization of contents according to the user profile. Personalization uses the context information of previous use and physical location to accommodate the differences between who interact with adaptive applications [47]. Therefore, properties of the user can be used as parameters for queries. The retrieval would not limit the number of results, but the relevance and the order of appearance would be customized according to the user's preferences and expertise. Incidentally, knowing the user exper-

tise through his/her profile, it is possible to influence the annotation process when multiple domains are considered.

These important aspects of multimedia systems are, one way or another, dependent on how media resources are stored and retrieved. Security, copyright, licensing, and other distribution constraints depend on the implementation of an access control module to check permissions of those media, which are candidates of retrieval. The storage and retrieval of multimedia resources locally do not allow distribution, thus the system is limited to only one user section at a time and its computational capacity restricted to a single computer, which might impact the performance of feature recognition algorithms. Moreover, the full potential of features, such as segmentation and annotation is only achievable when they are accessible to several users. Therefore, media resources should be stored in an accessible environment and capable of expansion, both in the storage capacity and in the processing power.

2.2 Requirements

We have taken these aspects as an inspiration to define a list of requirements. They drive the study and comparison of existing multimedia archiving systems, in order to explore the evolution of such systems until the current state of multimedia storage, description and retrieval. Relevant concepts, such as segmentation and annotation, are emphasized due to their importance on the description and retrieval mechanisms. These requirements are also used as parameters to highlight our main contributions. The requirements are:

- **Multiple Media:** the support for multiple media forms is a fundamental requirement of every multimedia system.
- **DRM (Digital Right Management):** since multimedia content is an intellectual creation, it is subject to copyright policies. These policies are managed by a DRM module, which controls the distribution, accessibility, versioning and other aspects of media protection.
- **Distributed:** the application is accessible everywhere the network is available, supporting several users, concurrent connections, and serving media content in a growing scale.
- **Segmentation:** users can excerpt media to delimit relevant contents and the same structure can be used by detectors to permanently persist recognized features for posterior searches and comparisons.
- **Annotation:** usually implemented with the segmentation to allow the description of segments' content.

- **Feature Recognition:** depending on the target audience, the application can recognize some patterns in the content, such as people's faces, predominance of colors, etc. This requirement can help on the generation of segments and assist on creating annotations.
- **Personalization:** creating and maintaining a user profile, the system can provide some personalization considering user preferences and interests.
- **Multi-domain:** the system can be used or applied for several needs, independent of the application domain.

These requirements are ordered from basic features to advanced ones. Support for multiple media and DRM are considered as basic requirements for any kind of multimedia archiving system. From a standalone to a distributed architecture, the system should evolve technically and it demands a lot of programming effort to avoid data conflicts, deadlocks, inconsistencies, as well as to maximize parallel processing, security, scalability and other challenges. As a plus, segmentation, annotation and feature recognition can be supported to describe the content of media resources. Segmentation and annotation can be done manually and feature recognition is a way to automate these features, increasing productivity when dozens of resources should be described. At last, personalization and multi-domain can profit from distribution and annotation respectively, thus they are expected when the system has gained some maturity. This order influences the organization of our investigation, starting from an overview of existing multimedia archiving systems, then analyzing their support for segmentation and annotation and how rich they are in terms of representativity; and lastly, the applicability of multimedia storage and retrieving on different problems beyond the traditional media library to organize collections.

2.3 Standards

Before going forward, it is important to describe some standards that are taken into account by multimedia systems. They are ISO/IEC standards developed by the Moving Picture Experts Group (MPEG). In summary, these standards are:

MPEG-7 (ISO/IEC 15938) Formally named *Multimedia Content Description Interface*, MPEG-7 intends to describe properties, related information and interpreted meanings of audiovisual data content in multimedia environments. The standard uses XML as the language of choice for the textual representation of content description, facilitating the interoperability between applications that implement MPEG-7 [63]. It tries to encompass many details of the media resource including people involved in the cre-

ation and production process, rules for the use of the content, storage format, segmentation, low level features, even how to browse the content in an efficient way. However, defining which information is covered is also a way to limit its application. On the other hand, it is possible to extend it using description schemes and descriptors, but the decision to use XML brought negative aspects, such as the lack of expressive power present on the XML Schema [13].

MPEG-21 (ISO/IEC 21000) Formally named *Multimedia Framework*, MPEG-21 is an extensive standard aiming to define a normative open framework for multimedia delivery and consumption. It considers digital media resources and users interested in these resources to allow exchange, access, consumption, trade, manipulation and other applications [7]. MPEG-21 defines a XML schema called Digital Item Declaration (DID) to specify the structure of the digital item and provides means to uniquely identify digital items and its parts. In addition to these fundamental aspects, it specifies how to include rights and permissions on digital resources, accompanied by a well-structured dictionary; provides descriptions for tools to assist with the adaptation of digital items; includes methods that can be applied to process digital items; supports systems-related specifications and includes a file format that supports exchange of digital items, including still and dynamic media with both textual (e.g. XML) and binary data (e.g. MPEG-4 presentation). It also considers extending MPEG-7 to fulfill its particular requirements of content representation.

Some of the systems described in section 2.4 implement these standards, and chapter 3 describes how they are considered in the proposed multimedia archiving framework.

2.4 Multimedia Archiving Systems

Exploring the literature and observing other initiatives that are calling attention of the market and communities of specialists, we could compile a list of potential systems that fulfills very well, at least, a minimum of three requirements.

We started investigating MMSRS [100], a Multimedia Storage and Retrieval System that aims to provide enough storage capacity, efficient extraction of video fragments and improved quality of stream. However, it is dedicated to medical image and videos data. To address efficiency in the extraction of video fragments, it adds annotations to the videos grouping the annotations in four classes: persons, events, objects and locations; and subclasses that describe detailed annotation semantics. This aspect is charac-

teristic of a *content-based retrieval application* that has the goal to facilitate transferring and viewing videos by only delivering fragments of videos that correspond to the users' requests. It helps users make a more sound decision before a complete video is retrieved.

In order to expand these storage and retrieval systems towards the use of different kinds of multimedia data (e.g. supporting images, video, etc.) or the capacity of semantic modeling (e.g. description of content beyond keywords), the work in [25] extends a Multimedia DataBase (MMDB) system based on Oracle to use, query, index and store multimedia and its content description using MPEG-7 [92]. In their system, multimedia applications can be implemented to access the MMDB, such as applications for content-based image retrieval and audio recognition. The MPEG-7 MMDB can be used to maintain MPEG-7 documents, to do image search and audio recognition and retrieval. When users need more sophisticated usage, the system enables users to implement new application libraries so they can create more detailed queries to search on the database. When the user needs to create new indexes, the administrative usage enables extending the internal libraries. MMDB focuses more on inserting and querying the stored multimedia data. Feature recognition could be provided if an application is developed for that purpose and digital right management is not fully expressed, it only mentions an internal library that lists essential rights of multimedia contents.

This system has been validated with two multimedia applications: Figure 2.1(a) shows one application in the field of audio recognition, where users can search for music in their multimedia database by music title, interpreter and genre, and also by audio signals recorded with a microphone. Figure 2.1(b) shows an image retrieval application using the blob theory that considers color and texture.

There are several MPEG-7 based software, but most of them are designed to a specific media type. For instance, Caliph & Emir [60] is a MPEG-7 based JavaTM prototype for image annotation and retrieval; Mecca [104] is a hypermedia management system aimed for videos; and a music portal offering content-based music retrieval services [110]. However, multimedia standards, such as MPEG-7 lack expressiveness regarding semantic annotations, for instance, semantic annotations are decoupled from low-level visual features and can be modeled in multiple ways, which may lead to ambiguity and limited interoperability of tools retrieving and manipulating multimedia content [48].

The *User-Centered Multimedia* (UCM) environment intends to put users as an active and controlling element in the process of presentation creation, selection and rendering [9]. In this environment, the content is stored in a media repository system that can later be accessed, transferred and rendered. They have developed an environment with a personal digital recorder device that serves as content storage, management and sharing aimed to support peer-oriented information distribution. They work with both composite (all

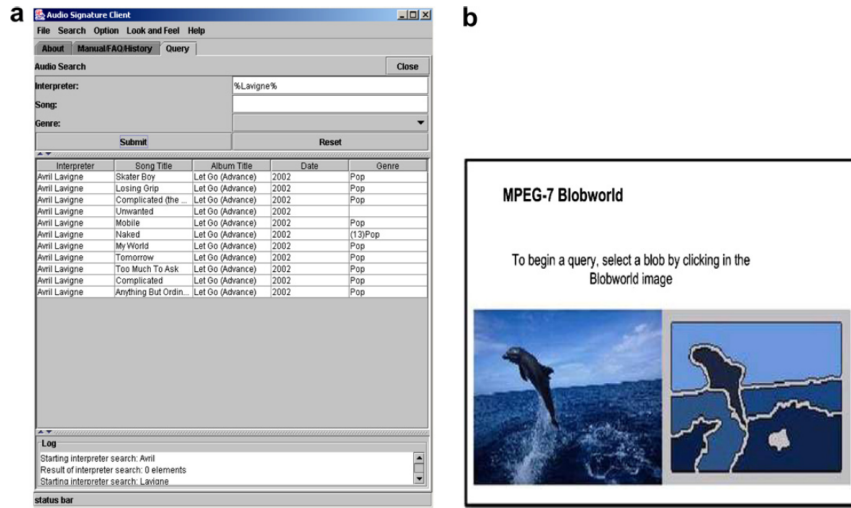


Fig. 2.1 MMDB (a) audio recognition (b) blob-based image retrieval, from [25]

of the content is accessed via a single transfer control stream) and discrete (a collection of related independent media components) objects. With discrete objects, multiple components may be stored in a distributed environment and need to be synchronized for presentation. They propose three control options for users when interacting with multimedia data. First, it decouples media components so users have more freedom in selecting the content they need. Second, it segments the multimedia content in diverse components in order to enable users to navigate through the content other than through a timeline-based navigation. Third, it enables users to specify accessibility to certain segments with a simplified rights management approach and also to add annotations to segments (e.g. links related to the content). Their solution to address rights management constraints is to place control operations in an intermediate file that is used as the basis for content sharing. The focus of their proposal, however, is to support user-control and personalization within multimedia presentations so users can edit content only as they are interacting with it; thus there is no support for content managers.

SemaPlover [94] is an application that enables users to explore and view semantic data. The SemaPlover storage infra-structure consists of different semantic data sources, even though it is mostly focused on images. Together, they form a heterogeneous data set hosted on virtual machines on Amazon's Elastic Computing Cloud (EC2). EC2 implements a special architecture called cloud computing, which is Internet-based computing, where shared resources, software, and information are provided to connected devices on demand [113]. Even though this application is using a cloud computing infra-structure, it is not itself in the cloud because it does not encapsulate how

it accesses or manages the distributed data sets. In any case, SemaPlorer uses blended browsing and querying to let users interact with the application while different queries are built in the background and sent to the storage infra-structure. The facets used to do the blended browsing and querying are location (e.g. population of a city), time (e.g. weather in a certain period of time in that city), people (e.g. people currently visiting that city) and tags (e.g. tags from Flickr).

In SemaPlorer, users initially state a text query on the top left corner of Figure 2.2. When the user selects the city of Berlin, the application shows a map of the city at the center of the figure. At the same time, a query is executed to show interesting places on the map and a second query is executed based on what is currently seen on the map to fill the context view with information about the location, people, time and images related to the selected city, depicted on the right hand side of the figure.

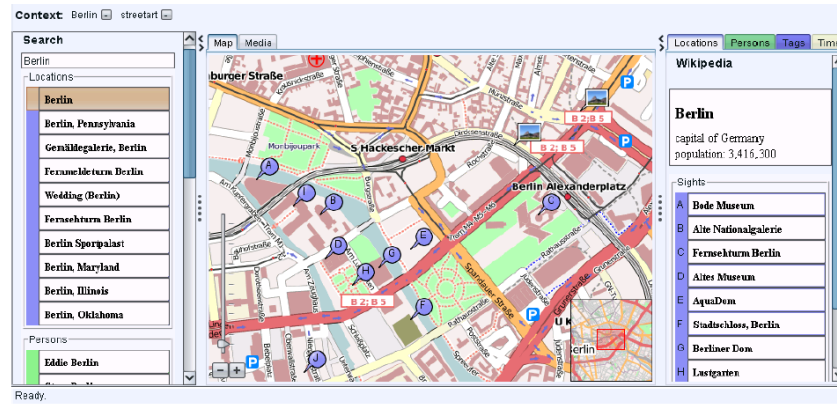


Fig. 2.2 SemaPlorer application showing street art in Berlin, from [94]

Considering the MPEG-21 standard, some systems define their requirements and architecture based on this multimedia framework. [82] proposes a system compliant with the MPEG-21 Digital Item Processing specification to give content authors full control of how to process their digital items. This system is applied in the context of an interactive museum where people can access multimedia content provided by the museum using a PDA containing an MPEG-21 terminal.

[54] proposes a system architecture for a medical information system using the MPEG-21 multimedia framework. It uses wireless sensors to detect information about the patients such as blood pressure, video for remote assistance for surgery, preoperative medical images, etc. The MPEG-21 framework helps with the organization and exchange of medical data between sensor nodes,

networks and terminals; and with the user interaction regarding data filtering and information tagging.

There are also some systems that are focused on the adaptation of digital items ([45, 58]) or on digital rights management ([98]), but they are not detailed in this analysis because these aspects of MPEG-21 are considered as future work.

While MPEG-21 is being used in several applications, it also has a series of shortcomings. In [96], the authors have identified that the MPEG-21 Digital Item Adaptation standard contains some restrictions in order to describe scalable video bitstreams. More precisely, there is an increasing memory consumption and a decreasing generation speed as a function of the bitstream length. This is practically unacceptable in the world of digital video, which cannot simply be solved by software optimizations. While in [101], the authors have identified that MPEG-21 cannot accommodate multiple adaptation requests or manage shared resources, neither does it allow to describe users' adaptation history.

2.5 Support for Segmentation and Annotation

Segmentation and annotation are not required features in multimedia archiving systems, but they are requirements of this research. The work of John Smith, recognized multimedia researcher, was responsible for the development of the IBM Multimedia Analysis and Retrieval System (IMARS) [71], available at [alphaWorks](http://www.alphaworks.ibm.com)¹. This system is used to index, classify, query and retrieve multimedia contents. It automatically analyzes visual characteristics of images and videos, allowing a faster organization of large amounts of content. Figure 2.3 shows a simple search box interface as well as an advanced visual user interface that makes digital image and video databases searchable.

Jyi-Shane Liu et al. [57] proposes a very concise workflow with well delimited segmentation and annotation phases, but the solution only supports the annotation of pictures. Purchase and Naumann [86] created a multimedia semiotic model called TOMUS. This model is organized in three dimensions (signal, syntax and modality) using definitions from semiotics. Signals can be represented by concrete icons (photography of the author of the book), abstract icons (road sign) and symbols (the word 'exit' written on the door). Syntax can be individual (unique object: photography, sound), augmented (adds characteristics to the object, contributing to its meaning, such as color, size, etc.), temporal (considers time to communicate the meaning of the object, e.g. a movie of a moving car can show that the car is actually moving),

¹ <http://www.alphaworks.ibm.com>

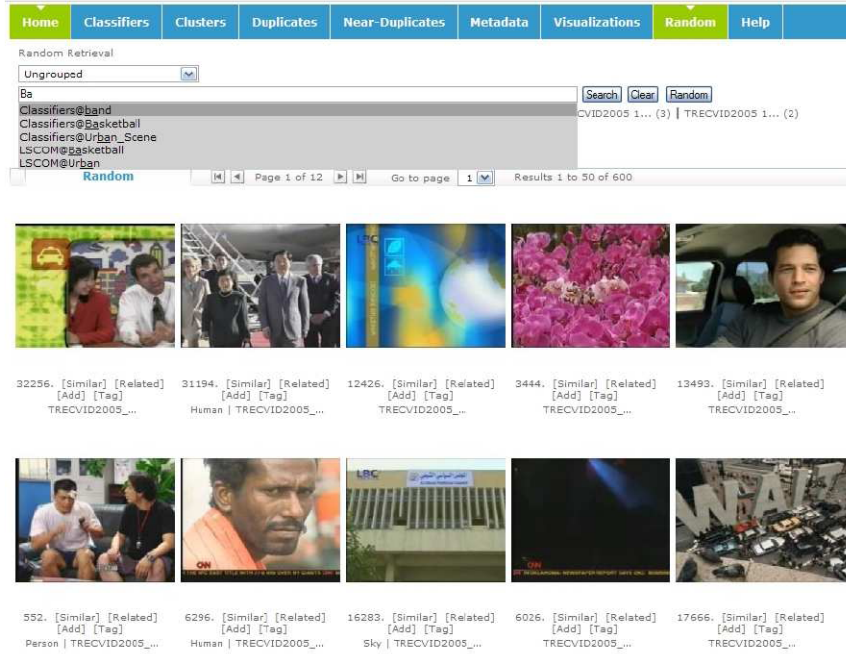


Fig. 2.3 IBM Multimedia Analysis and Retrieval System, from [71]

linear (positions the object in a sequential way, e.g. a movie of a train arriving and leaving the station) and schematic (uses spatial indicators to show the information structure, e.g. graphic). Modality can be represented by vision and hearing. The model can be used to define a composite multimedia communication, where each component is represented by a different modality (e.g. an audio-visual museum guide).

Simoni Santini [93] calls attention to the fact that representing the nature of a multimedia signal is a very complex task. Defining the meaning of a multimedia signal, like an image, is influenced by cultural aspects. A multimedia signal can, at the same time, represent an icon and a symbol. A picture of a red Ferrari denotes a red Ferrari, which is an icon, but also denotes richness and success, which is a symbol. In this case, the multimedia signal interpretation is made in two stages: an interpretation of the signal and its connotation. However, a lot of attention has been directed to a standardization of the first stage (MPEG-7, RDF), not for the second stage, which represents a relevant factor for research on semiotics and multimedia.

The environment proposed in [9] has segmentation and annotation features, however the segmentation is done beforehand and presented to the user as a facilitator for navigation within the multimedia content and the

annotation is resumed to adding links to contents. In addition, the segmentation and annotation of video is made on the user interface that presents the media and not directly on the media content itself. Figure 2.4 shows that a user can select individual parts of a newscast and add links to associated content.



Fig. 2.4 Content segmentation and annotation, from [9]

The support given for segmentation by [25], which relies on MPEG-7, encompasses visual and audio multimedia data, such as videos, images, however it does not support vectorial images and animations. They support textual annotation with free text, structured text using specific fields and complex textual annotation with syntactic dependency between grammatical elements forming sentences.

There is a *unified framework* that automatically classifies an image by recognizing, segmenting and annotating objects within the image [53]. It proposes a hierarchical generative model to identify and tag objects in an image as a collection of patches and regions. The model also predicts which tags are visually relevant in the image and automatically learns, hence not only accounting for irrelevant tags but also for missing correspondences between visual concepts and textual annotations. This framework has been tested and the results show that it outperforms the other classification, segmentation and annotation methods in the literature. However, it is focused only on images and it does not account for human annotations in cases of need to

maintain and improve classifications, neither for categorization of the same image for multiple domains. Figure 2.5 shows the learning process where (a) is the original image, (b) lists tags from Flickr, (c) lists the filtered tags, (d), (e) and (f) show the different stages of the annotation proposed by their approach, where tags are predicted by the visual component and green tags are non-visual.

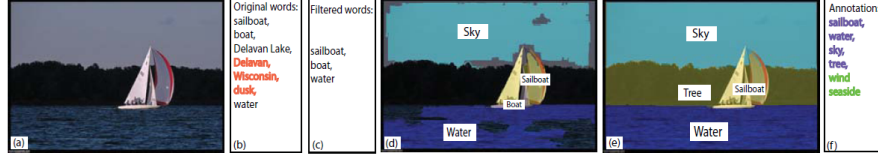


Fig. 2.5 Process of segmentation and annotation, from [53]

COMM [111] is a Core Ontology for MultiMedia annotation that is applicable to describe image, audio and video. It enables decomposing the multimedia content in regions, representing the properties of the content with annotations that can be done collaboratively by several users and associating the annotations with the content. It is able to extract low-level image features, such as color, through image analysis algorithms that can be further annotated by users. Aiming for efficiency in retrieval and presentation of multimedia content, they propose to use SPARQL queries that create shortcuts in the complex annotation graph.

The K-Space Annotation Tool (KAT) is a framework for semi-automatic, semantic annotation of multimedia content. It is based on COMM. Figure 2.6 depicts the KAT image annotation tool displaying an image with annotated persons that were generated by an automatic plugin. It also contains a content browser that displays the result of a semantic query for a specific concept (e.g. person).

Similarly to COMM, [6] through a proposal of semantic annotation, called M-OntoMat-Annotizer, had already called attention to link low-level visual descriptions of multimedia data to semantic Web ontologies and annotations in order to bridge the semantic gap. Their software enables users to draw segments on the image and link concepts to them.

Figure 2.7 shows the VDE plugin where it is possible to load and process visual content. The user can draw a region on the image and apply the multimedia extraction procedure. By selecting a specific concept in the domain ontology browser and selecting a region, the user can link concepts with the image. The visual descriptor selection and extraction manages the low-level feature extraction and linking process.

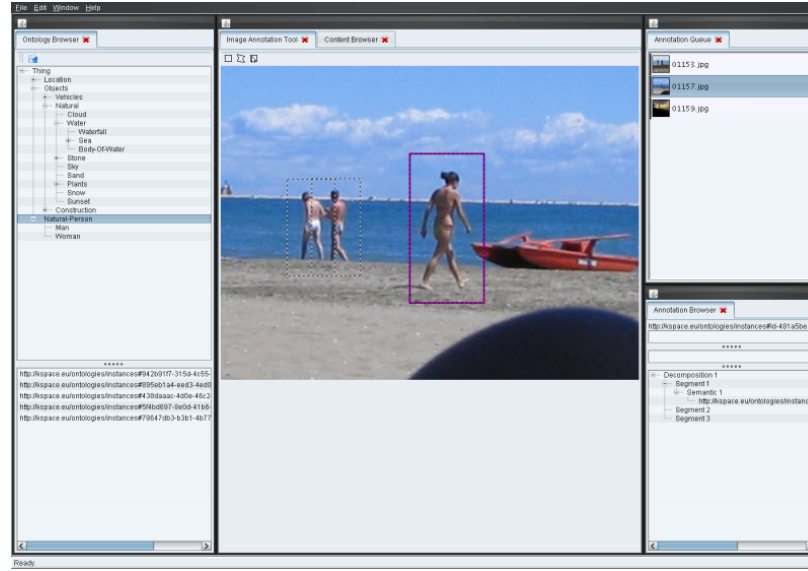


Fig. 2.6 Image annotation tool, from [111]

SemaPlover [94] is able to recognize some information in the content by retrieving the missing information from the connected data sets, but it does not provide annotation by itself. Actually, it inherits annotation by using an external subsystem that is designed to semi-automatically annotate multimedia data (KAT), the same one as used in COMM [111].

2.6 Multi-Domain

The Multimedia Database system proposed in [25] relies on MPEG-7 to describe multimedia data. Its descriptive features allow describing the same material using different features in order to make it meaningful to different domains. However, it does not use ontology, but creates its own Description Definition Language (DDL). Ontology uses the same language as the ones used by web semantics, such as OWL and RDF, which is more easily integrated with existing applications and enables reusing domains that have been in use for quite a long time.

Using ontology, there is a work that suggests simplifying multimedia ontologies based on MPEG-7 because of their resulting complex RDF graphs that are inefficient for end-users to search. With this goal in mind, they use COMM [111], a core ontology for multimedia annotation that is simpler,

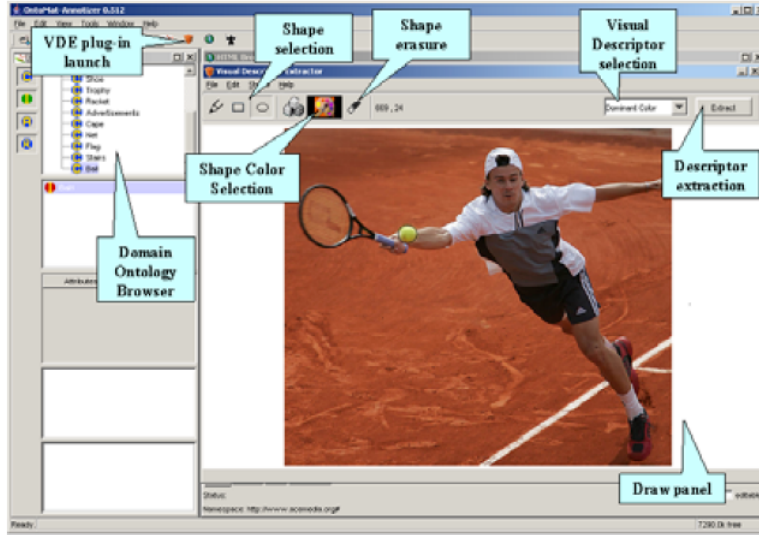


Fig. 2.7 M-OntoMat-Annotizer user interface, from [6]

but still profits from the expressiveness present in MPEG-7. In this scenario, end-users can use any domain specific ontologies to semantically describe multimedia content. However, this approach is focused on the ontology and it does not specify the architecture of the supporting tool, neither how it supports a distributed environment.

A similar approach [6] supports describing concepts using multiple descriptions of a domain concept, considering different perspectives or varying visual characteristics of a concept, but inside the same domain, not across domains. For instance, a tennis racket might have its shape described in terms of viewing angle and occlusions; not using sports (e.g. shape), news (e.g. used by whom in which tournaments) or manufacturing (e.g. brand) domains for the same concept.

2.7 Comparative Analysis

To summarize what was gathered throughout this chapter, we have compiled a matrix comparing qualitatively the most relevant works investigated on the literature. The highlighted systems are COMM, MMSRS, M-OntoMat-Annotizer, MPEG-7 MMDB, SemaPlorer, UCM, and Unified Framework. Figure 2.8 depicts these systems in an alphabetic order and the requirements are presented in the same order they were described in section 2.2. The level

in which a system implements a requirement is represented by three signs: 1) “+” means that the work fully addresses the requirement; 2) “-” means that the work does not address the requirement; and 3) “+-” means that the work addresses the requirement to a certain extent, as discussed in the previous sections.

	Multiple Media	DRM	Distributed	Segmentation	Annotation	Features Recognition	Personalization	Multi-domain
COMM	+	-	-	+	+	+	-	+
MMSRS	+	-	+	+	+	-	-	-
M-OntoMat-Annotizer	+	-	-	+	+	+	-	-
MPEG-7 MMDB	+	+-	+-	+	+	+-	-	+
SemaPlorer	+-	-	+-	-	+	+-	-	+-
UCM	+	+-	+	+	+	-	+	+

Fig. 2.8 Comparison of systems according to requirements

The matrix highlights UCM, MPEG-7 MMDB and SemaPlorer as the ones that better fulfill the requirements. They are compared with the proposal presented in chapter 3. COMM, MMRS and M-OntoMat-Annotizer are the less compliant ones, thus they are not considered for comparison purposes afterwards.

2.8 Conclusion

This chapter presented an analysis of the literature on multimedia archiving systems. On the exploration of these systems, we have remarked that while there are several works that focus only on one type of media, most of the works analyzed in detail support multiple media types. However, they either focus on storage capacity of the media database or on efficiency to extract media.

They also tend to use standards for multimedia contents or applications, but there are still some constraints in these standards in terms of expressiveness regarding semantic annotations beyond the low-level features. To address this issue, some of them use ontologies to provide an easier integration with web semantics. However, these works have been done with descriptions of media within a domain, not across domains, or focused on the specification of ontologies, not on a tool for distributed environments.

On the support for segmentation and annotation, some of the works provide feature recognition to help with annotation but do not account for human annotations to maintain and improve classification of media content. There are those tools that support segmentation and annotation, but they are simplistic in terms of the annotation capabilities that are offered (e.g. just keywords, links). The ones that go beyond that, offer free text and structured text annotation; however, they do not support all types of media (e.g. vectorial images and animations).

Some of the works propose tools solely devoted for end-users and focus on storage and automatic feature recognition to facilitate retrieval with no support for content managers, and it is rare to find applications beyond the traditional media library to organize collections.

Chapter 3

Multimedia Archiving Framework

This chapter describes a multimedia archiving framework developed in the context of this research. It gives an overview of the supported media types, explains how media are archived, indexed and retrieved, formalizes a workflow for the media processing and goes into details on the architecture and implementation, from the user interface to the media storage.

3.1 Introduction

The richness of multimedia contents makes them special resources that should be treated appropriately. Contents of pictures, videos, audios, 3D models, etc., are full of meanings, represented visually, aurally, spatially, and interpreted in many different ways.

Taking this evolution as a vision, we decided to consolidate our researches in signal processing and multimodal interaction, identifying what could be a technological basis to make more robust solutions for these fields. Signal processing research works with large datasets of media and these media should be organized somehow to fulfill machine learning algorithms, to preserve obtained results, to allow parallel processing of the same dataset, and so on. Multimodal applications count on signal processing to give more freedom to the user interaction and rely on a long term memory to improve the precision of recognition algorithms.

The consolidation resulted on the development of a multimedia archiving framework, aiming to help researchers on the investigation of problems related to the processing of multimedia datasets. It works as a back-end for applications that perform recognition on media files and need to store the recognized data. At the same time, the support for manual segmentation and

annotation is needed to fix existing data or when recognition is not present, or not robust enough for the case.

This multimedia archiving framework is denominated Yasmim. Within the last three years, this framework was conceived and developed in the context of two research projects. In the IRMA Project [59], it was adapted to become a multimedia library for audiovisual companies, such as television channels, radio stations, surveillance companies and others. Another project, 3D Media [70], reused IRMA results and sponsored the addition of 3D content as a new kind of media, besides video, audio and image. It extended the audience to animation companies, engineering and entertainment industry.

While Yasmim was being maintained by those projects, it was also applied in other research problems, as we see in chapters 4 and 5. Consequently, it was gradually migrating from an applied software to a generic framework for multimedia archiving, serving multiple purposes. According to [46, 12, 30, 29], framework is a reusable design together with an implementation. The design represents the model of the problem being addressed in a particular domain and the implementation defines how this model can be performed, completely or partially [88]. Hereafter, we explain how the framework was designed to fulfill the needs of those projects and its applicability in other solutions.

From this point forward, we are going to use the Unified Modeling Language (UML) notation to represent the design of the system. UML is a standard modeling language in the field of software engineering. It includes a set of graphic notation techniques to create visual models of software systems, simplifying the complexity behind them to improve communication and reuse [34].

3.2 Application Modules

Yasmim was developed in a modular way to allow its extensibility according to the needs of its several applications. Figure 3.1 depicts the interdependency between the modules of the framework. Right in the middle is the *resource* module that manages media files by organizing them in the file system, avoiding duplicity, updating, removing, versioning, etc. This module depends on the *access management* module, which encapsulates DRM's features and takes advantage of the operating system security layer and the application server authentication and authorization services to protect the resources. The *segmentation* module access resources to delimit their relevant contents. The *annotation* module describes the segments. At last, the *retrieval* module uses information from resources, segments and annotation to retrieve media according to queries built by end-users.

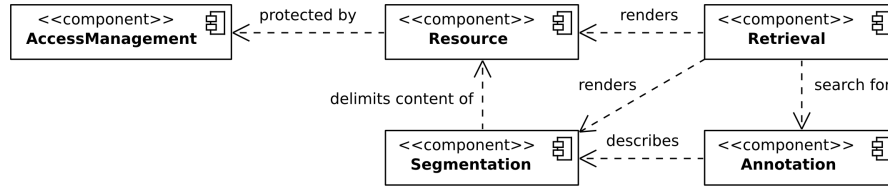


Fig. 3.1 Component Diagram showing dependency between modules

Based on [80], these modules are:

- *highly cohesive*: cohesion refers to the module's degree of focus on its purpose. It means that internal elements of the module are related to one or more module's purposes. A highly cohesive module has more focus on a specific purpose, which is a good design practice to avoid the risk of its elements evolving in different directions. Yasmim is highly cohesive because each of its modules has one and only one specific purpose.
- *loosely coupled*: coupling refers to the degree of interdependency between the modules. Loosely coupled means that a module has low degree of dependency with other modules, while tightly coupled leads to a high interdependency. Minimizing dependency is a good practice because modifications on third-party modules will not have a strong impact on the dependent module, reducing failures and maintenance costs. Yasmim is loosely coupled by avoiding bi-directional references between modules.

3.3 Access Management

The access management module allows users to identify themselves to the system through user name and password. Once authenticated, users can perform operations, such as search, view, export, upload, update, and others, if they have the required authorization. Authentication and authorization is performed in an application server level, as described in section 3.10. The configuration of these features is described in Appendix C.4.

Authorization is divided in two parts:

1. *Authorization to execute a functionality* (e.g. add new annotations to a segment or delete a resource): user groups are created according to the user profiles foreseen for the application. Then users are distributed through these groups according to their respective profiles. Each group has the right to execute a set of features, thus users in the group inherit these rights.

2. *Authorization to access the media* (e.g. video not available in the user's region, minimal age not met, etc.): defines constraints to access the content according to copyright's terms, also known as DRM. The media starts with no restrictions at all and they are added if and when necessary. Constraints can be related to location (e.g. a photo album can be shown in France, but not in Germany), age (e.g. a video is appropriate only for adults), licensing (e.g. mp3 music to sell in an on-line music store), and others.

For the moment, DRM only refers to the whole content of media. Or the access is given to the whole content or no access is given at all. It does not protect segments neither annotations, which could be a special need required by future users. Therefore, it is highlighted in the conclusion as a future work.

3.4 Resources

In the Resource module, *Resource* is the main entity and it is extended to each kind of media: *Image*, *Audio*, *Video* and *Animation* and *3D* [79]. The entities linked to resource are also valid for its sub-classes, thus, each resource has a location in the file system; participants, who collaborated to create the media, assume specific roles; and the user who added the resource in the repository may be (or not) a participant, but he/she is primarily considered as a contributor.

Figure 3.2 uses the UML notation to represent details of the resource module. According to the notation, the rectangles represent classes that can be instantiated to objects, following object-oriented principles. A class has three parts: 1) *identifier*: a name that cannot repeat in the same model, located on top of the rectangle, indicating the name of a logical entity; 2) *attributes*: properties of the class, describing the object that it will represent; there is also the 3) *methods*: which define the behavior of the objects but they were omitted because the diagram intends to represent only data.

Classes are linked to each other through arrows of different types. An arrow represents an association between classes. The direction of the arrow indicates that the origin is aware of the target, but the target is not aware of the origin. Taking as an example the association between *Image* and *Resource*, we have an inheritance indicated by the closed triangular arrow. *Resource* is not aware of *Image*, but *Image* can access attributes and methods of *Resource* and even modify them. The association between *Contribution* and *Resource*, an open arrow, is unidirectional to *Resource*, meaning that *Contribution* has an attribute of the type *Resource*, virtually next to the basic type *Date*, but it does not appear in the list of attributes because it is already represented by the association. When the association does not have any arrow it means

that both sides are aware of each other, which is the case of *Resource* and *Location*. Both classes have an attribute referring to the other part.

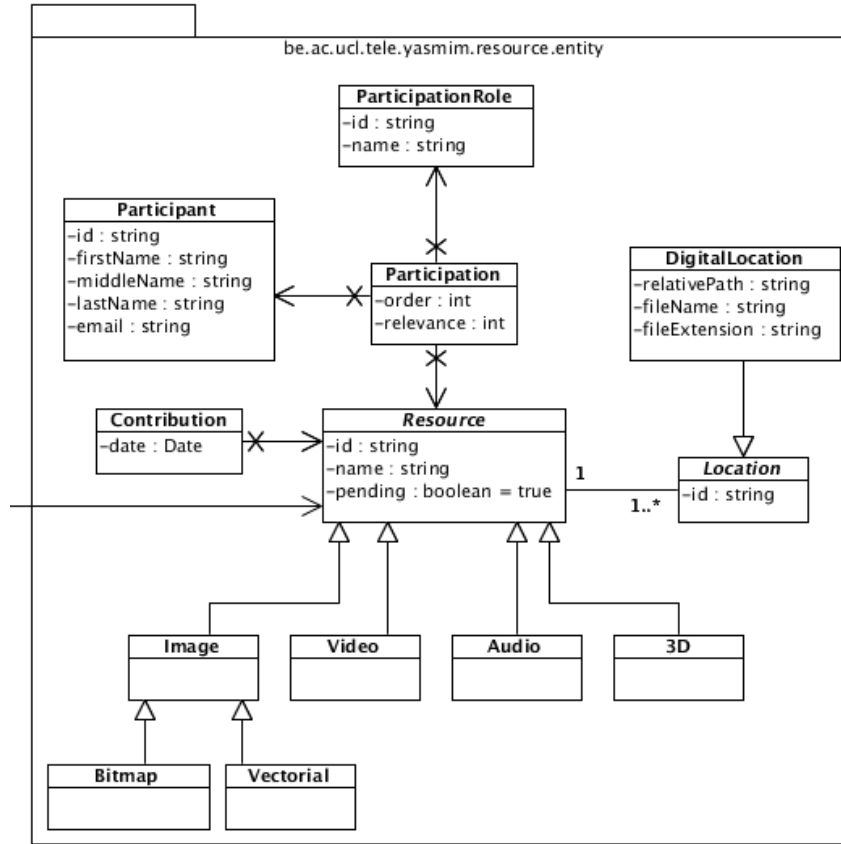


Fig. 3.2 Resource Model

The entity *Location* was designed to support several kinds of locations, but only digital locations, places where binary data are stored, are supported. Positioning location as an abstract super-class, it is possible to extend the support for physical location, for example, which might be useful for multimedia libraries and old contents stored in magnetic artifacts. However, physical locations are not in the scope of this research.

The production of multimedia contents, sometimes, demands the effort of a lot of people. Since we are specially interested on the content and its subjectivity, it is important to keep track of those people who participated in the production process. This is the function of the *Participant* entity,

which is associated with the resource through the *Participation* entity and qualified in the production process through the *ParticipantRole* entity. As an example, a movie is one of the most complex kinds of media produced. For its realization, it was necessary the contribution of the writer, the director, several actors and actresses, cameramen, assistants, photographer and many others. Usually, at the end of the movie, participants and their respective roles are listed. These entities enable keeping track of their contributions in a computational level.

After describing the entities related to resources, we are going to give more attention to the supported resources, formats, low level features and standards. In summary, Yasmim is able to segment and annotate images, audios, animations, videos and 3D models. Within these media forms there are also dozens of formats and the framework can virtually support any of them because it works overlapping a logical layer, where segments and annotations are abstracted over the media content, instead of physically interfering in the file integrity, which would be very complex in the case of compressed files. In practice, Yasmim stores those formats that are supported by media clients, such as video and audio players, photo albums, etc. For the purpose of demonstration, we are going to use JavaTM and JavaFXTM technologies to develop those clients. Consequently, we are going to consider only media formats supported by those technologies. We describe details of these forms and formats as follows.

3.4.1 Image

Image is static, bi-dimensional and can be bitmap or vectorial. A bitmap image is composed of pixels and each pixel is endowed of color and dimension, influencing the image size and resolution. Image processing consists of applying mathematical algorithms over these pixels to identify or modify their characteristics. The supported formats are JPEG (method of compression for photographic images of the Joint Photographic Experts Group), GIF (Graphics Interchange Format), BMP (Bitmap), TIFF (Tagged Image File Format) and PNG (Portable Network Graphics). JPEG is a better format for photos and not efficient for graphics. The higher is its compression the lower is the quality and vice-versa. On the other hand, GIF is appropriate for graphics, but supports 256 colors only, which does not make it appropriate for photos. Its differential is the possibility to make some parts of the image transparent. PNG was developed as an alternative to GIF because this one is patented and limited in terms of colors, while PNG is patent-free and supports the same number of colors of JPEG, which makes it appropriate also for photos, but not so efficient in terms of compression as JPEG. TIFF is used to exchange images between applications, supporting several color

models including CMYK, which makes it an appropriate format for printing purposes.

A vectorial image is a set of geometric shapes, of different types, arranged in a way to well represent the desired information. These shapes are mathematically represented and every time the image is going to be rendered the processor and graphic controller translate shapes into pixel values for displaying on the screen. This translation allows vectorial images to have good quality even if it is resized because the displaying pixels are recalculated according to the new dimensions. In the case of a bitmap image, a zoom would decrease its quality because the number of pixels is fixed and the size of the pixel grows instead of multiplying, causing aliasing. On the other hand, vectorial images would not be efficient to represent photos with millions of colors because each one of these colors would be a shape, millions of shapes, which imply on millions of calculations to render a simple photo, causing a delay that does not happen with a fixed number of pixels.

In any case, the image's content, rendered by pixels or vectors, is endowed of meanings. These meanings are spread in the picture as a whole or in specific regions of the canvas and they are subject of segmentation and annotation, as described afterwards.

3.4.2 Audio

While images propagate electromagnetic waves, from a shorter length (violet) to a longer length (red), audio propagates sound waves, an energy that is only able to go from a point to another through a material medium capable of vibrating. In a range of 20Hz to 20 Khz, sound waves can be arranged to transmit information that human beings are able to perceive, such as musical notes, speech, noise, etc.

The supported audio format are mainly: WAV (Waveform Audio File), MP3 (MPEG-1 Audio Layer 3), MIDI (Musical Instrument Digital Interface), and ACC (Advanced Audio Coding). WAV does not consider audio compression, which is appropriate for maximum audio quality. On the other hand, there is MP3, whose popularity was due to its compression capability, allowing long audio streams to be compressed and distributed through the Internet. AAC is an improvement over the MP3 format, capable to support even higher quality sound at a good compression rate, which makes it the standard format for the most popular audio devices, music stores, music players and games. More limited, but not less useful, there is the MIDI format, which is used to allow communication, control, and synchronization between musical instruments, controllers, sound cards, and other devices.

The waves of sound contain information through time, which can be subject of segmentation and annotation.

3.4.3 Video And Animation

Video is a sequence of images that reproduces situations or expressiveness, sometimes synchronized with audio to enhance perceptions and interpretations. It has gained popularity through the Internet due to a gradual reduction of production costs and free web services dedicated to video dissemination, such as youtube.com, vimeo.com and others.

Video files might become very large, making them resources that are difficult to store, manage and distribute. Any innovation on creating efficient ways to optimize videos has a large scale impact. It has been a challenge for many researchers worldwide, which justifies a large number of formats available on the market. We are supporting just a few of them, which are: MPEG (method of compression for videos of the Moving Picture Experts Group), AVI (Audio Video Interleave), WMV (Windows Media Video), and MOV (Apple's movie files extension). MPEG 1, 2 and 4 are ISO standards for videos, representing a gradual evolution of the MPEG format, in terms of compression efficiency and media compatibility (CDs, DVDs, Blu-ray, etc). AVI and MOV are container formats, specifying how data is stored, not coded. For that, there are video encoders and decoders, or simply codec, that compress and decompress respectively. To be able to reproduce them, the player should also include the appropriate codec. WMV and MOV are vendor-specific formats from Microsoft and Apple, respectively, both offering good balance of compression rates and quality. All these formats have specific channels for audio and video.

Information in video's content is spread in several dimensions. It can be in a unique frame up to several frames, in the audio channel and in the synchronization of both. Therefore, a video segmentation meets the complexity of segmenting image and audio in the same resource.

Considering the flexibility of vectorial images, coordinated changes on mathematical variables can imply on effective animations. These animations can be predefined or responsive to user interactions. The technologies to allow that have gained a lot of attention, such as Adobe Flash, Microsoft Silverlight, JavaFXTM, and more recently, HTML5. These technologies have programming languages behind them to interfere on those mathematical variables and render animations according to the application needs. Because of its flexibility, an animation would not be necessarily sequential or dependent on time to represent meaning, like video is. But it is equally challenging to be segmented due to its several dimensions.

3.4.4 Tridimensional

A tridimensional (3D) graphic is composed of elements with width, height and depth. A point in a 3D model is called a vertex, and at least four vertexes are necessary to create a basic 3D object. For more complex objects, thousands of vertexes are mathematically positioned to represent forms in a virtual space. 3D models are vectorial, thus they have the same advantages of the graphics discussed previously (see section 3.4.1), which can be resized without losing their quality and animated by model recalculations.

In a 3D model, information is present on the objects that compose the scene. Consequently, the segmentation of those objects should be equally 3D. On the contrary of other media described previously, here, we support only the 3D model API provided with the JavaTMPlatform (JavaTM3D API) and no specific file format. The support for other formats is part of future works.

3.5 Segmentation

Segmentation, or fragmentation, consists of delimiting meaningful regions of media contents [97]. What determines whether the region is useful or not is the purpose of the system application. This purpose guides the direct intervention of the end-user when manually creating segments and also define the recognition techniques to automatically recognize media contents.

Media's characteristics described before have inspired us to design a segmentation model compatible with several kinds of content. This model takes into consideration spatial and temporal dimensions of the content in order to delimit identified meanings. These dimensions define the types of segment, which are: *Spatial*, *Temporal* and *Spatio-temporal*. Each one of these types suites one or more types of media. Table 3.1 shows the compatibility between segments and media types.

Table 3.1 Compatibility between segments and media types

	Spatial	Temporal	Spatio-temporal
Image	x		
Audio		x	
Video and Animation	x	x	x
Tridimensional	x		x

When the segment is created it does not interfere in the media's content. If they need to be rendered for the user visualization, it is done in a transparent layer over the media canvas. This layer is synchronized with the media reproduction in order to render the segments at the exact instant and location where they were defined. If the user does a zoom, the layer is also zoomed. The same occurs with other operations like translation, rotation, play, rewind, fast forward, pause and so on. The main advantage of this approach is that it preserves the integrity of the content, keeping segments in a logical level.

In order to be flexible enough to adapt to those types of media, segments were designed to be vectorial. It makes it appropriate for vectorial contents, such as graphics, animations and 3D, and for bitmap images and rendered audio waves, unifying its adoption on different media and segment types.

More details about the types of segments, their graphical representation and application on different media, are described as follows.

3.5.1 *Spatial*

A spatial segment delimits a static region in a visual media content. It can be bidimensional or tridimensional. Bidimensional segments can be used for images and video frames. Tridimensional segments can be used for 3D models. Bidimensional segments contain a set of points involving a region of interest:

$$S_s = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \quad (3.1)$$

where x and y are coordinates of the points in the Cartesian plane of the content. The value of x and y are correspondent to the pixels' position of a bitmap media. In a vectorial media, x and y are correspondent to the current canvas size, where the number of pixels is dynamically defined by the graphical controller. The limits for x and y are based on the image resolution and the pixels gradient.

Tridimensional contains a set of vertexes:

$$S_s = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n)\} \quad (3.2)$$

where x , y and z are coordinates of the vertexes. The value of x , y and z are correspondent to the current canvas size, where the number of pixels is dynamically defined by the graphical controller.

The use of spatial segments is more frequent in images, whether bidimensional or tridimensional graphics, and less frequent in videos. The full content of an image can be seen at once, while a video is usually composed of thou-

sands or even millions of frames, and its content is more meaningful when these frames are presented as a sequence. Therefore, the segmentation of only one frame might be insignificant.

3.5.2 Temporal

A temporal segment delimits a sequential fragment of audio or video. It defines when a relevant information starts happening and when it finishes, but without any spatial delimitation. Taking an audio media as an example: a podcast is a series of digital media files that are released episodically and often downloaded through the web. These files contain rich discussions about relevant subjects, thus they are candidates to be segmented. Each interesting part of the content starts and ends in specific timestamps. The difference between the end time and start time delimits the temporal segment. It is also applicable on videos. A TV news, for instance, presents several subjects during a daily edition. These subjects can be distinguished from each other by their correspondent segments. Therefore, the definition of the temporal segment is:

$$S_t = [T_s, T_e] \quad (3.3)$$

where T_s is the starting timestamp and T_e is the ending timestamp, both included. The difference between T_e and T_s is equal to the duration of the segment.

3.5.3 Spatio-Temporal

The spatio-temporal segment is a merge of the Spatial and Temporal segment's concepts. It associates a time tag for each one of an uninterrupted sequence of frames. In an implementation level, it preserves all the characteristics of the spatial and temporal segments, but associates additional properties to the spatial segment, which are a sequential number and a correspondent time instant. Because there is the possibility of having two or more frames in the same timestamp, the sequential number was introduced to keep the sequence of those frames, independent of their timestamp. We can conclude that the formal definition of a spatio-temporal segment is:

$$S_{st} = [T_s(\{S_{s1}, S_{s2}, \dots, S_{sn}\}), T_e(\{S_{s1}, S_{s2}, \dots, S_{sn}\})] \quad (3.4)$$

where T is a timestamp of the temporal segment and S_s is a spatial segment in a certain instant of time. Each timestamp can be correspondent to one or more spatial segments. S_s can also represent a bidimensional (i.e. for videos and animations) or a tridimensional (i.e. for tridimensional graphics) spatial segment. This type of segment is applicable on videos, animations and 3D content and it is appropriate for tracking objects, people and other elements in the scene.

In general, none of the above segment types has constraints about the overlapping of two or more segments. In theory, overlapping segments of the same type or different types does not have any advantage or disadvantage, but it could become a concern when those segments are annotated, since it could generate duplicity or demand disambiguation. In addition, a segment is continuous and indivisible, simplifying its handling, but implying in some limitations further explored in the next section.

3.5.4 Segmentation Model

Figure 3.3 depicts the segmentation model using UML. The notation used can be explained in terms of classes and associations using the same description of figure 3.2, but there are three new elements this time:

1. *Interface*: the element *SemanticContainer* is an interface, whose role is to indicate which classes may contain meanings susceptible to be semantically described by annotations. *Segment* and *Link* (described in section 3.6) are the ones that implement *SemanticContainer*, thus they can be annotated. The implementation of an interface is represented by a dashed line and a closed arrow.
2. *Aggregation association*: occurs between *Temporal* and *SpatioTemporal* classes. This association was chosen because the class *Temporal* can be used in two situations: 1) by itself when representing a temporal segment and 2) as part of *SpatioTemporal*. When *Temporal* is part of *SpatioTemporal* but can independently exist, then it is just aggregated to *SpatioTemporal*.
3. *Composition association*: has a slight difference to aggregation. While an aggregated class can exist independently of the aggregator, the composed class cannot. The association between *Frame* and *SpatioTemporal* is a composition, indicating that *Frame* can only exist within a *SpatioTemporal* object. The same occurs between *Spatial* and *Coordinate*.

Segment is an abstract class, thus it cannot be instantiated. Instead, its attributes and methods are inherited by *Spatial*, *Temporal* and *SpatioTemporal*

classes. *Spatial* class represents spatial segments and it contains a minimal of three coordinates that are necessary to create a closed shape with internal area. Coordinates have a sequence that indicates in which order they are linked. *Temporal* class represents temporal segments and *SpatioTemporal* class represents spatio-temporal segments, which are a combination of spatial and temporal segments. More information about the class *Link* is described in section 3.6.

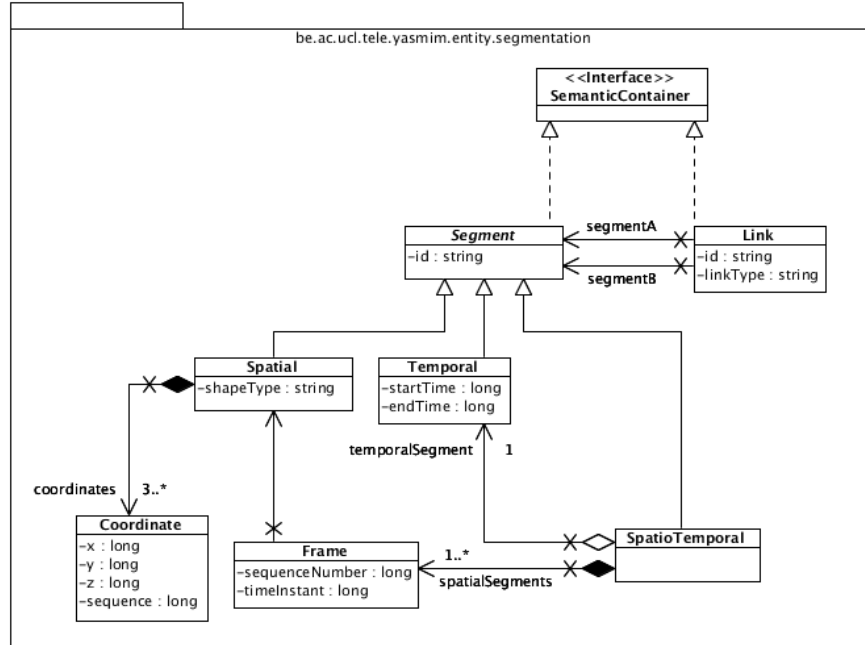


Fig. 3.3 Segmentation Model

3.6 Link Between Segments

Segments by themselves are capable of meeting the needs to delimit several content samples. It allows a precise attribution of meanings to the right location, using annotations. However, there still exists gaps to fulfill in terms of representativeness. These gaps are in the limbo between segments; and our approach to fulfill them is using links.

Links are relationships between segments. Figure 3.4 shows links annotated in the same way that segments are annotated, complementing the representation of the content. Take as an example a bitmap image which shows overlapped elements, as in the case of the coffee maker in this figure that is partially hidden by the arm. When a part of an object is hidden by another object, its visibility is thus restricted to distinct areas that actually represent what can be seen of that object. Actually, only one segment is not able to represent the entire object since parts of it are hidden, but if we use more segments and link them, all visible details of the object will have the same meaning. In this case, the bottom and the top left of the coffee maker are distinct segments that are linked to represent only one object: the coffee maker. A link is also used to connect two parts of an object (e.g. half lemons) and indicate that together they represent one object: a lemon. Another example illustrates the relation that exists between different elements: two spatio-temporal segments trace a football player and a ball, respectively. The link between these segments can describe the intention of the player, which is to chase and handle the ball to achieve the goal. The applicability of links is vast and it depends on the user needs.

The proliferation of links results in graphs where segments are seen as nodes. The arrangement of a graph has several configurations, such as: *sequence*, *hierarchy*, *composition*, *cause and effect*, and many others. Seen as

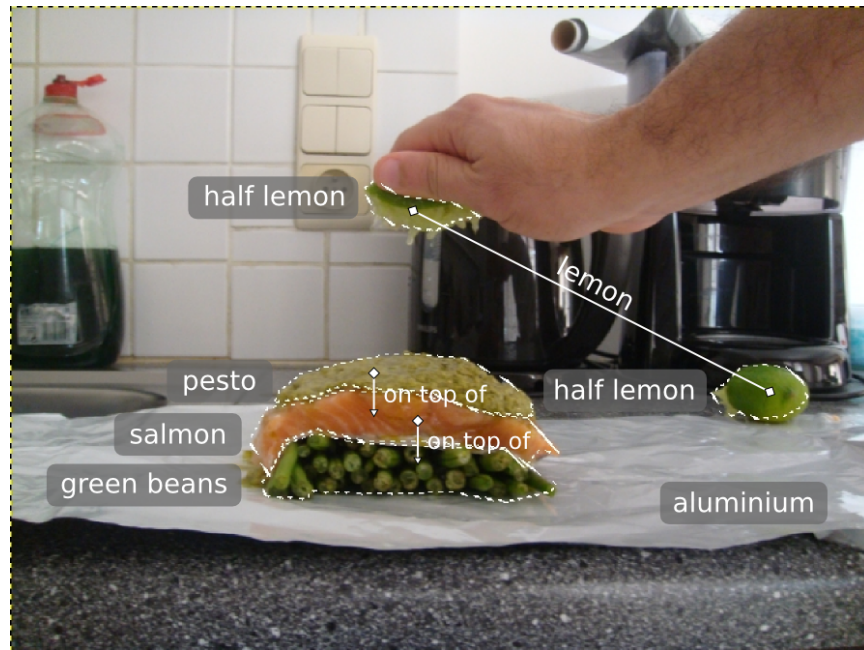


Fig. 3.4 Links represent the relationship between segments

a sequence, the graph indicates that there is a logical order between the segments, a hierarchy indicates the refinement of a large segment in several smaller ones, a composition connects the parts of a bigger element, and a case and effect indicates the impact that a segment may cause on other segments. As we can observe, the contribution that links brings to the content detailing is as important as the segmentation in itself. The meanings that can be embodied in segments and links are described as follows.

A link connects two and only two segments. It can be unidirectional, when a segment has influence over another, or bidirectional, when both sides are complementary. A link and a segment implement the same interface, *SemanticContainer*, that represents both entities externally to the segmentation module. The annotation module, for example, refers to this interface to allow annotation of both entities.

3.7 Annotation of Segments

Annotation is the assignment of meanings to segments in order to describe their contents. When done manually by the user, a segment is selected and annotations are written or dragged to it. When done automatically, segments are created by recognition algorithms, indicating where the elements are located in the media, and described according to what the algorithm was trained to recognize. The annotations can be used to:

- index media to increase the efficiency of search mechanisms;
- allow different people to describe media according to their point of view, from authors to professionals and consumers;
- make media understandable not only for people, but also for computers, allowing interoperability between them and automatic processing of those information; and
- dynamically extend or enrich media contents during their reproduction, but not changing the original content, which is the case of subtitles, labels, translations, and so on.

Yasmim supports several annotation types. They go from a simplistic to a robust form of knowledge representation, giving more flexibility to different user profiles and being domain-independent. They can be assigned to segments and links, covering from simple to complex media contents. Due to the number of annotations supported, we are adopting the classification of annotations introduced by [40] and [89]. There are three categories:

- *Structural Annotation*: describes physical and logical properties of the content (e.g. for a video: its definition, frame rate, dimensions, format, etc), usually extracted from a low level media file processing and from media descriptors.
- *Linguistic Annotation*: has the pure format of text (words, sentences and narratives) and oriented to human readability, structured according to the grammar of the language. They could be referred to also as Textual Annotations or Lexical Annotation.
- *Semantic Annotation*: corresponds to the addition of semantic data or meta data to the content given one or more agreed ontology. Oriented to machine readability, adopts the structure of a graph composed of triples (subject + predicate + object).

Table 3.2 Classification of supported annotations

Category	Annotation
Structural	Property
Linguistic	Tagging
	Transcription
	Description
	AdHoc
Semantic	Domain Concepts

Table 3.2 classifies the supported annotations within these categories. These annotations are described as follows:

3.7.1 Property

A property is an annotation associated to a label or key. This key indicates the meaning of the value. Formally speaking, a property is composed of a *key* and a *value*, where the key qualifies its respective value. Properties are appropriate to describe file characteristics, for example: dimensions, resolution, size, format, volume, duration, etc. As an example of property's syntax we have `<size> = 25GB`, where *size* is the key and *25GB* is the value identified by the key. Some media formats already contain metadata describing the content of the file. The MP3 format, for instance, has a metadata container entitled ID3 [73], used by authors, distributors and others to describe music properties, such as artist, song title, genre, album, etc. These data are in a high semantic level and can also be seen as properties.

3.7.2 Tagging

Tagging is the assignment of keywords to the media content. Each keyword represents a simple word that identifies the content in the segment or in the links between segments. Keywords are simple, efficient and widely used nowadays to create indexes of information on the web. However, it has limited representativeness when compared with other forms of annotation, although it is more practical for most people and more efficient in terms of searching because most database systems nowadays have good support for text searching.

3.7.3 Transcription

Transcription is a textual and complete description of a speech, dialog or music lyric. Practical applications are the automatic recognition of speech in audio sequences, sub-titles, optical character recognition (OCR) in images containing text, etc. It is precise in terms of content representation, but very complex in terms of computation because the extraction of meanings depends on the syntactic and semantic analysis of the transcription. The search is not so efficient either. Large textual contents demand long processing time for each annotation registry. Syntactic and semantic analysis are needed to allow computers to identify meanings in the text. However, none of these disadvantages decreases the need for this kind of annotation technique.

3.7.4 Description

Description is a detailed text explaining the content of the media content. It has the same advantages and disadvantages of transcription, but with a different purpose. Practical applications are story telling material, textual summarization, situation description, scenario-based prototypes, etc.

3.7.5 AdHoc

AdHoc annotations do not have commitment to be accurate in terms of content meaning. They could represent opinions, comments, external links, references, etc. AdHoc also does not have any priority in the searching mechanism.

and it is retrieved when the related media is already available for the user, appearing as an additional or complementary information. This is due to the fact that AdHoc annotations are informal, free-text, and can lead to erroneous decisions [48].

3.7.6 Domain Concepts

A good balance of representativeness and performance is the use of ontologies to annotate segments. Ontology is an explicit specification of a conceptualization. It provides a shared vocabulary that can be used to model concepts and their properties and relations [37]. It is used to describe a domain of knowledge [3], which is composed of a taxonomy of concepts from a certain domain of knowledge, semantic relationships between these concepts, and instances of these concepts that are representations of scenarios under the modeled domain. Concepts are more representative than keywords because they are well positioned in the domain, but they are also less efficient than tagging because there is an additional cost of exploring the graph of meanings related to them. Comparing with transcription, ontologies are less representative than full transcription, but more explicitly, computational friendly, and enable derivation of implicit knowledge through automated inference. The search goes through the graph instead of fully text processing, and no further analysis is needed, such as syntactic and semantic ones.

When an ontology is designed, it describes only one domain of knowledge in order to be cohesive and contribute to its coherence, reuse, compatibility with other ontologies and less risk of duplicity and ambiguity. On the other hand, when annotating a media, the user may find more than one domain represented in its content, showing that the support for multiple domains is needed to address real world situations.

To illustrate it, we have considered the scenario in Figure 3.5. It shows the preparation of a Jamie Oliver’s recipe called “*Salmon Au Pesto and Green Beans*” [74]. It is possible to identify two domains: a) *cooking* brings together all ingredients to prepare the dish: lemon, pesto, salmon, green beans, and aluminum; and b) *kitchen* describes the location where the dish is prepared: balcony, sink, water boiler, coffee machine, and soap. These domains are not necessarily complementary, for instance, the soap is not one of the ingredients, and the ontologies are applied on distinct objects.

Another example where multiple annotations are assigned to the same segment may reinforce the utility of domain concepts. A picture shows a lot of vegetables and three specialists are instructed to annotate this picture. A nutritionist would describe the nutritional properties of each vegetable. A farmer would describe what is necessary to cultivate those vegetables and

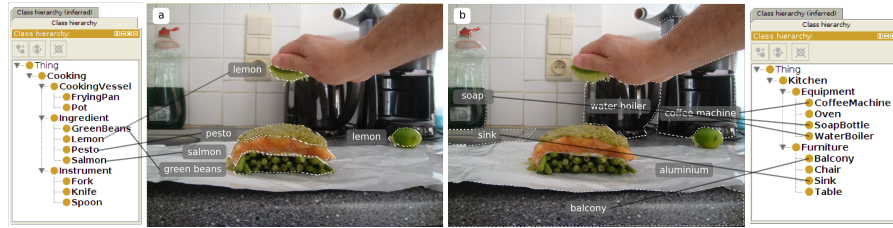


Fig. 3.5 Annotating using (a) cooking and (b) kitchen ontologies

the behavior of the tilth through the seasons, while a chef would see possible combinations of ingredients for an exotic dish. It shows that each specialist needs his/her domain of knowledge accessible to be able to create comprehensive annotations, and the domain of one specialist might not be useful to other specialists.

Yasmim allows the use of multiple ontologies to annotate segments and links. While observing the media, the user decides which ontology is more appropriate for the case, considering his/her domain of expertise. For a better abstraction, each annotation type is presented in a separate layer over the media content, except for the domain concept, which is not a unique layer, but one layer for each applied ontology.

Figure 3.6 depicts the annotation model which is composed of an abstract superclass and several sub-classes, one for each type of annotation. Using inheritance, it is possible to extend the application to support more types of annotations without changing the existing implementation. The class *Annotation* refers to the interface *SemanticContainer*, indicating that every implementation of this interface can use one or more types of annotation available. See the classes that are implementing this interface in figure 3.3.

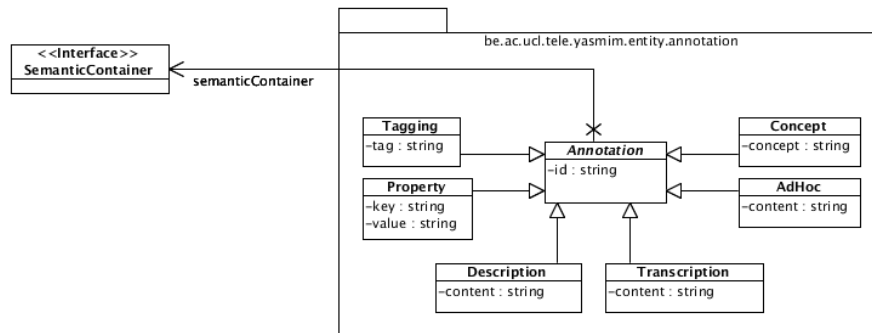


Fig. 3.6 Annotation Model

Going back to Table 3.2, property is a structural annotation because it has a fixed structure, composed of key and value, to qualify a text regarding the media content. Tagging, transcription, description and AdHoc have no fixed structure, but they are richer in terms of textual content, fitting well the linguistic category.

At last, domain concepts may be represented in two languages:

- *Resource Description Framework* (RDF) is a foundation for processing metadata. It provides interoperability between applications that exchange machine-understandable information on the Web [115]. It can represent a graph, making it more powerful than pure XML.
- *Web Ontology Language* (OWL) is a language to develop ontologies with formally defined meanings [116]. Its primary goal is to make Web content more accessible and meaningful to machines. OWL ontologies are primarily exchanged as RDF documents.

These languages make domain concepts structured, but their main value is the semantics that they are able to represent.

3.8 Compliance with MPEG-7 and MPEG-21

The ISO/IEC standards MPEG-7 and MPEG-21, created by the MPEG, were presented in section 2.3. They were considered by some existing multimedia archiving systems, as reported in the state of the art, and after stating how Yasmim describes media, we have enough elements to analyze how compliant is Yasmim with those standards.

We start with MPEG-21, which specifies norms for multimedia framework. MPEG-21 specifies that a multimedia framework should implement digital items, or media resources, as the object that allows interaction between two or more users. Interaction can be seen as creating, providing, modifying, archiving, rating, enhancing, delivering, aggregating, consuming, subscribing and trading resources between parties involved. A party can assume the profile of individuals, consumers, communities, organizations, governments that work with media. Specifically about archiving, which is one of the focuses of Yasmim, MPEG-21 is not clear. Instead, it states that archiving is a recommendation for further work [7]. The same for content description, in which the encoding of XML defined by the MPEG-7 specification is planned to be extended to fulfill MPEG-21's requirements. On the other hand, Yasmim implements some of its recommendations:

- it implements MPEG-21's entity *Digital Item* through the *Resource* entity, which is also a term used by the standard to represent digital items;

- the use of URI (Unified Resource Identifier) to identify resources, which is a fundamental aspect of the architecture, as detailed in section 3.10; and
- the availability of resources through web services, making possible the interoperability between two or more parties, independent of their profile.

We do not intend to implement the full MPEG-21 standard, neither all details specified for the above items. Instead, we plan to improve more and more the descriptive capability of the framework to represent more complex semantic situations. With that in mind, we believe that the MPEG-7 standard has more to contribute than MPEG-21.

MPEG-7 is a standard for describing the multimedia content, which is one of the main objectives of Yasmim. This standard is very rich in terms of representation and can fulfill most of the current needs. It has three elements, as depicted in Figure 3.7, to allow specialists define what they want to collect from media resources, which are:

- *Descriptor* (D) defines the syntax and the semantics of each data that can be collected from media resources;
- *Description scheme* (DS) organizes descriptors (and even other description schemes) in a structured global scheme, as defined by the description definition language; and
- *Description definition language* (DDL) defines the syntax of descriptors and description schemes, allowing the creation and modification of such elements and enabling the customization according to user needs.

These elements are textually represented using XML and systems and specialists should be aware of the syntax in order to process and extend the basics. This XML is associated or attached to the media resource to make it searchable. Yasmim is responsible for making this association between media resource and the MPEG-7 description. Since annotations do not have a specific format, except for property and domain concept, they can be used to store description schemes. *Description* seems to be the most appropriate annotation technique for this purpose.

Other MPEG-7 limitations can be overcome by the rich support offered by Yasmim. For example:

- not limited to XML, thus other formats, such as RDF, OWL [13] and JSON, can also be stored in the annotation database;
- by supporting RDF and OWL, Yasmim can reuse existing ontologies to describe the domain instead of forcing the system to rewrite the domain using DDL, and these languages are widely adopted by web semantics initiatives and DDL is not, which hinders its use on the web; and

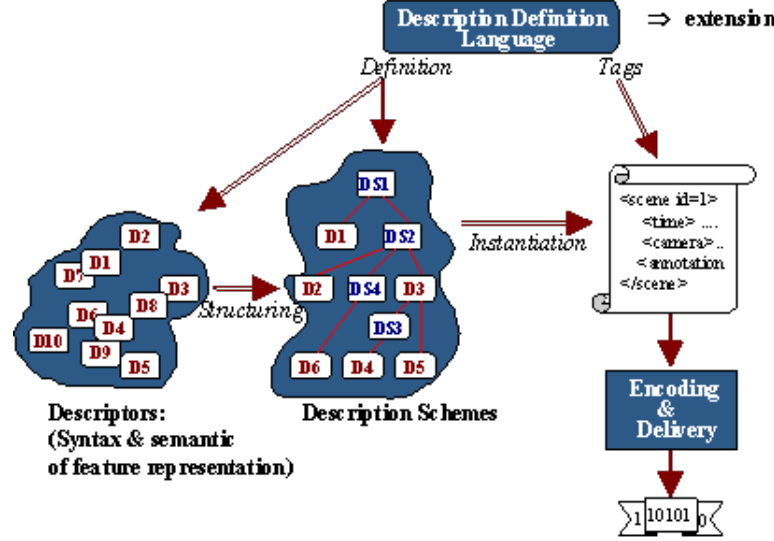


Fig. 3.7 MPEG-7 main elements, from [63]

- Yasmim allows full annotation of links, which is limited in MPEG-7 to simple predicate expressions, such as *part of*, *contained in* or *related to*.

When Yasmim developers decide not to adopt MPEG-7 to describe their media, they always can opt for informal descriptions, using tagging, description, transcription and AdHoc annotation techniques. No additional library or conversion of validation is needed. On the other hand, the segmentation model defined in the standard is more complete and can cover more situations than what was foreseen in Yasmim's model. At the same time, the current segmentation model was enough for the needs of all implementations so far and its simplicity contributes to its adoption.

3.9 Multimedia Archiving Workflow

In order to position segmentation and annotation in the process of multimedia archiving, we have defined a workflow which was also a need of the IRMA project, when aligning the technology with market demands. The process is instantiated for each media file submitted to the system. Thus, the media file is the token that navigates through the process.

To design the process, we are using the Business Process Modeling Notation (BPMN) version 1.2, which is an Object Management Group (OMG)

standard, defining several visual elements to represent from simple to complex business process situations [77]. Process activities are distributed in swimlanes, one for each role involved in the process. In this particular case, roles are used to indicate each particular stage where data is manipulated. They are: user, application client and server. Users are those people that consume and work with media using application clients. These clients are applications running in computers different from the ones where the framework is running, using the network to interchange data with the server. The server hosts the framework and the repositories where media, segments and annotations are stored and it manages connections from all clients.

The rectangles are activities, describing the steps of the process, interconnected by links that define the activities' sequence of execution. Links between activities in the same swimlane are continuous and in different swimlanes are dashed. The process starts and ends with circles representing the starting point and the ending point. When an activity has a '+' sign on the bottom it means that it is more complex and more details about it are described in a sub-process.

We describe a workflow that takes into consideration six macro-activities, describing globally the information flow through the system. Afterwards, we go into details, exploring lower lever activities and the roles to perform them. The macro-activities are:

- **Processing:** the processing activity aims to elicit all the data that is possible to extract from the media file, such as size, dimensions, resolution, histogram, frame rate, and other values that might be used as parameters for algorithms and to complement annotations. It also collects data from file descriptors wrapped with the media file and calculates the checksum of the file, using the MD5 hash algorithm [1], and compares the resulting 32 bits key with existing keys to avoid duplicity. Data collected here are associated to the default media segment as described in the section 3.7.
- **Segmentation:** at this point, the media is available to be segmented, manually or automatically. Following the rules and features described in section 3.5, meaningful parts of the media are selected and linked to be then annotated. In case of automatic segmentation, the user can bundle recognition algorithms as a plugin and attach them to the application, to be executed during the processing activity. In case of manual segmentation, a rich user interface is offered to allow the creation of segments and links over the media.
- **Annotation:** the annotation activity is performed almost at the same time as the segmentation. For each segment or link created, one or more annotation techniques can be applied, once again, automatically or manually. When manually, the expertise of the user may guide him/her to chose

the most appropriate annotation according to the nature and complexity of the content.

- **DRM:** it defines which privileges and constraints users need to fulfill to access multimedia contents, authenticating and authorizing users before allowing them to have access to service outputs, checking if the media can be reproduced in the user's current location, if his/her age is equal or greater to the minimum to visualize the content, and so on.
- **Querying:** querying considers the existing annotations to find meaningful information for the user. The better annotated is the media, the more precise are the results. However, precise results also depend on queries well built, rich in terms of representativity, and a more user friendly query interface. A query is built using an XML-like language and passed by parameter to a search service that performs a map reduced function to find segment candidates in the database.
- **Retrieval:** The output of the querying is a list of media candidates to be shown to the end-user. The retrieval is performed in 2 steps:
 1. the user receives a list of media that corresponds to the criteria of the query and also filtered with an access security check.
 2. the user selects one of the media to visualize and a second security check is performed to verify if the user has rights to play the media, considering constraints such as location, age, licensing, media quality and others.

Going into details in the workflow, Figure 3.8 refines the steps to add a media in the repository, considering processing, segmentation, annotation and DRM. The user of a Yasmim client opens a media file, which immediately triggers the file processing as previously described. Following, the end of the processing triggers any available automatic segmentation and annotation, both done at a client level. Manual segmentation and annotation might still be available for the user. They are performed together and eventually, the output of a manual segmentation may imply in an automatic annotation (i.e. the recognition is restricted to the area of the segment) or an automatic segmentation may imply in a manual annotation (i.e. some features were detected and delimited, but the meaning of the content is added manually).

It is important to mention that every resource has, at least, a segment representing its full content. This large segment is useful to associate annotations to the media as a whole. For example, the title of a movie would be a typical annotation related to the whole content. Other characteristics such as file size, dimension, resolution, etc., are not restricted to some specific part, but to the entire media. This segment is created in the process to add a resource to the repository. However, which kind of segment should be created for each kind of resource? Table 3.1 is an important guide for the selection of

the appropriate segment. In the case of an audio file, for example, a temporal segment with the full duration of the content is created; the same for a video file; if the file is an image, a spatial segment has the same dimension of this image; and so on.

The media and its related segments become available for the right management, which can add restrictions, such as, what age the audience should have, or from which location they can have access, or what is the maximum media quality one can experience, etc. Those restrictions are represented in a meta-data format, as well as segments, links and annotations. They are uploaded to the server, together with the media file, by the client application.

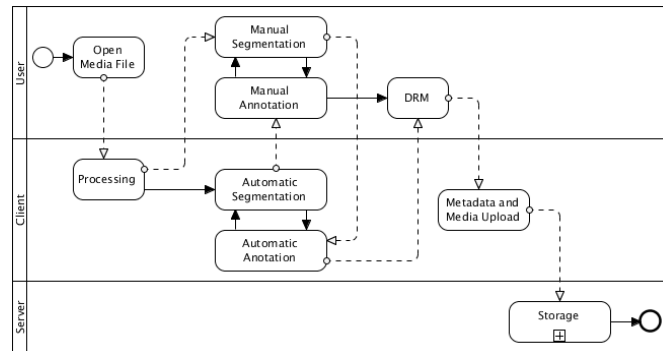


Fig. 3.8 Multimedia archiving workflow

On the server side, the storage process is performed to handle data from the client. This process is shown in Figure 3.9. If the media file is sent to the server, it is stored in a temporary place until the final commit of the process, when everything is persisted. The first step is to check if the file already exists. For that, a 32 bits checksum of the file is compared with the checksum of existing files. It minimizes the risk of content duplicity, but it can not be completely avoided because the comparison is made in a bit level. In a higher level it would be very time consuming. If the file is valid, the process continues through two parallel activities: 1) the extraction of a sample to be displayed in search results (retrieval) and 2) the persistence of the segmentation data, which also includes links. The extraction of a sample may take some time, meanwhile, annotations and DRM data are also persisted.

Figure 3.10 depicts the process when searching for multimedia contents. It starts with the user building a query. A query could be from a simple keyword to a complex logical sentence. A service on the server receives the query and translates it to a native query supported by the database. The database then returns a list of candidates and sends it to the client. If needed, the client can have access to the media samples using web links available in

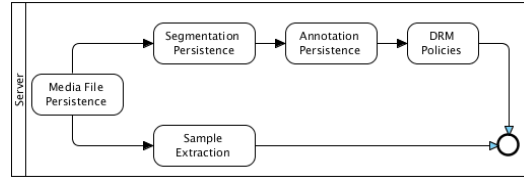


Fig. 3.9 Multimedia archiving workflow

the list's elements. The user makes a selection of desired media and makes a new request to the server to obtain them. The selected media are retrieved to the client and become available for reproduction, together with existing segmentation links and annotation data.

Retrieved media are immutable, thus they cannot be edited by a client. Any modification would break the original principle of multimedia archiving, which is to preserve media. If a media file is edited, it becomes a different resource because the introduced changes might have left all segments and annotations obsolete, as a consequence of a content shift, modification of meanings, properties, etc. The fact that media can not be modified by the client, does not mean that the server can not send a modified version to the client. Due to accessibility or security constraints, the server can send a modified version as long as it also preserves the original version in a safe place. On the other hand, segments and annotations can be added, edited, and removed, since they do not have an impact on the media content. Therefore, the same segmentation and annotation mechanisms used to add new media are also applied at this point, including the DRM features. This time, only metadata are uploaded to the server because the media is already on the server. The storage process is also performed from here.

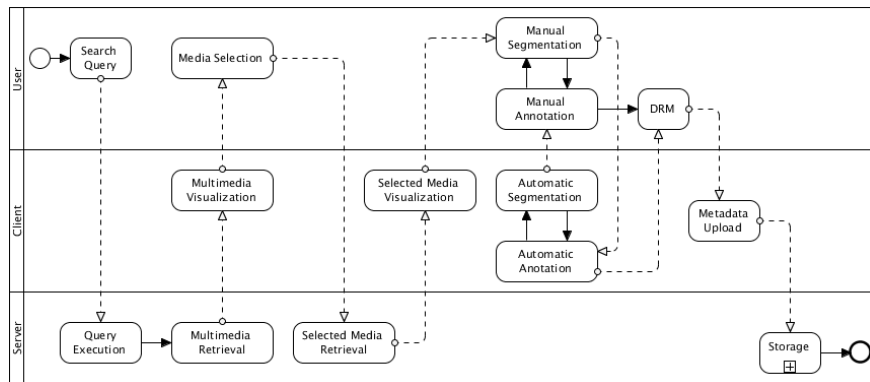


Fig. 3.10 Multimedia archiving workflow

3.10 Implementation

Yasmim was designed to offer a rich description of media, attaching semantics to contents of images, videos, audios, and 3D models. Using Yasmim, developers and researchers do not deal with the usual complexity of managing media contents. An API to perform operations over those media is made available through web services. Therefore, instead of developing one more multimedia archiving system, we are contributing by simplifying the way multimedia is added to existing applications and making distributed multimedia management accessible for non-specialized developers.

The designed architecture aims to provide scalability, extensibility, and robustness. Scalable because it is stateless (i.e. it does not save any state or temporary data, such as user's sessions, navigation, etc) and uses a peer-based distributed database with bi-directional replication (i.e. several databases interchanging data from the same application, generating several copies of the same data to attend more clients), allowing dynamic addition of new server nodes as the demand increases. Extensible because several existing solutions can be used with a minimal integration effort. Robustness because the chosen technologies are extensively applied on many other solutions, they have years of existence and large communities around them.

Yasmim runs entirely on the server. Its user interface is essentially administrative (see Figure 3.11). In order to access and maintain media, application clients should be developed to access the server. The communication is made through web services, using Internet protocols. They are also stateless, thus any temporal data should be managed by the client and sent to the server when necessary. Being stateless allows the system to dedicate all its memory to process media. It also simplifies the use of several machines because it is not necessary to synchronize temporary data between them.

Figure 3.12 depicts the general architecture of a system using Yasmim for multimedia archiving. Yasmim is right on the middle, intermediating data between several data sources and several clients. The middleware behind it is an application server called Glassfish [16]. It can manage instances of the same application on spread machines, expanding the processing capability according to users' demands. The application manages the information that comes from clients and organizes them in different databases. Each database was chosen according to the data that they are most appropriate to store. These databases are:

- *Media File Repository*: Media files are stored directly in the file system. The optimal efficiency on file access depends on the operating system and the storage system in use.
- *Annotation Database*: Annotations are stored in a document database server, which processes text more efficiently than relational databases.

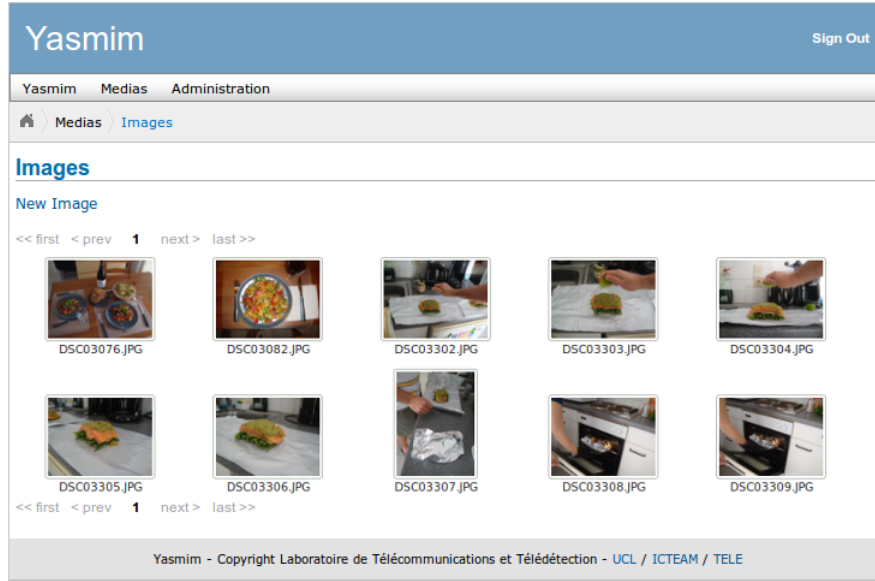


Fig. 3.11 Yasmim's administration user interface

- *Segmentation and Indexation Database*: A relational database is used to store references to files in the repository because of its robust indexation mechanisms. It is also used to save segmentation data because tables have better support to store and retrieve numbers.

The role of the client side is to process heavy operations, such as the support for several modalities, automatic segmentation, automatic extraction of meanings, and also to provide rich user interaction for intuitive manual segmentation and annotation. The data is synchronized with the server, making the media and all related data available for searching and sharing.

Figure 3.13 shows the server side where Yasmim is running. This infrastructure is compatible with several kinds of clients developed in different languages, platforms and devices. It provides REST web services [31], a style of software architecture for distributed hypermedia, where the most important aspect is the addressability of resources. Every media and related information are reachable through a unique identifier. This identifier follows the URI (Uniform Resource Identifier) standard¹, which is used by the HTTP protocol to locate resources on the web. With a REST-based framework, we could attach segments and annotations to media, making slight modifications on the URI. This way, not only search mechanisms can benefit from the media description, but many other practical applications as well, since REST web

¹ <http://www.w3.org/TR/uri-clarification/>

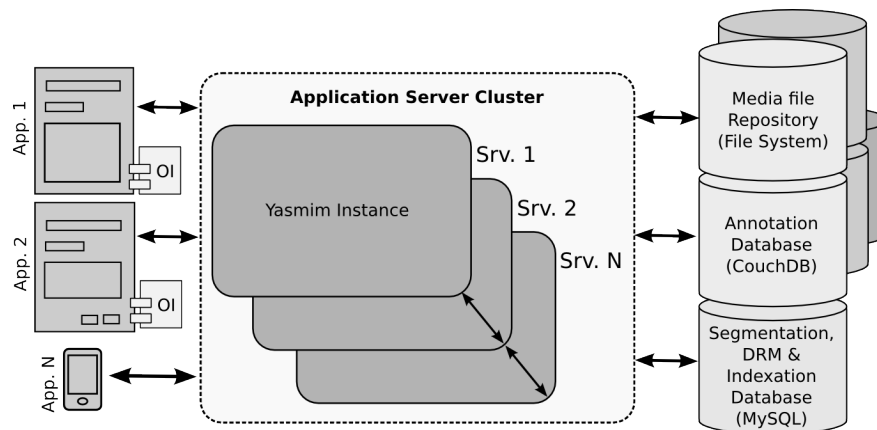


Fig. 3.12 Architecture

services are easily accessible by any socket library. The key abstraction of information and data in REST is a *resource*, which is aligned with how we define media (see figure 3.2), thus every media has its unique URI, and every related information is obtained extending or parameterizing the media's URI.

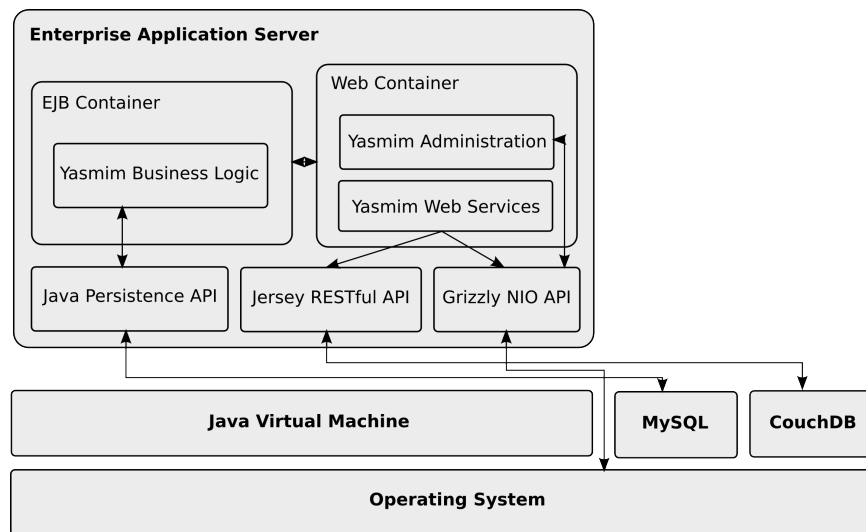


Fig. 3.13 Architecture

The application server has two execution environments: The EJB Container and the Web Container. The first one is appropriate to handle transactional data, which is suitable for operations with the relational databases.

The second one is appropriate to handle and generate content based on the target users. This content is stored and retrieved by the EJB Container and other sources and appropriately transformed for different purposes. The EJB Container runs Yasmim's business logic that takes care of the database data. The Web Container runs Yasmim's administration, which is the user interface to perform back office (maintenance) operations, and the web services, which are the interface with clients.

These containers have some facilities to access data. The *JavaTMPersistence API* is a JavaTM specification for relational data access [85]. It is capable of mapping table with JavaTM classes in order to transform table tuples in JavaTM objects, consequently simplifying data access. The *Jersey RESTful API* is an implementation of the JavaTM specification for providing and consuming REST web services [84]. It is used to implement web services for clients and to consume web services of the document database server. The *Grizzly NIO API* is an implementation of the New JavaTMIO specification [83]. It is used to store, retrieve and modify media files asynchronously, optimizing parallelism.

The platform is composed of the JavaTM Virtual Machine (JVM), MySQL database, and CouchDB document database. They run on top of the operating system, which is also responsible for the media storage. The JVM is responsible for the execution of JavaTM programs, which includes the application server and the applications running on it [17]. MySQL is one of the fastest relational database available on the market [18]. Its indexation, relational and transactional features are essential to deal with a large amount of numbers and unique references, that should be consistent. Finally, we are using CouchDB, a document based database, to store annotations because it is more efficient for text processing [32]. It is considered as a NoSQL database [26], a term used to represent a new generation of non-relational databases, distributed, and horizontally scalable. Because annotations perform an important role in this research, it should be stored in a high scalable way. All data in CouchDB is accessible by REST web services, allowing clients to access it (read-only) directly, without Yasmim mediation, although only Yasmim can write data there.

3.11 Client Implementations

The decision to clearly define the role of the server and the client has an important impact on the overall performance. While the server is focused on organizing data provided by several clients, the client is concerned with the user experience, specific needs of media manipulation and the problem in itself. In other words, clients are domain and context oriented and Yasmim

is multi-purpose and scalable to support clients' demands. In the following subsections, we present some existing clients and other potential applications for Yasmim.

3.11.1 *Reference Client Implementation*

In order to help developers on the implementation of their clients to support Yasmim's services, perform exploratory tests and test the services in situations of real use, a reference client implementation was developed. This implementation tries to adopt the best practices in software development and spread good quality clients to Yasmim. Instructions about how to access the reference implementation are available in appendix B.

Figure 3.14 depicts a screenshot of the image annotation tool, illustrating how an annotation is created. All data presented and collected are not stored locally, but transmitted through web services between the client and the server.

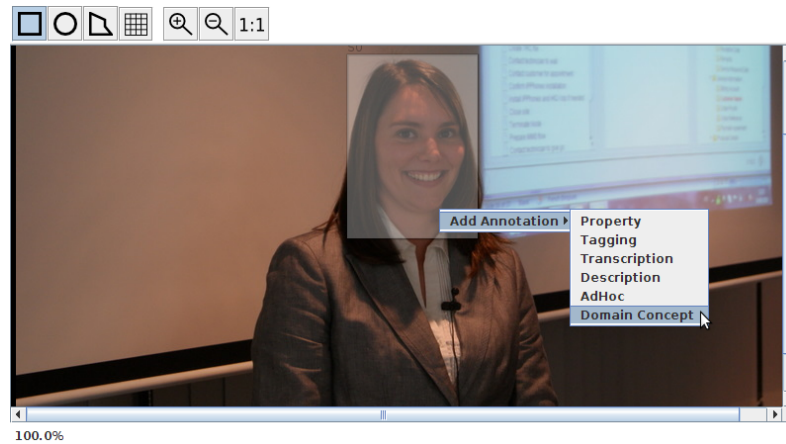


Fig. 3.14 Yasmim reference client implementation

3.11.2 *IRMA Project Clients*

During the course of the IRMA Project [59], some clients for Yasmim were developed, each one for a different purpose. Figures 3.15, 3.16, and 3.17 depict

the user interface of a client with the purpose of navigating through Yasmim's repository. Initially, the user creates a query using a friendly user interface, submits the query to the server, gets a set of media candidates and finally sees more details of the selected media.

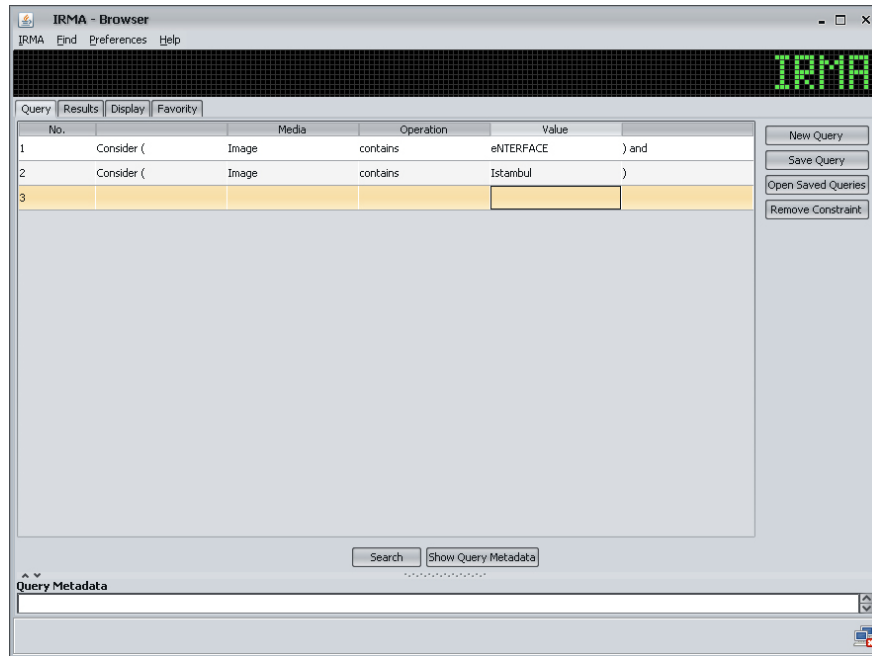


Fig. 3.15 IRMA query interface

The query interface is particularly interesting because it constructs first order logical expressions that are consistent with supported annotation types. For example, in order to get:

- all media containing a yellow car, the expression would be:

```
[all media] [contains] 'yellow car'
```

- sport news aged less than 3 months:

```
[video.date] [is greater than] '30/01/2007' and  
[video] [contains] 'sport news' or  
[audio.date] [is greater than] '30/01/2007' and  
[audio] [contains] 'sport news'
```

- Medical records of user 'Paulo'

```
[image] [contains] 'Paulo':
```

- Occurrences of license plate XYZ-789:

```
[video] [contains] 'XYZ-789' and
[audio] [contains] 'XYZ-789'
```

The user interface contains columns with a set of fixed values. The first column indicates the kind of media the user intends to find. The second column contains logical operations as listed below:

- *contains* : check if there are annotations containing a specific term associated with media files.
- *does not contain* : it is the opposite of *contains*. It is not recommended to put this constraint alone, but as the complement of another one.
- *is* : the informed constraint is a direct reference to a specific content. The result is more precise because the research is also semantic.
- *is not* : it is the opposite of *is*. As well as *does not contain*, it is not recommended to put this constraint alone.
- *begins with* : it retrieves all segments where the annotation property starts with the constraint.
- *ends with* : it retrieves all segments where the annotation property finishes with the constraint.
- *is greater than* : it retrieves all segments where a specific quantitative property of the resource is greater than the informed constraint.
- *is less than* : it retrieves all segments where a specific quantitative property of the resource is less than the informed constraint.
- *is similar* : it retrieves all segments with a certain degree of similarity, comparing their properties.
- *is not similar* : it retrieves all segments under a certain degree of similarity, comparing their properties.

Another partner of the project, FPMs, developed a speech recognition software, depicted in Figure 3.18. According to the report of the project [4], this software gets audio files containing radio news and transcribes the full speech of the news anchor into text. Parts where the anchor speaks are delimited by segments and full transcription annotations are associated with their respective segments. This software used the initial version of Yasmim, still at the beginning of the IRMA project, and did not access the framework using web services, but using JavaTM Remote Method Invocation (RMI).

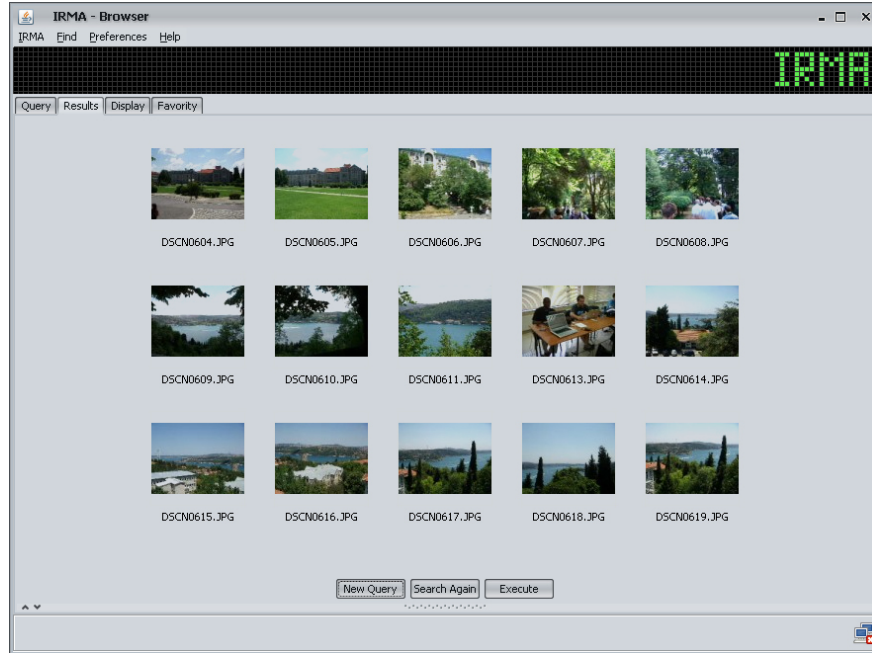


Fig. 3.16 IRMA retrieval interface

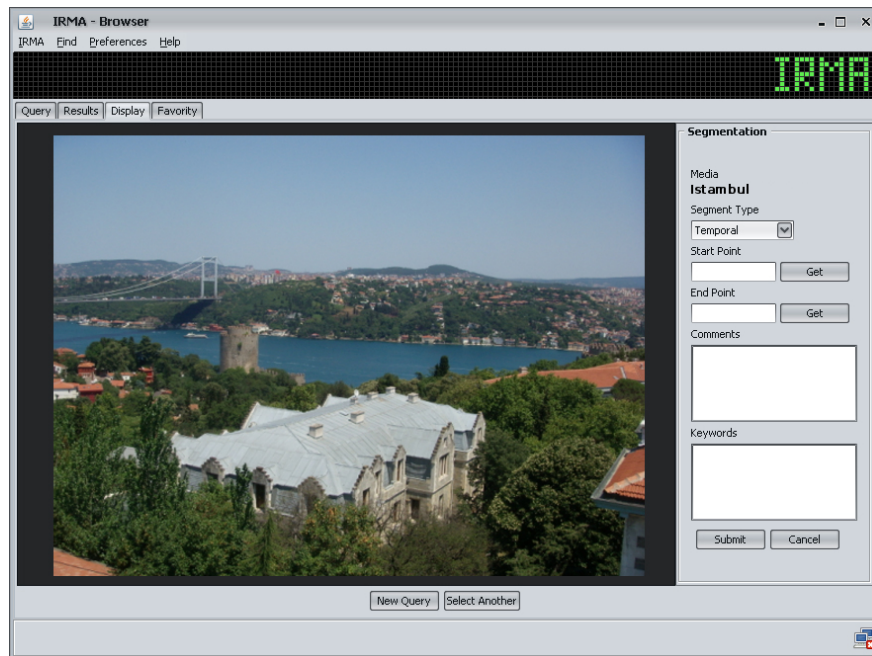


Fig. 3.17 IRMA annotation interface

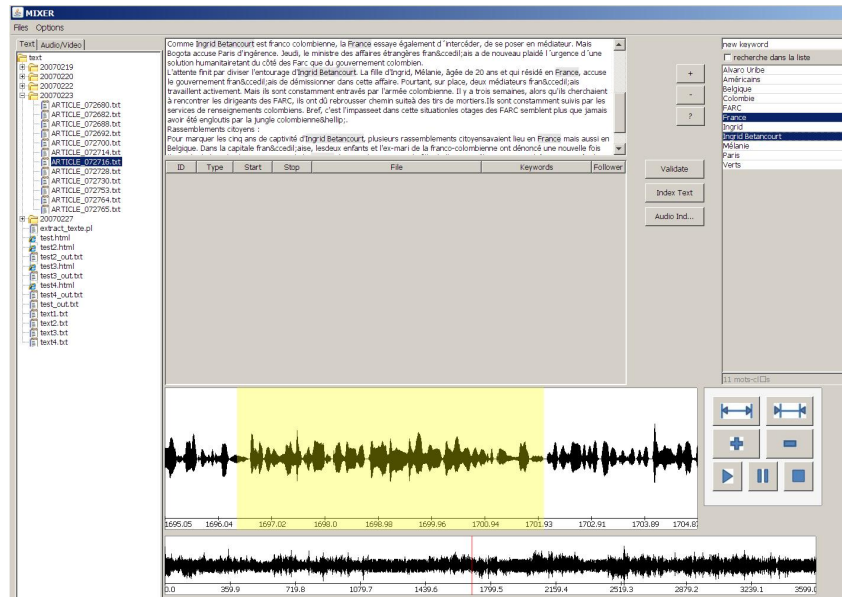


Fig. 3.18 Speech Transcription Tool

3.11.3 MedicalStudio

MedicalStudio is a component-oriented platform designed to ease the creation of medical imaging workstations [109]. Besides simplifying the work of developers, MedicalStudio also streamlines clinical trials and end-user's experience. The platform provides a collection of reusable components that can be assembled to produce new applications considering image processing, data access, interaction design and others. Each assembly of components will produce a different application that may target different uses. In the case of an image registration application, for example, each algorithm will be seen as a component, and there will be several UI components to meet different user profiles:

- a configuration UI for tuning algorithms used by engineers;
- another configuration UI for tuning options oriented for clinical researchers;
- a visualization UI for doctors to perform their specific clinical diagnosis.

Figure 3.19 depicts MedicalStudio's user interface, where an image of a mammography is shown. Looking at the image, the doctor can visually identify micro-calcifications, select them using spatial segments and annotate these segments using domain-specific annotations, which is well known by the doctor, who is a specialist on the area.

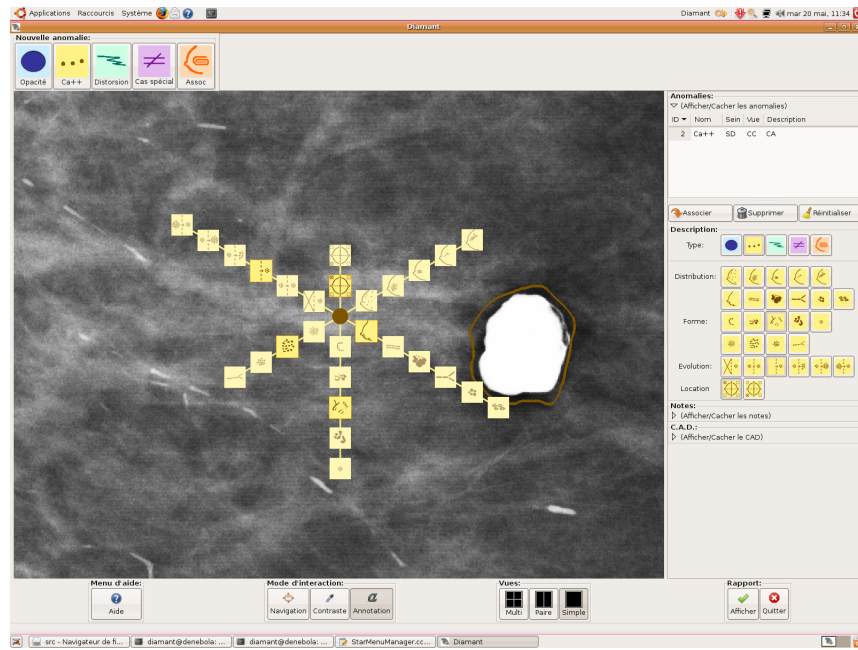


Fig. 3.19 MedicalStudio running components for mammography

The framework is entirely written in C++ and relies on well accepted and powerful libraries, such as VTK² (Visualization Toolkit) for visualization, ITK³ (Insight Segmentation and Registration Toolkit) for image segmentation and registration, DCMTK⁴ for DICOM⁵ (Digital Imaging and Communications in Medicine) interoperability and GTKmm⁶ for graphical user interface. These libraries are not always enough in specific cases, so that list is not fixed and the architecture is flexible enough to allow interoperability with any other toolkit as far as it can be bound with C++.

It is part of MedicalStudio's plans to be integrated with Yasmim in a short term perspective, since both technologies were developed in the Laboratoire de Télécommunications et de Télédétection (TELE). MedicalStudio supports segmentation and annotation, but they are stored in a DICOM format, which is not supported by Yasmim because it was developed for very specific needs on the medical domain. In order to support DICOM, Yasmim's data model should be carefully compared with DICOM's data model. If needed, modifica-

² <http://www.vtk.org>

³ <http://www.itk.org>

⁴ <http://dicom.offis.de/dcmtk.php.en>

⁵ <http://dicom.offis.de>

⁶ <http://www.gtkmm.org>

tions should be incorporated in Yasmim's model, since DICOM is a standard and it cannot be easily modified. Therefore, the integration of MedicalStudio and Yasmim depends on how compliant Yasmim's model is to DICOM.

After the compatibility check, MedicalStudio would stop storing in a DICOM format and start storing in Yasmim. The images and 3D models would be stored in the media repository, segments in the relational database, and annotations in the document database. An additional web service would be developed to load all this data and generate a DICOM file on demand. The resulting file can then be distributed to other medical systems. The algorithm used by MedicalStudio to manage this format could be reused on the implementation of this web service and posteriorly removed from MedicalStudio after its integration with the new web service. There are two advantages on this approach: 1) MedicalStudio would become less complex by migrating part of its source code to Yasmim, consequently reducing the maintenance cost; and 2) other medical applications would profit from the new web service, by simply reusing it to generate their DICOM files.

The current MedicalStudio version shows that it implements one kind of segmentation, the spatial one, and three kinds of annotations, which are: 1) *property*: used to annotate low level features of the image; 2) *description*: if necessary, some description of the segment can be added; and 3) *domain concept*: the most common annotation, since MedicalStudio implements an ontology of breast medical terms that are represented by a contextual menu where the doctor can select the more appropriate concept for the selected segment.

Once integrated with Yasmim, MedicalStudio has another immediate benefit, which is the possibility to support multiple domains. It would allow specialists from different medical specialties to analyze the same image, enabling multi-disciplinary diagnosis. It would also allow the annotation of other kinds of images, besides mammographies.

3.11.4 Multimodal Clients

A natural tendency on the use of multimedia archiving system is the implementation of new kinds of user interactions. The amount of information present in multimedia files demands enhanced user experience and such need calls for multimodal interactions in order to provide intuitive ways of controlling the system, allowing expressiveness beyond typing (i.e. keyboard) and pointing (i.e. mouse).

For performance reasons, the support for multi-modalities is kept out of the Yasmim framework and implemented on the client side, where the com-

putation capacity can be fully allocated for this purpose and has no impact on the network bandwidth. Also, the need for multiple devices makes the adoption of modalities complex to implement and distribute. Fortunately, an extensible research has been developed on the direction of simplifying the architecture of multimodal systems. The result of this research is the OpenInterface (OI) Kernel and Worbench [50], a high-fidelity prototyping solution to make multimodal interaction reachable for most application developers. These developers benefit from the plethora of existing signal-processing algorithms (e.g. gesture recognition, face tracking, sound source localization, input devices, etc.), reusing them to ensure seamless connections between heterogeneous native components, assuring real-time (at the interaction point of view) performance, among other benefits.

Chapter 5 describes a case study where fusion of modalities, handled by OpenInterface, is implemented with the support of the Yasmim Framework to store content collected, segmented and annotated by OI components, and to retrieve semantics that show similarities between different signal channels. Besides the ones previously described, this is one more case of a Yasmim's client with a different purpose of the usual multimedia archiving implementations.

3.12 Conclusion

We have presented a multimedia archiving framework that supports several media formats, three types of segmentation, described by six types of annotation and supporting links between segments that can also be annotated. The framework is compatible with a formal workflow oriented for audiovisual companies and has a scalable, extensible and robust architecture to support large datasets and reuse existing modality implementations.

In order to position Yasmim in relation to other available options, we have considered the three most relevant systems presented in the state of the art (chp. 2) and compared them with Yasmim, using the same methodology described in section 2.7. These systems are MPEG-7 MMDB, SemaPlorer and UCM. They are not fully considered as a framework because they were not fundamentally designed to be like one, despite being able to run customized code and accessible by remote clients, which is the case of MPEG-7 MMDB. However, they are comparable with a pure framework like Yasmim because they comply with the set of requirements that should be implemented, as presented in section 2.2. Therefore, this comparison is more positioned in a level of requirements than in the system as a whole.

Figure 3.20 depicts a compacted version of the matrix presented in Figure 2.8, including Yasmim in the comparison, which is located in the darker

line. Some columns are highlighted when Yasmim implements the respective requirement, simplifying the comparison with other tools.

	Multiple Media	DRM	Distributed	Segmentation	Annotation	Features Recognition	Personalization	Multi-domain
MPEG-7 MMDB	+	+-	+-	+	+	+-	-	+
SemaPlorer	+-	-	+-	-	+	+-	-	+-
UCM	+	+-	+	+	+	-	+	+
Yasmim	+	+	+	+	+	-	-	+

Fig. 3.20 Comparison of approaches considering Yasmim

From those eight requirements, Yasmim implemented six, which are: support for multiple media, DRM, distributed, segmentation, annotation, and multi-domain. Going into details in each requirement:

- **Multiple Media:** it is able to segment and annotate images, audios, animations, videos and 3D models. Within these media forms, there are also dozens of formats and the framework can support most of them, inheriting this capability from the JavaTM platform.
- **DRM:** basic implementation of the right management, authenticating users every time a service is invoked, checking if the user has rights to invoke the service, checking if the media can be reproduced in the user's current location, if his/her age is equal or greater to the minimum allowed to visualize the content, and so on.
- **Distributed:** it provides REST web services widely available on the network that are accessible for clients developed in different languages, platforms and devices. In essence, every technology that has support for communication using sockets can access Yasmim's data and perform operations on the server, characterizing a modern approach of distribution, aligned with existing web architectures.

- **Segmentation:** it implements spatial, temporal and spatio-temporal segments as vectorial structures, making it applicable not only for bitmap content but also for vectorial contents, such as graphics, animations and 3D. In addition, segments can be mutually linked, creating a graph of connections and filling the gap of meanings floating in the limbo not segmented.
- **Annotation:** it provides a greater coverage of the descriptive formats that can be given to segments, which are: property, tagging, transcription, description, adhoc and domain concepts.
- **Multi-domain:** it allows the use of multiple ontologies to annotate segments and links. While observing the media, the user decides which ontology is more appropriate for the case, considering his/her domain of expertise.

Feature Recognition and personalization were not yet considered by Yasmim as a matter of priority. This framework transfers the responsibility of feature recognition for the clients because this processing can be costly, thus impacting the performance of the system as a whole if done in the server (centralized). On the other hand, it would be very necessary for mobile devices, which does not have enough processing power. Or some recognition made on the server can be saved and retrieved faster the next time it is demanded. About personalization, it demands large experimentations with real end-users, studying their interaction with clients in order to build an effective user profile. It can be perfectly done as a future work, as the adoption of Yasmim increases. In addition, the DRM module is limited in terms of contribution because it was incorporated from third-party contributions.

From Figure 3.20, none of the tools is comfortable with feature recognition. MPEG-7 MMDB can run it only if third-party developers implement it and attach it as a plug-in. MPEG-7 MMDB does not distribute these plug-ins by default. SemaPlorer is able to recognize some information in the context, but aided by additional metadata retrieved from the server. These implementations are not robust enough to assert that this category of system has evolved in this field. About personalization, only UCM is committed to that. It supports user-control and personalization within multimedia presentations. On the other hand, it lacks on the security part, since it may limit user's freedom in a first moment.

In the requirements that all are competing for, such as multiple-media support, distribution, annotation and multi-domain, each technology has its own goals, thus the implementation of these requirements are naturally different. What is important to emphasize are the trends that these kinds of technologies are following and how the newest platforms and systems are allowing the evolution of multimedia solutions.

Chapter 4

Segmentation and Annotation of User Interfaces

Chapters 4 and 5 describe application scenarios of Yasmim on two distinct problems, which demand multimedia management support as a potential solution. These chapters aim to validate Yasmim, as described on chapter 3. This chapter 4 describes the use of segments and annotations to build user interaction models. These models help to describe the elements of the user interface, the navigation between screens and the flow of interaction through UI elements.

4.1 Introduction

Nowadays, we are constantly interacting with computers, through their UIs, to work, create, communicate, and many other daily tasks. The quality of these UIs directly impacts the efficiency of the task to be done. Oddly, this argument is not strong enough to convince software developers not to design by conventions, but by user needs. This culture is hard to be changed because the demand for new software or software change is higher than the offer of skilled professionals for this work. Thus, the deserved importance to the user comes when the user interface becomes very cumbersome to work, usually when the software is already in production.

Considering this fact, the investigation of the user experience when interacting with software in production seems to be more appropriate for a world in constant transformation. A common approach to analyze the interaction is observing users in their work environment [72]. Although being efficient to identify usability problems, user observation is relatively expensive due to:

- allocation of specialized professionals;

- demand of end-users' time to perform experiments, attend meetings, and other tasks;
- small samples of end-users, not representative to typify the whole organization; and
- limited period of time on which the evaluation remains up-to-date, since it does not cover future releases.

The use of UI patterns [106], based on a rich previous experience of UI designing, might considerably reduce the occurrence of usability problems, but does not completely neutralize them. The dynamics of information technology (IT) offers new kinds of interaction techniques very often. Thus, making the simple adoption of a multi-touch device a motivation to rethink established patterns. Multiple devices and heterogeneous platforms also contribute to make a UI patterns system very complex to manage and to adopt. Therefore, how would an efficient way to improve the user experience be in a pragmatic world where specialized professionals are limited and the cost of traditional usability analysis might be prohibitive and non effective? The answer might be in a cultural change that is going on due to a social revolution promoted through the Internet. This change is increasing the interest of people to interact with other people through social networks, such as Facebook, YouTube, Wikipedia and others, by producing or sharing content according to the user judgment of other people's interests. If we get advantage of this cultural change to constantly motivate people to share their daily experiences when interacting with UIs, by reporting errors, giving suggestions, following contributions of other users, we would have a permanent evaluation process made by those who actually perform the job. These social interactions produce data that can help system analysts and designers make consistent decisions concerning what should be prioritized, how to change according to the users' point of view, what is the impact of each change and how rigorous the tests should be according to the frequency of use, and many other conclusions.

We aim to propose a solution for user feedback enhanced by a multimedia archiving framework capable of organizing information based on a UI representation model. The combination of a UI model with a multimedia archiving is intended to maximize the feedback representation and minimize the time to share it. This chapter explains how this combination occurs and which benefits we can get from it. We initially give an overview of the feedback system, explaining how all pieces are integrated. Then, we present our approach of user interface decomposition and how it can be applied on several applications independent of the platform and of the adopted technology, while underlining existing approaches. Afterwards, we enter into details about the multimedia archiving framework that supports segmentation and annotation of several media, but limiting to images, which is suitable for this particular case. Finally, we present some relevant use cases, discuss them and conclude with a qualitative analysis and discussions on future works.

4.2 User Feedback Steps

As software is so error prone, there are several techniques to minimize failures and produce the best software possible. With that purpose, exploratory testing [118] is a challenging and successful strategy that supports testers with guidance in order to discover hidden bugs, usability issues, security concerns, etc. The main goal of such techniques is to achieve higher-quality testing and higher-quality software, as well.

There are normally two groups of people who find issues in software applications: the testers and the users. All these challenging techniques are aimed for testers so they can find errors before the application is delivered to customers and used by end-users in their real environment. However, users normally find several problems when they are using software to get their work done. This is common because testers simply cannot simulate all situations of usage to find and prevent such problems. This becomes even more complex when we consider the variety of scenarios that include different ways of interacting, devices being used, infra-structure, etc. This is a typical scenario that shows the importance of getting users involved in testing, that is, informing which problems they have faced while doing real work with real data in real environments.

Atividades

Atividades a serem executadas no projeto.

Organização: Assessoria Executiva

Projeto: Implantação do Escritório de Gestão de Projetos na VTI

Mais Informações.

de 02/04/2007

até 14/09/2007

Ordenar por: Personalizado

Pesquisar

Nova Abordagem: Gráfico de Gantt

Atividades

Nº Prodec.

Duração

Unid.

Início

Término

Concluído

GERENCIAMENTO

39

257

dia(s)

02/04/2007

14/09/2007

75%

TERMO DE ABERTURA DO PROJETO

41

1

dia(s)

02/04/2007

02/04/2007

100%

DECLARAÇÃO DO ESCOPO DO PROJETO

42

41-TI

2

dia(s)

02/04/2007

03/04/2007

100%

EAP - ESTRUTURA ANALÍTICA DO PROJETO

40

42-TI

1

dia(s)

02/04/2007

02/04/2007

100%

DICIONÁRIO DA EAP

44

40-B

5

dia(s)

02/04/2007

06/04/2007

100%

CRONOGRAMA DO PROJETO

43

40-TI

3

dia(s)

02/04/2007

04/04/2007

100%

PLANO DE GERENCIAMENTO DO PROJETO

45

40-TI, 42-TI

5

dia(s)

02/04/2007

06/04/2007

100%

RELATÓRIOS DE ACOMPANHAMENTO DO PROJETO

46

36-TI

120

dia(s)

02/04/2007

14/09/2007

0%

RELATÓRIOS DE ACOMPANHAMENTO DO PROJETO

47

36-TI

120

dia(s)

02/04/2007

14/09/2007

0%

FASE INICIAL

35

34

dia(s)

02/04/2007

11/05/2007

100%

DEFINIÇÃO DE CONSULTORIA EM GESTÃO DE PROJETOS

37

41-TI

3

dia(s)

02/04/2007

04/04/2007

100%

RECRUTAMENTO DA EQUIPE DO EGP

38

45-TI

30

dia(s)

02/04/2007

11/05/2007

100%

REUNIÃO INICIAL DO PROJETO

39

45-TI

1

dia(s)

02/04/2007

02/04/2007

100%

FASES DE DESENVOLVIMENTO

6

20

dia(s)

02/04/2007

02/04/2007

0%

E1: CONSULTORIA EM GESTÃO DE PROJETOS

23

3

dia(s)

02/04/2007

02/04/2007

0%

PAC01: CONTRATAÇÃO

26

1

dia(s)

02/04/2007

02/04/2007

0%

PAC02: DEFINIÇÃO DOS PROCESSOS DE GP

25

1

dia(s)

02/04/2007

02/04/2007

0%

PAC03: CAPACITAÇÃO INTERNA

24

1

dia(s)

02/04/2007

02/04/2007

0%

1 usuário on-line.

Fig. 4.1 Screen shot of a report annotated by the end-user

To illustrate this, we depict a real example in Figures 4.1 and 4.2 from a software development project, where users naturally created segments using screen shots of the user interfaces to give feedback and list the issues they faced when using the system. This example represents a suitable scenario

Fig. 4.2 Screen shot of a form annotated by the end-user

where the multimedia archiving framework is useful to support this kind of problem.

The whole point is to get a rich and detailed feedback from the user without making it a complex and time consuming task. The steps are divided in: *back office*, which includes people who act as supporters of the business operations; and *front office*, which includes those people who perform business operations, including those with direct contact with customers. The tasks are:

1. Back office

- a. Take screen shots of existing UIs and store them in the media repository. Using screen shots, any application of the organization is eligible to receive feedback from the end-user, independent of their technology, platform, or if the source code is available or not.
- b. Select relevant elements of those screens using image segmentation. A segment assumes the form of any shape that delimits useful information in the image. The definition of this information relies on the interpretation of the one who is segmenting it. Some rules of segmentation are needed, and they will be discussed in future works, but some level of freedom is also important not to limit expressiveness.
- c. Annotate those segments with properties and semantic data that are relevant to describe and to find them afterwards. Since segments do not contain descriptive properties, annotations fulfill this role and act as an index, retrieving data according to the user's criteria.

2. Front office

- a. Interact with applications to perform daily tasks. These applications can be of any kind, from text editors to complex Enterprise Resource Planning (ERP) systems, and end-users are everywhere in the organization, connected to the network.
- b. When an end-user faces a particular situation in a UI, from a confusing feature to a serious bug, he/she searches for that particular UI in the media repository, providing keywords related to the screen. If the screen is not found or it is not actually available, the end-user can take a screen shot himself/herself and upload it to the feedback application.
- c. Select one of the existing segments where the situation occurs or create segments according to his/her understanding of the case. The front office segmentation works exactly as the back office segmentation. However, at this point, no segments created by end-users are accessible to other end-users because we can not assure that these segments will be created according to the adopted UI model, bringing some implications for the understanding and visualization of segments.
- d. Add annotations to the segment that delimits the area where the situation occurs. In essence, the annotations are directly related to the situation, representing it semantically. Annotations made in existing segments are widely available for other end-users, who can collaborate in many different ways with their particular experiences.

4.3 The Abstract User Interface Model

A User Interface is a means by which users interact with the system as they perform their tasks, called UI tasks. Designing user interfaces is a technical process in which UI designers use UI toolkits, interactive patterns and different technologies to provide the requested requirements. At the same time, it is also a creative process in which UI designers organize components of the UI in ways that support an easy understanding, usage and navigation, contributing with a good user experience. All efforts on the UI creation are subject to user acceptance, which is subjective and dependent of the context. Variables, such as the users' profile, context, goal, available technology, impact the perception of the user experience.

The Human-Computer Interaction (HCI) field permanently evolves studying usability problems and at the basis of these studies are models and languages aiming to formalize the subjectivity behind the interaction with computers. This formalization intends to position computers as allies on the overall quality of user interfaces, supporting techniques of user interface evaluation, validation, simulation, generation, and others. However, one of the

main challenges of these models is to follow the fast technological evolution of devices and new kinds of interactions. On the web, for instance, new components developed from scratch and out of traditional toolkits arise constantly. New devices, such as iPod Touch and modern mobile phones, have been breaking paradigms and, apparently, no modeling language is evolving so rapidly to consider such a large variety of possibilities.

This problematic has been our major motivation to investigate which model is the most appropriate to cover so many possibilities. In a survey comparing existing UI modeling languages [102], the abstract UI model seems to be the one that covers the broadest possible range of targets for the broadest possible range of delivery contexts. We came up with an extension of the abstract UI model based on UsiXML, an XML-compliant markup language to describe UIs for multiple contexts of use [55]. This extension was proposed in [103] to link user interfaces with business processes.

The abstract UI model structure is comprised of broad UI components, which are components used for user interaction that start in broader levels and are further detailed until an atomic level. They are listed as follows:

- Screen group: a group of closely related screens and a container of a set of UI components, which are either in the set of screens, or in the set of screen groups, or in both;
- Screen: a state of the user interface when executing a UI task or part of a UI task and a container of a set of UI components, which are either in the set of screen elements, or in the set of screen fragments, or in both;
- Screen fragment: a portion of the user interface that is semantically related and a container of a set of UI components, which are either in the set of screen elements, or in the set of screen fragments, or in both;
- Screen element: any UI widget that serves as input and/or output for a UI task, it is the most atomic component.

Screen Groups do not have a visual representation because its function is to group screens related to an extensive task, which does not apply to this application. Screens, Screen Fragments and Screen Elements have their specific UI Component Types. Screens can be presented as pop-ups, dialog boxes, pages, among others. Screen Fragments can be represented as panels, menus, tabs, etc. Screen Elements can be depicted as fields, labels, buttons, links, combo box, list box, radio button, etc.

The organization of the UI model in these UI components makes it comprehensive enough to consider most types of UIs, even those with elements poorly specified or even ignored by UI representations in general, such as video, music player, 3D rendering, etc. For any new type of UI component, it is enough just to identify it as a new type in the existing set. They can be instantiated using one of the listed types by substituting the screen element

identified on the screen for a type of a screen element (e.g. link). Figure 4.3 depicts a form from a corporate application where it is possible to identify UI elements classified according to the model described. This screen is composed of screen fragments and each one has its own screen elements, illustrating how a common user interface can be annotated.

The broadest components are composable in nature, unlike the most atomic components. Table 4.1 depicts the composition and constraints of the UI components. Screen Groups are optional components, they may contain other screen groups and it must contain at least one screen. Screens are mandatory components, they may contain screen fragments and it must contain at least one screen element. Screen Fragments are optional components, they may contain screen fragments and it must contain at least one screen element. Screen Elements are mandatory atomic components.

The decomposition of the UI components adheres to the process of functional decomposition, which refers to decomposing a component in constituent components such that their joint complexity should be lower than the complexity of the original component. Since the decomposition can be applied recursively, the result is generally a hierarchy of concepts. For each concept in the hierarchy, there is a corresponding function that determines the dependency of that concept on its immediate descendants in the hierarchy [120]. This process is useful to gain insight into these constituent components, feasible in hierarchical and modular structures. Since UIs are arranged in a hierarchy, as classified in Table 4.1, and UI components can be put together

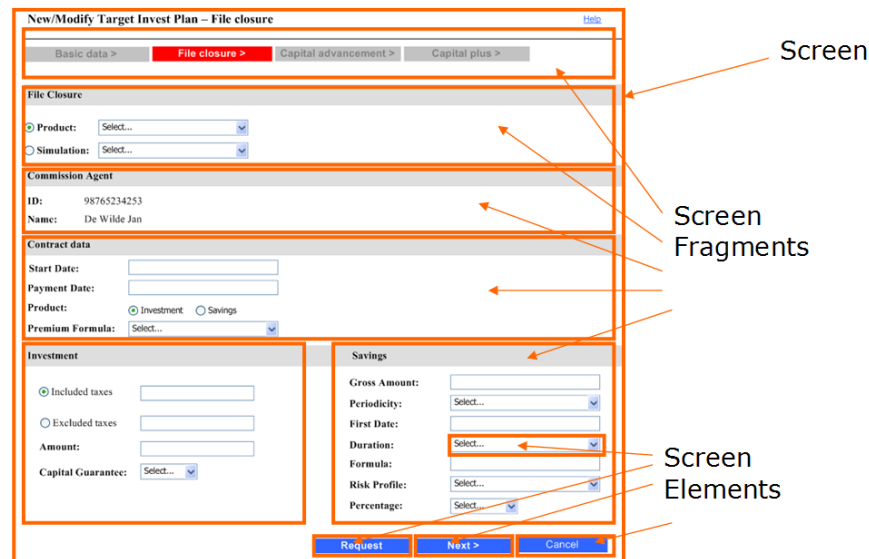


Fig. 4.3 Segmentation and annotation of a web form UI components

Table 4.1 Decomposition of User Interfaces

UI Component	Composed Of	Constraint
Screen Group	Screen Group Screen	Optional At least one screen
Screen	Screen Fragment Screen Element	Mandatory At least one screen element
Screen Fragment	Screen Fragment Screen Element	Optional At least one screen element
Screen Element	-	Mandatory

in modules to compose applications, functional decomposition is valuable to provide insight into the structure of the application in terms of its UI components.

This model was initially created to allow alignment of user interfaces with business processes. This UI model addresses several types of user interfaces, from different platforms, which are very common on business environments and the diversity of user interface components, which arises constantly due to the richness of modern user interface toolkits. But its simplicity and technology-independence makes it applicable to different contexts beyond the corporate setting.

Each UI component can be segmented by drawing a rectangular format with varying height and width, forming a specific dimension. This simple format is generic enough to accommodate different formats of UI components, such as labels, fields, calendars, and videos. The segmentation of UI components is implemented using Yasmim.

4.4 Why a Model-Driven Approach

Model, as an abstract representation of complexities present in the real world, has been applied in most of the knowledge fields. It is not different in computer science, where models have an important contribution in research and industrial projects. The complexity behind systems are usually addressed by the Unified Modeling Language (UML), which defines classes, objects, activities, states and so on [76]. Other models help with the representation of data and its relationships, the domain and even the user interaction.

When models are applied in system projects, it is very important to follow a Model Driven Engineering (MDE) approach, which is open and integrative. It combines domain-specific modeling languages (DSML), transformation engines and generators to support the systems complexities [95]. A DSML repre-

sents a domain using meta-models with a structure to specify concepts, their relationships, semantics and constraints, which are useful to promote transformations between models and generation of other artifacts. These abilities are well appreciated in heterogeneous environments, complex systems and large applications.

The Object Management Group (OMG) defines the Model Driven Architecture (MDA), a derivation of the MDE, as an approach to keep business and application logic separated from platform technology [75]. When modeling according to MDA principles, a software can be developed in many programming languages and the effort to migrate to another language is reduced, since a significant part of the code can be generated. Specific platform features and behaviors can be considered to specify precisely the output of transformations and generations. Even the final user interface can be described by MDA-compliant models.

The power of models to represent complex problems, their maturity supported by MDE principles, and the recent contributions in the HCI field to address the user interaction through models, form the basis of this case study. We believe that using models, whose need and importance vary according to project needs, it is possible to improve the team productivity and reduce the software maintenance overhead. We are inspired on, but not limited to, two very representative scenarios:

- Software projects for different devices; and
- Many user interfaces in large systems.

Concerning the first scenario, possibilities of user interaction with systems are increasing at the same time as the evolution of devices and architecture grows. Some years ago, the main business of user interaction was related with personal computers. Nowadays, the technology brings more capability with less space, more communication and mobility, increasing the computational power of small devices, like mobile phones, personal digital assistant (PDA), tablet PCs, and so on.

The variety of devices increases the demand for innovative applications. In many cases, such variety brings improvements for current software products to cross the desktop frontier, like a PDA application, or to improve the operation of the system, like a sensitive touch screen. In both cases, the user experience is improved, giving more productivity and efficiency in different contexts of use.

However, there exists a considerable effort behind the scene. A new device brings new development constraints, like small screens, or requires new skills to manipulate improved capabilities, like speech recognition. A field of dimensions, colors, depth, sounds and senses are challenges to be handled by

UI designers, architects and developers, who spend a lot of time to conceive, design, build and test their products in different devices.

Usually, projects for multiple devices have more requirements to address. They consider common requirements, which should be implemented in all devices, and also particularities of each device, such as the processing time of a workstation or the mobility of a PDA or even the communication power of a mobile phone. Particular requirements are expected because it helps to get all advantages of the environment to which the device was designed for. But, common requirements take into account the time and the cost to develop the same need in different devices with some difficulties to reuse codes, user interfaces, integrate all data and other artifacts.

Another important scenario where the application of models makes the difference is on management of many user interfaces in large systems, common in ERP and workflow systems. Usually, this kind of system considers the business process model as an important artifact. A business process model is a structured set of activities designed to produce a specific output for a particular customer or market [23]. In the user point of view, this artifact is important as well because part of the processes' activities demands user interaction. The role of people who are responsible for those activities and what is necessary to do in each one are essential requirements to conceive UI supporting the roles they play in the organization. Therefore, when the process changes, probably, the software should change accordingly. But, to determine what exactly needs to be changed is subject of a precise requirements control and traceability, which becomes more complex proportionally to the software's size and frequency of changes.

Models can represent how users interact with business' activities. In parallel, they help to link activities and its respective user interfaces, promoting traceability to optimize maintenance. When an activity changes, the model determines the revision of the user interactions necessary to execute such activity. It could involve one or more screens, depending on the complexity of each case. Changing the right code keeps the alignment between processes and systems [103].

To address these and other real world problems, model management, an specialized branch of metadata management, organizes a collection of models and mappings between them to allow transformations, merges, generation and many other operations [5]. Surprisingly, even with a good support from tools, models for user interfaces still demand a model management support that effectively considers project teams, a large scale of model instances, operations and reuse.

So far, in the domain of engineering of interactive systems, the community has reached some consensus about a common way to structure models for user interfaces. One of the most cited methodological frameworks used for this purpose, but not the only one, is the Cameleon Reference Frame-

work [11], which provides the necessary support for those user interaction challenges. This framework has four abstraction levels that are intended to express the UI development life cycle, as shown in Figure 4.4. These abstraction levels follow MDE principles, achieving the necessary support for different platforms, devices and contexts of use and a considerable effort is running to provide computational and methodological applicability. In this case study, the adopted level is the one devoted for the abstract user interface, as previously explained in this section.

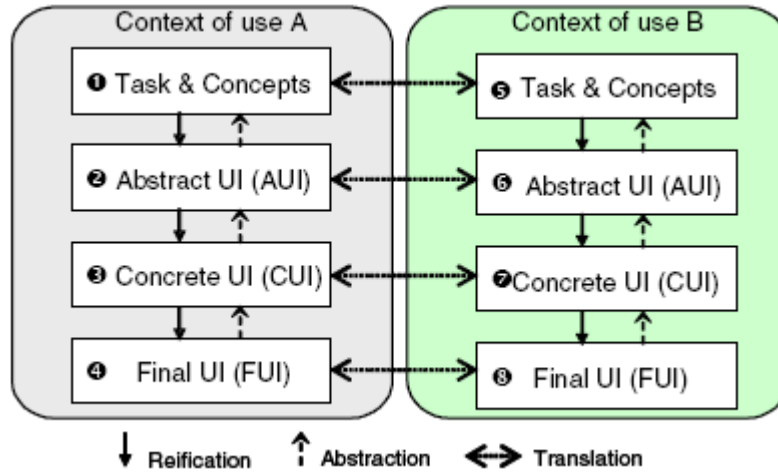


Fig. 4.4 Abstraction levels of the Cameleon Reference Framework

4.5 Applying the Multimedia Archiving Framework

Using Yasmim's workflow, during the processing phase, we have the media file, which is a screen shot. It is analyzed to extract low level data, such as size, resolution, colors, etc. These data are used to compare the media with existing ones in order to avoid duplicity, and also to parametrize algorithms.

At the segmentation phase, the user selects meaningful parts of the screen shot using spatial segments. The system gives the freedom to add as many segments as necessary and allows overlapping. Segments can be linked to each other and the link can also be annotated, describing a possible relationship between two or more segments. This feature can be useful specially in situations where elements of the user interface are interrelated, which is the case of an alignment button when pressed aligns the text in a text area ([but-

ton] -format- [text area]), or a slider that resizes an image ([slider] -resize-[image]).

Semantic information is related to segments at the annotation phase. The expertise of the user may guide him/her to choose the most appropriate annotation technique according to the nature and complexity of the content. The querying phase considers that segments were created and annotated. Using annotations as an index, screen shots can be located in the database according to the user needs. The querying mechanism is explored when the end-user faces a particular situation in a UI and needs to find the corresponding screen shot to give the feedback. Last, but not least, the retrieval phase returns the media or a list of candidates according to the submitted query. The retrieved media can start the cycle again, in case the user decides to create new segments or annotations.

The use of segments in itself would be enough to express the user feedback. However, doing it in a more structured way would bring more benefits than in an ad-hoc manner. The adoption of a structured representation gives a common understanding of the segments, defines how they are interrelated and allows segments to be reused in other possible applications.

In this case study, Yasmim is used to store screen shots of the enterprise systems and the *UI-Business Alignment Platform* (Usi4Biz) accesses Yasmim to associate these screen shots with the business processes of the company.

One of the main functionalities of the *UI-Business Alignment Platform* is *Impact Analysis*: When any element is selected from either a business process model, task model, UI components, it is possible to see, for the selected element, which are the related elements on the other models that would be impacted if a change is to be made on the selected element. The traceability between these models is demonstrated by navigating in a network of links, so when a node in a model is selected, the traceability lists the elements that are related to it in the other models.

Figure 4.5 depicts, on the left of the platform, the business process shown in a tree structure, where one activity is selected. On the middle, there is the task model with impacted UI tasks, marked with dashed lines. On the right, there are the impacted UI components, marked with dashed lines. This trace is created through the mappings between the models.

This functionality provides clear visualization of information needed to make an impact analysis and decide whether or not a specific change will be made. A change can be performed on: a) the business process model, which is detected as a new version of this model is imported; b) the task model, which is done directly on the task model; c) the UI components, importing a new list of components or manually changing the components. Each detected change in any of the models is subject to analysis and approval.

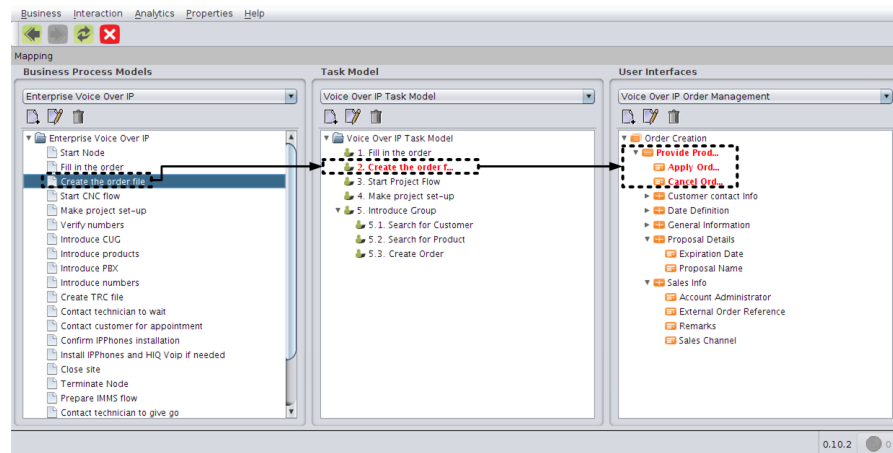


Fig. 4.5 Impact Analysis Visualization

Figure 4.6 depicts that the user can associate a screen shot with a specific screen of the application and segment it using the structure of the abstract user interface model. This way, users can identify screen fragments and screen elements on the screen shot.

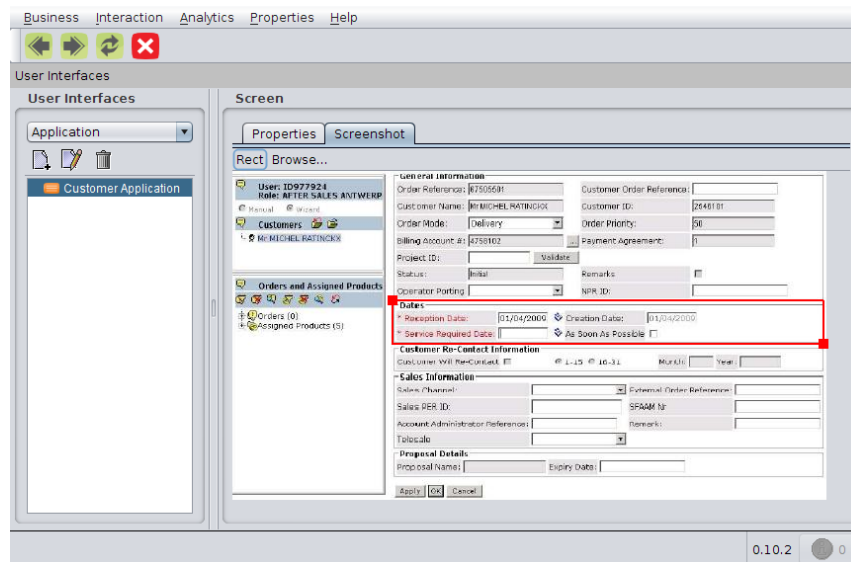


Fig. 4.6 Segmentation and Annotation of Screen shots

4.6 Links Between UI Components

The way user interfaces are designed holds useful information that communicates what designers thought about the user interaction. Based on the designer's experience and adopted guidelines, the navigation between screens is defined, the appropriate layout for the desired presentation is chosen, the arrangement of UI components follows the expectation about the user interaction, and so on. In an analytical point of view, it is possible to visualize links between design elements. When defining the navigation, designers create links between pages or screens. When defining the layout, links help to represent the relative position of one screen fragment (e.g. header) over another (e.g. lateral menu). Considering the arrangement of components, besides defining the relative positions, links also help to specify the sequence of UI components presentation. Therefore, the relationships represented by links have associated meanings and fulfill an important role in the UI understanding and further analysis.

The formalization of the relationship between UI components has been supported by model-driven UI development [11, 55]. For the interaction dynamics, task models have been used to precisely represent how users interact with applications. Task relationships are one of the main characteristics in task models that have an impact on UI design and they can be seen as links between design elements. Both are useful source of information of what designers thought about during interaction design.

Links may have different meanings according to their different applications. We refer to the meaning of the link application as a perspective. The identified perspectives are:

- **Navigation:** screens are statically linked (depends on the design) or dynamically linked (depends on the user input) to represent the flow in which information is visualized and managed.
- **Association:** in some cases a component has influences, of any kind, over other components. A date field, for example, that has a calendar feature activated by a button on the right, is associated with the button to provide a better user experience. In this case a link can represent this association.
- **Simulation:** links between UI elements can help simulate the user interaction for test purposes. Developers can create several configurations, or sequence of use, trying to predict the user interaction and analyzing the application behavior.
- **Generation:** a UI model containing links between components can be used to generate final UIs in the exact sequence of the resulting pipeline.
- **Tracking:** links represent the order in which UI components are used at run-time by users, which may defer from the one specified at design time.

From the segmentation point of view, each UI element is properly segmented using spatial segments and links are created between those segments to enable the relationships. What determines if the link is navigation, association, simulation, generation or tracking is how the link is annotated.

Annotations define the kind of link and additional properties, such as the influence a segment has over others. Some annotation meanings are inspired on operators already in use by existing UI models. The task model, for instance, follows the LOTOS temporal operators. LOTOS [44] is an international standard notation used in various industrial and research centers to specify applications which have, as an important feature, temporal ordering among possible actions. These operators have been used in task models and their related methodologies and tools (e.g. [68], [69], [112]) quite successfully.

From the experience with task modeling and UI design, we have decided to adopt task relationships as the set of links between UI components. As follows, we present the task relationships in Table 4.2 followed by the definitions for each *relationship type* [56].

Table 4.2 Representation and definition for Task Relationships

Relationship	Lotos symbol	Description
Enabling	$T1 \gg T2$	T1 has to be finished in order to initiate T2.
Enabling with information passing	$T1 \square \gg T2$	T1 has to be finished in order to initiate T2 and T2 is synchronized with T1 on some piece of data.
Suspend / resume	$T1 \gg T2$	T2 may interrupt T1 before the termination of T1. Once T2 is finished, T1 may be resumed.
Deterministic choice	$T1 \square T2$	Once one task is initiated the other cannot be accomplished anymore.
Disabling	$T1 [\gg T2$	T2 may interrupt T1 before the termination of T1, but T1 cannot be resumed after T2 has terminated.
Order Independence	$T1 = T2$	T1 and T2 can be performed in any order.
Concurrency	$T1 T2$	T1 is interleaved with T2.
Concurrency with information passing	$T1 \square T2$	T1 is interleaved with T2 while they synchronize on some data.

The set of relationships between UI components uses this same set of relationships that are necessary for UI design and user interaction. In addition, if users find the need to add new relationships that are necessary for UI design

and user interaction, such as input (i.e. a text field that receives date from calendar; a text field that receives file location from file explorer; etc), they can do so using Yasmim.

One advantage of using annotation on UI components in comparison with the relationships in traditional task models is that we can create links in different directions, other than unidirectional relationships used in the current task models. This flexibility enhances the expression of possibilities of user interaction, such as representing that users can return to a previous UI component.

4.7 Conclusion

Our proposal of user feedback is an efficient way to put the user at the center of the user interface improvement. The main point is to make the feedback permanently happening at the moment of the user experience, not waiting for meetings or large textual descriptions. We have contributed with the integration of a multimedia archiving framework with an abstract UI model, aiming to simplify and enrich the feedback content, by empowering the user with more expressiveness through the multimedia support.

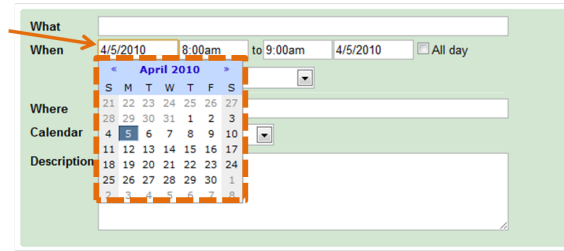


Fig. 4.7 Floating component

To consider several types of user interfaces and interaction possibilities is a great challenge. Situations where components appear floating on the screen, such as a dialog box or a calendar helper (Figure 4.7), and components that dynamically change their size according to user's activity, such as a text area that grows as the text is inputted or a data table that resizes proportionally to the volume of data (Figure 4.8), are very common.

From the modeling point of view, this challenge can be well addressed by adding two boolean properties in UI components, which are Floating and Elastic. Floating indicates that a UI component can change its position on the screen, manually or dynamically. Elastic means that it can change its

The figure displays two screenshots of a web application interface for creating and managing books. Both screenshots show a form titled "Create a new book" with input fields for "Title: *" and "Author: *", and "Add" and "Reset" buttons. An orange arrow in the top screenshot points to the "Add" button.

The bottom screenshot shows the form with a table of records below it. The table has columns "Title", "Author", and "Operation". The records are:

Title	Author	Operation
Committed	Elizabeth Gilbert	Remove
Made to Stick	Chip Heath and Dan Heath	Remove

Fig. 4.8 Elastic component

size without negatively interfering on the overall design. On the other hand, from the implementation point of view, the dynamism brought by these two properties requires the current implementation of the multimedia archiving framework to consider a re-engineering of the segment implementation.

Chapter 5

High Level Multimodal Fusion

This chapter describes a high level multimodal fusion system that uses a multimedia framework, with segmentation and annotation support, to manage a long-term memory to preserve what was learned from the modalities. We analyze an office scenario where two co-workers have a conversation. Based on their behavior and speech, we realized their intentions and managed them to give advices in order to help the execution of their tasks.

5.1 Introduction

Modality fusion fulfills an important role on the development of applications with support for multiple modalities. It integrates data and meanings coming from different sources. These applications are progressively evolving towards more robust semantic interpretation of the user's intentions and a fusion mechanism is important to combine data streams by reducing uncertainty. To formalize the coordination between modalities involved in the same multimodal interface, it is required to extract relevant features from signal representations and thereafter to proceed to high level fusion, which is a non-deterministic approach for the integration of multimodal semantically grounded input commands [35].

The effort on the implementation of fusion mechanisms has shown that solutions are dependent on the context [81] and other variables such as available devices, granularity of collected data, available analytical algorithms and a large heterogeneity of technologies are difficult to integrate. These issues lead people to implement fusion mechanisms focused on specific case studies. This focus imposes researchers to rewrite or duplicate a solution for each new case study, which causes waste of time and resources, rewriting requirements that are common between them, but not reusable.

To contribute with the consolidation and unification of multimodal fusion approaches, this work proposes a system to provide what is common in most fusion implementations and support extensions to allow developers and researchers to attach their algorithms [66]. The system manages important requirements, such as:

- parallel processing of data coming from different devices;
- synchronization of data on time;
- formalization of a well-defined process, composed of activities executed sequentially and continually; and
- an agent-based layer responding to events of the process to perform different kinds of fusion.

The system is not a fusion mechanism in itself, but it is the basis to instantiate fusion mechanisms. Developers can use it to plug their existing tools, devices and fusion techniques. It manages the execution and communication between all attached pieces, avoiding this basic structure to be repeatedly implemented.

Relevant works have been developed in this direction. Parageorgiou et al. [78] propose a multimedia, multilingual and multimodal research system (CIMWOS), which is robust in terms of archiving, indexation and retrieval, but limited in terms of the number of supported modalities and types of contents. The work of Peng Dai et al. [22] also proposes a multimodal archiving system. However, it is used for a particular case, which is meeting scenarios, and considerable changes are needed to support other domains.

5.2 Fusion Essential Requirements

Many fusion engines have been developed so far with different approaches to allow data fusion, but they have specific purposes, solving particular scenarios of fusion [8, 27, 61]. When a new scenario is identified, a common practice is to implement the fusion from scratch due to the lack of tools or systems providing the basis for multiple implementations. We believe that the field can be substantially improved if a general support for fusion is provided, taking into consideration common features of existing fusion mechanisms. A recent survey on fusion engines for multimodal input corroborates this by proposing a research agenda for future works. It remarks the importance for engineering aspects of fusion engines including software tools to fine-tune fusion by designers and tools to rapidly simulate and configure fusion engines to a particular application [49]. Exploring existing contributions, we have identified the following basic requirements:

- *Synchronization*: Since we have at least two modalities to justify fusion and they are both active during the user interaction, the synchronization in time is an important variable to analyze the correlation between events coming from each modality. This is based on the premise that if two events have approximate time stamps then they have a high probability to be correlated.
- *Cognitive Algorithms*: input modalities usually produce a high volume of data to be analyzed. Efficient data analysis can be provided by expert algorithms, using statistical models, first order logic, fuzzy logic, production rules and other approaches.
- *Context Representation*: It is important to know the context of the data to be analyzed in order to improve efficiency. The context [11] is a complete description of the environment, where the interaction occurs; the user, who interacts with the system; and platform, which supports the multimodal application and its fusion properties.
- *Visualization*: large volume of data under analysis must be appropriately visualized by specialists, who follow the recognition, identify problems or non-recognized data. Visualization is a sort of high level test and demonstration.
- *Simulation*: Sometimes it is difficult to repeat an experiment many times to validate the fusion mechanism. Simulation is important to perform tests during the development or demonstration of the fusion mechanism, without increasing time and resources during the process.

To provide a general support for fusion, it is necessary to consider more than the features of existing fusion mechanisms, but also a model to support the variety of implementations and platforms in use. For instance, the literature presents Neural networks [119] and Bayesian networks [108] as distinct approaches to fuse data, but considering a general system, both approaches should be supported and even complementary when necessary. This particularity shows the need for requirements not considered in fusion mechanisms so far, which are:

- *Component-based*: because of the complexity of data fusion, mechanisms are usually composed of several implementations. A component-based architecture allows high cohesion and low coupling to provide extensibility and reuse.
- *Multi-platform*: there are useful implementations available for fusion in different programming languages and platforms. The possibility to reuse what is already available instead of re-implementing in the chosen platform, might allow fast development and better results.

- *Scalability*: a fusion mechanism could be heavy and time consuming due to the volume of data to be analyzed. A multi-thread architecture is needed to allow fully use of the computational resources.
- *Distribution*: sometimes a unique machine is not powerful enough to produce efficient results and using multiple machines can fulfill the demanded computational power.

We have developed a system to fulfill these requirements. It was developed in JavaTM on top of the OpenInterface (OI) Platform [51], further described in section 5.4.

5.3 System Design

The design of the system is based on the essential requirements discussed in the previous section. It is organized in uncoupled and complementary layers, which are:

- *Modality Processing Pipeline* (MPP): a set of components with well delimited roles that transforms data by reducing its granularity from a signal level to a semantic level;
- *Autonomous Fusion Agents*: agents activated during the execution of the modality processing pipeline to perform modality fusion techniques on the data under transformation; and
- *User Interaction*: user interface to allow configuration, execution and visualization of the fusion process.

The modality processing pipeline is a bus of data coming from each modality. The autonomous fusion agents constantly analyze these data, comparing with existing data and performing fusion. These layers are described in more details hereafter.

5.4 Modality Processing Pipeline

The modality processing pipeline is composed of four stages, which aim to reduce the granularity of data until a level of natural understanding (i.e. data readable by human beings). The process starts when data representing signals are received by the system through TCP/IP connections. This protocol was chosen because of its reliability, which is more important when analysis is

necessary, and to allow distribution. The TCP/IP connections are managed by the Grizzly framework [43] to allow scalability. The result of the recognition is then segmented to determine the useful part of the content. This part is once again analyzed to identify possible meanings and these detected meanings are finally annotated according to the domain of the context under experimentation.

We are using the OI platform to create the MPP. This platform is composed of the OI Kernel, which is a generic runtime engine to integrate heterogeneous code by means of non-intrusive techniques, and the SKEMMI, which is a tool used to manage and compose components on top of the OI Kernel [50]. This tool is used in this project to create the MPP as represented in Figure 5.3, where each component has its own representation and they are linked through their inputs and outputs.

We are also using Yasmim to manage all collected data within the process. As data collectors, we have cameras and microphones. Cameras produce videos and microphones produce audio, which are stored to be segmented and annotated for posterior decision analysis.

5.4.1 Recognition

The developer implements a recognition component to manage detection devices, such as cameras, microphones, and accelerometers. The recognition phase is focused on the analysis of signal features to identify useful content. For instance, considering that a camera is used to track people in a room, the recognition component should return the segments delimiting where those people were located in the scene. The same image can provide the relative position of a person considering other objects in the scene.

5.4.2 Segmentation

The segmentation component receives recognized data and creates segments to delimit the useful data from the original content. Other segments delimit the objects identified in the scene. The type of segment depends on the type of content. A temporal segment can, for instance, delimit all instants where the microphone detected people's conversation, but it rejects parts of the audio where noise was detected. A spatio-temporal segment can delimit all instants and positions where people were moving in the scene.

Once detected, segments are stored using the Yasmim framework, which also associates those segments with annotations afterwards. Temporal segments are created for audio and video contents. Spatio-temporal segments are created for video. Spatial Segments are not used in this case.

5.4.3 Meanings Extraction

What is recognized and segmented is not actually endowed with meaning. A second round of analysis is needed to identify the semantics of the useful content. This is the role of the meanings extraction component. Here, the segmentation component offered an important support by reducing the scope of the analysis to the delimited content, improving the performance of the meanings extraction. An example of meanings extraction is the semantic analysis of a sentence, recognized using a speech recognition tool. It could also be the meaning of the relative position of a person to an object in the scene. The validation of extracted meanings is made by searching the terms found in the knowledge base, which describes the context in the format of an ontology.

5.4.4 Annotation

Lastly, extracted meanings are instantiated in the knowledge base according to the concepts of the domain described in an ontology. Because the adopted ontology model is based on RDF (Resource Description Framework), meanings are stored in a triple format, composed of a subject, an object, and a predicate linking them [115]. Some examples of triple: 1. `person -is close to computer`, 2. `gesture -is- circular`, 3. `John -wants to call- Nick`.

Figure 5.1 depicts a modality processing pipeline for each modality. The system is responsible for creating a new channel each time a new device is connected. The data received through the channel is processed by the pipeline. It is important to notice that no fusion is done in this layer, but all data is prepared to be fused by autonomous fusion agents described as follows.

The annotation stage was initially inspired from the work of Jean-Claude Martin, who adopts Anvil for video annotation in a multi-modal system [62]. We have formalized and extended the annotation support in the Yasmim framework. In this case, we are applying the domain concepts annotation,

based on a restricted domain described in an ontology. The ontology is described in section 5.10.

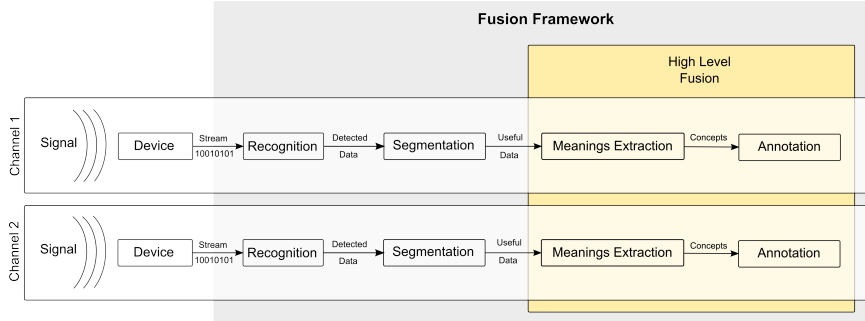


Fig. 5.1 The fusion system life cycle

5.5 Autonomous Fusion Agents

The MPP should be processed without interruptions of the fusion implementation. It could be time consuming, interrupting the modality processing and, consequently, losing important data coming from devices. A standard implementation of fusion in separate threads could be done, but it represents more effort to detect events from the MPP. Therefore, a multi-agent architecture is supported to fulfill this need.

This layer assumes the architecture of a multi-agent system, which is a system that includes multiple autonomous entities with diverging interests [99], called intelligent agents [91]. The intelligent agent is capable to perceive the environment and act according to its conclusions. Its capability to perceive monitors the meanings extracted from segments. Since the environment is described in the ontology, the agent uses it as its knowledge base. Its capability to act performs reasoning with extracted meanings from all connected modalities to produce a conclusion about the user intention. Previous conclusions about the user intention are reused by the agent in new reasoning cycles to improve the latest conclusion.

We have integrated the Soar [67] cognitive tool in the fusion system to fulfill the reasoning needs of this layer. This tool intends to help with the development of systems that exhibit intelligent behavior. It is a theory of what cognition is and an implementation of that theory through a software architecture for Artificial Intelligence (AI). Soar handles the capabilities of intelligent agents and uses production rules to govern their behavior [52].

Production rules are standard patterns of inference that can derive conclusions leading to a desired goal [91]. It is a statement that has two parts, an if clause (precondition) and a then clause (action).

5.6 User Interaction

To simplify the use of the system and provide practical information about the modality processing and fusion results, a user interface was developed for this purpose. Figure 5.2 depicts a representative screen shot of the main screen. It shows horizontal bars corresponding to each channel in a time line, which are divided in bars representing identified segments. The content of the segment and its associated meanings are shown in a pop-up window when the user clicks on a bar. These bars show, in real time, what happens during the MPP processing.

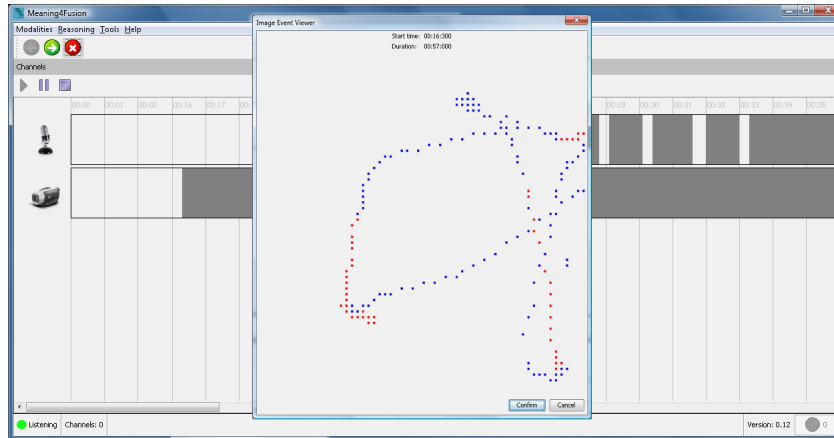


Fig. 5.2 A screenshot of the fusion system

This user interface also helps the user to configure and perform the modality fusion. In configuration time, the user composes a pipeline in the OI platform (Figure 5.3) to deal with each modality and associate it to a kind of channel, such as video analysis. A channel is instantiated for each input modality, mapping and connecting computers and devices. Once configured, the system can be started, activating all channels at the same time. Each channel opens a port and waits for data from different sources. The user interface provides all support to start, pause and stop this process.



Fig. 5.3 The OI pipeline of the system

5.7 Case Study

This case study considers the use of the system to instantiate a fusion mechanism to be applied in an office environment scenario. The purpose of this application is to assist workers in their daily tasks in the office. The application analyzes their behavior, perceives their intentions and recommends them to perform the task. In order to develop and validate the whole experiment, we defined five different scenarios where two people interact mutually in a room, talking in a natural way and behaving without restrictions. Each scenario tries to explore specific combinations of speech and behavior to increase the robustness of the system. This system is able to track people, analyze their behavior, movements, speech, and make decisions about how to prompt necessary information when required or provide any other assistance.

In order to efficiently analyze behavior, the system has to correctly process, interpret and create joint meaning of the data coming from speech analysis and video scene analysis. We consider that human behavior is goal-oriented, so our main aim is to recognize users' plans and produce an advice about how to better perform the task.

The system manages the data streams arriving from two sources: video scene and speech. In particular, we show a technique distinguishing between the data from different modalities that should be fused and the data that should not be fused, but analyzed separately.

The fusion mechanism [114], instantiated on the fusion system, employs various components in the MPP layer according to Table 5.1. In the autonomous agents layer, we have considered a decision-making framework, Soar, to perform the high level fusion and Protégé [105] to manage the knowledge base.

Table 5.1 Fusion Mechanism Components

Component	Implementation	MPP Part
Speech recognition	Sphinx-4	Recognition
Speaker identification	Matlab	Recognition
Syntactic parsing	C&C Parser	Segmentation
Semantic analysis	C&C Boxer	Meanings extraction
Video analysis	Open CV	Recognition
Behavior analysis	Own	Meanings extraction

The system employs various components integrated into a single system: speech recognition, speaker identification, syntactic parsing, natural language semantic analysis, video analysis, human behavior analysis component and a cognitive architecture that serves as the context-aware controller of all the processes, manages the domain ontology and provides the decision-making mechanism.

Figure 5.4 describes how all components mutually collaborate when connected to compose the overall solution. The process starts when an audio or a video signal is detected for the first time. There is no restriction about which signal should start first because all modalities can be processed in parallel and independently.

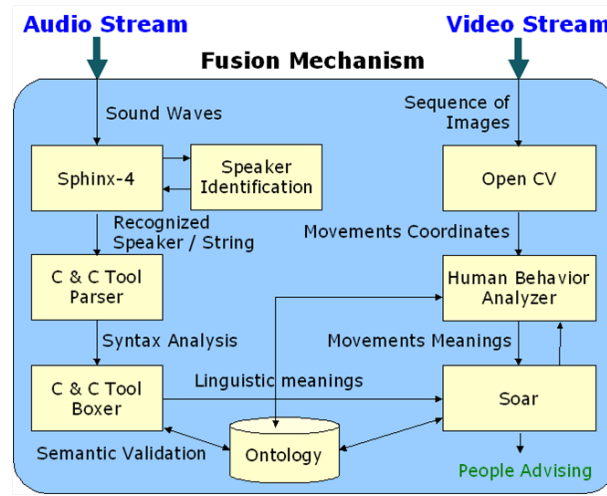


Fig. 5.4 The system architecture

5.8 Speech Analysis

When an audio stream is received, the speech recognition component processes it, generating a string of what was said. The same signal is sent to the speaker identification component, which associates what was said with who said that. The string is sent to the syntactic parsing component to identify the syntax of each word, which is important for the natural language semantic analysis component, responsible for the identification of the subject, the agent, the predicate, the object of interest and other elements. From the semantic analysis, it is possible to extract semantic structures very similar

to the structure of the knowledge base, represented by the ontology. If we find the identified semantic in the ontology, then it means that the sentence is valid inside the context and can be useful to fuse with other meanings coming from other modalities.

5.8.1 Speech Data

The speech was recorded by two non-native English speaker subjects, one 23-year Chinese male and one 32-year Swedish male. The data was recorded using 16kHz, 16 bit audio. The 5 scenarios consisted of a total of 72 sentences and 148 seconds. For the development of speaker identification, we used ten phonetically rich sentences for training, and for parameter tuning we used another ten phonetically rich sentences and ten words of different length. To illustrate the recorded data, we put below the dialogue of the second scenario, where the conversation is short, but also complex to detect the user intention.

1. Beto: Hi Ronald! How is life going?
2. Ronald: I am fine.
3. Ronald: I want to call Nick.
4. Beto: What for?
5. Ronald: He mentioned that he attended a wine tasting course.
6. Beto: It sounds interesting, I like wine.
7. Ronald: Actually I plan to join the next class. He also mentioned a book about French wines, but I cannot recall the name of the author.
8. Beto: Why don't you send an e-mail to Nick?
9. Ronald: Maybe I can find a book about it in the library.
10. Beto: Yes, you are right.
11. Beto: Did you find it?
12. Ronald: Yes, I did.

5.8.2 Speech Recognition

For speech recognition, we used Sphinx 4, which is an open source, JavaTM-based speech recognizer [117]. For acoustic modeling, we used the 8 Gaussian triphone models, trained on the Wall Street Journal Corpus, which are sup-

plied along with Sphinx. Since we wanted to allow the system to monitor a discussion between two or more people, we want to have a large vocabulary language model. For this purpose, 3-grams with a maximum of around 5000 words were trained using the orthographic transcriptions from the Wall Street Journal Corpus. The 5000 words were selected as the most common ones plus the ones that are present in the scenarios.

In order to enhance understanding of concepts, we wanted to evaluate different methodologies to pick hypotheses from the N-best list which Sphinx can produce. For this purpose, Sphinx was configured to produce as long N-best lists as possible given a memory constraint of 1 Gb. This resulted in list lengths of around 100 hypotheses for short utterances with good accuracy and around 10000 list lengths for long utterances with poor accuracy. Two different approaches may be used. The first is to estimate a re-scoring function on data, the second is to apply re-ranking heuristics. The first approach is likely to be more efficient, but it requires enough data to be done optimally. The second approach is easier to apply and is likely to be more robust to less knowledge about what the users may say.

By making an assumption about the domain, a word list consisting of all nameable things in the ontology was queried. Three kinds of re-ranking heuristics were tried out. In each of these, a score was assigned to each hypothesis and the N-best list was re-sorted using this score in such a way that the initial ranking was preserved if two or more hypotheses had equal score.

Method A: The score is set to one if one or more words from the ontology are found.

Method B: One scoring point is added for each matching word in the ontology word list.

Method C: A score is added, which is equal to the word length in characters, for each matching word in the ontology word list.

Table 5.2 shows word accuracy and correctness for hypothesis selection methods and Table 5.3 shows concept accuracy and correctness for hypothesis selection methods. In Tables 5.2 and 5.3, the performance gain by this approach is compared to selecting the 1st top hypothesis and by letting an oracle select the hypothesis which maximizes Accuracy. These results were obtained using a preliminary version of the ontology. While Accuracy takes substitutions, insertions and deletions into account, Correctness only takes substitutions and deletions into account. In Table 5.3, Accuracy and Correctness are computed only by using the words from the ontology as a reference, forming Concept Accuracy and Concept Correctness. The idea is to examine to what extent information, which the cognitive modules can process, is present.

Table 5.2 Word accuracy and correctness

Hypothesis selection method	Accuracy	Correctness
1st	56,30	64,52
Method A	50,13	62,98
Method B	46,53	63,50
Method C	47,81	64,01
Oracle	74,81	78,15

Table 5.3 Concept accuracy and correctness

Hypothesis selection method	Accuracy	Correctness
1st	54,12	67,06
Method A	27,06	68,24
Method B	10,59	78,82
Method C	12,94	81,18
Oracle	78,82	74,12

From the results presented in Table 5.2, it is clear that the methods do not improve over the baseline (1st hypothesis). It should also be noted that these results are obtained using a preliminary version of the ontology. From the results in Table 5.3, it should be noted that while there is a large drop in Accuracy, there is a clear increase in Correctness, which means that insertions are present. This means that the information content in terms of words known to the ontology is increased on the expense of word insertions.

5.8.3 Speaker Identification

Speaker Identification is the task to determine who is speaking. For the application described in this report, a standard speaker identification system was considered. It is based on Mel-Frequency Cepstral Coefficients (MFCCs) and Gaussian Mixture Models (GMMs), as in [87]. We used 28 log-Mel filters between 300 and 8000 Hz, cosine projected to 24 dimensions, where the first 12 with their delta were used. A simple data driven procedure for speech detection was tried: The MFCCs were clustered using the k-means algorithm with euclidean distance for two clusters. Then the cluster, which had the highest energy was marked as “speech” and the other one as “non-speech”. Then these two clusters were used for frame-based segmentation. Visual inspection showed that the approach seemed reasonable. No channel compensation was used. This system was implemented in Matlab.

The two speakers in the scenarios were enrolled, using ten phonetically rich sentences. For parameter tuning, another ten phonetically rich sentences and ten words were used. The performance measured in classification accuracy for various number of Gaussians are shown in Table 5.4 with and without speech detection (SD). Further analysis showed that the errors occurred for test utterances containing very short words, such as “no” and “hi”. Using speech detection did not improve accuracy. Based on these results, 16 Gaussians were chosen for evaluation using the recorded data for the scenarios. The accuracy for the evaluation data was an average of 94%.

Table 5.4 Speaker identification accuracy

Gaussians	No SD	SD
2	85,00%	92,50%
4	90,00%	92,50%
8	97,50%	92,50%
12	97,50%	92,50%
16	97,50%	90,00%
20	97,50%	95,00%
24	92,50%	95,00%
32	97,50%	92,50%

5.8.4 System integration of the speech components

A speech module, consisting of Sphinx 4 and the Speaker Identification system described above was fully implemented in a TPC/IP client. Also on the server side, the re-ranking method A was implemented in a way that makes it easy to extend to Method B or C.

5.8.5 Syntactic and Semantic Analysis

The next stage after speech recognition is syntactic and semantic analysis of the discourse. For our purposes, we use the CCG (Combinatory Categorical Grammar) parser, Release 0.96, developed by S. Clark and J. Curran [19]. The grammar used by the parser is taken from CCGbank developed by J. Hockenmaier and M. Steedman [41].

CCGbank is a treebank containing phrase-structure trees in the Penn Treebank (WSJ texts) converted into CCG derivations. It allows easy re-

covery of long-range dependencies, provides a transparent interface between surface syntax and underlying semantic representation, including predicate-argument structure. The grammar is based on “real” texts, and that is why it has wide-coverage, thus making parsing efficient and robust. The CCG parser has Boxer, as add-on to generate semantic representations - Discourse Representation Structures (DRSs), the box representations of Discourse Representation Theory (DRT) [2]. DRSs consist of a set of discourse referents (representatives of objects introduced in the discourse) and a set of conditions for these referents (properties of the objects).

Our initial experiments with the CCG parser together with Boxer showed that it suits well for our application in parsing speech, but on the other hand, it is naturally broad enough to need wide-coverage tools for processing. DRSs can be generated in different output formats in Prolog or XML. To link the DRSs output with the ontology we use the XML format.

5.9 Video Scene Analysis

When a video stream is detected, the image is processed by the video analysis component. This component analyzes some image features to calculate the position of each person on the horizontal plan of the scene, their movements' direction and it identifies who is each person according to a predefined profile. It is also important to identify people through this modality because we have to know the position of who is talking in order to associate the user intention with his/her actions.

The next step is to analyze the human behavior, comparing the movements of the user with a set of rules. The behavior is relative to fixed objects in the scene, which are defined in the context domain and are directly associated with the aid to be given by the system. The rules define the boundaries of what is near or far from a certain object. Then the result of the rule processing at this stage is: “[person] -is near- [the telephone]”, “[person] -is far from- [the computer]” or “[person] -is moving to- [the library]”. This result is produced for each person in each frame of the video. Individually, these results are not significant enough for fusion. We have to analyze the movements in many frames in order to have final conclusions. For instance: if in the last 80 frames, the rule engine produced “[person] -is moving to- [the library]” then we can conclude that there is a real intention to reach the library, considering some variables of the environment, such as area of the room.

The video sequences were recorded using a distributed 8-camera voxelised visual hull [21]. The description of the environment is obtained by processing each image of the videos using computer vision algorithms. In these video

sequences, there are 3 types of fixed objects (a telephone, some books and a computer) located in different positions inside the scene, there are also two persons that are moving and interacting with these fixed objects. In order to have a good description of the environment for each one of these scenarios, it is necessary to extract the information of the position of each fixed object, the position of each person at any moment and also the motion direction of each person. Using this information, it is possible to know if a person is near or far from each object or if a person is moving to an object. By this way, the system has good information about the human behavior in order to make better decisions.

Extracting all this information from the video sequences are common issues of the Computer Vision field. These issues are mainly:

- Object detection - to find the position of each fixed object;
- People detection and tracking - to find the position of each person; and
- Motion analysis - to find the motion direction of each person.

The computer vision system to extract this information was implemented in C++ using OpenCV library [15] developed by Intel. The objects (the telephone, the books and the computer) in these video sequences are always fixed, also they do not change size or rotate because the camera's viewpoint is always the same. Hence, the detection of these objects can be done by using a template matching algorithm [14]. These algorithms compare a template with a region of an image in order to determinate a similarity measure, wherein the similarity measure is determined using a statistical measure.

To obtain these templates, a sample picture of each object is captured from any image of the video sequence. The OpenCV operator `cvMatchTemplate` was used, this operator returns the probable positions in the image where the template can be located. This way, the most probable position corresponds to the area where the object is detected. The similarity measure that provides better results for this problem was the correlation coefficient normalized. Because each object never changes position, this template matching step is only done once in the captured image of the video sequence. Then, these positions are used for all the images in the video sequence.

There has been much work in the area of people detection and tracking using computer vision techniques. This problem has been a big challenge since many real-time vision systems require robustness in different environments. In the video sequence of this project, two people are talking with each other and also moving randomly inside the scenario in order to interact with these objects. The issues here are mainly:

1. the shape and size of each person can change over time because the people can move far or close to the viewpoint in different parts of the scenario;

2. the people can approach too much each other in the video, making the identification of each one more difficult;
3. one person can be partially occluded by the other person;
4. some body parts of each person can be outside the scenario because of the viewpoint of the camera; and
5. each person moves in a random way.

To solve these issues, a color-based tracking was used. The color of the clothes of each person was used, assuming that both people in the video have different color clothes. However, a background subtraction and a blob detection technique are needed in order to make the people detection robust for the cluttered environment and discriminate noise. Motion Templates algorithms were used based on papers by Davis and Bobick [24] to find the motion direction of each person. These algorithms are very fast and robust. The implementation was done using OpenCV Motion Template functions. These functions can determine where a motion occurred, how it occurred, and in which direction it occurred. To calculate the motion direction of each person, the silhouettes are updated in time using `cvUpdateMotionHistory` operator, after the motion gradient is calculated using the information of the temporal silhouettes (applying `cvCalcMotionGradient`). Finally, connected regions of Motion History pixels are found using OpenCV operator `cvSegmentMotion`. With this result, we have region of motions with their gradient directions in the foreground image.

For the human behavior analysis, we need to know whether every object is near or far from each person and the object toward which each person moves.

In order to find this information, we apply some rules using Soar and define thresholds using the result of the computer vision system (position of each fixed object and each person and motion direction of the people).

5.10 Ontology Design and Reasoning

In order to describe the context and allow reasoning, an ontology with comprehensive modeling pattern of multi-modal actions and prior knowledge about the objects and users is necessary. We used open source tool Protégé [105] to create the ontology in classes, properties and individuals.

We used Soar cognitive architecture in order to create and apply rules to query semantically the modality triples in the ontology. The ontology is re-implemented for adapting the specific working memory structure in Soar environment. The elements class, property and individual in Protégé corresponds to identifier, attribute and constant in Soar, respectively. In order to

have this knowledge base for reasoning, the ontology is created as states when agents are initialized. After receiving the input from external application, an operator is proposed and applied to map the input triple to the knowledge base and then to give out the result to the output link.

For the interaction between Soar and the external world, we use SML (Soar Markup Language) to send and receive commands packages as XML packets. The agent created by Soar is loaded by the SML codes and run until there is an outcome or run for a certain number of circles that is also set by codes. The outcomes are also checked by SML code and kept for further actions.

5.11 Results

In our experimental work, we used scenarios only with natural human language. We did not work with isolated words, commands, restricted language or similar features. The goal was to experiment with normal complete utterances expressed in a natural way.

In this experiment, to make multi-modal data fusion means to interpret human behavior, identify users' plans, and infer their intentions. Having understood what the people in the scene want to do, the system makes the decision on how to assist them in the given case. Looking at our challenging example scenario, transcribed in section 5.8.1, we can see that there are four points in this dialogue where the speakers express their plan to do something.

In (3) Ronald wants to call Nick, in (7) Ronald plans to attend the next class, in (8) there is a possible plan to send an e-mail and then this possible path of decision is changed to another route, in (9) there is an intention to find a book in the library. How shall the system realize which plans to take into account and which ones not? When to react and when not? This is why we employ multi-modal information. When the plan is identified from the user's words, we look at the other modality data to see if the person is going to "confirm" his/her words with the corresponding actions or not. In (3), (7) and (8) the people expressing their spoken intentions were still standing, just continuing the talk. And only in (9) Ronald moved to the bookshelves. That is why only in this last case of the plan expression, the system reacted and prompted where to find the desired book. In the phrase "Maybe I can find a book about it in the library", we have to resolve ambiguity between the library in the room and a library on the web. We do that using information from the other modality. We look if the person is moving to the books in the room or moving to the computer.

When identifying the person's plan from speech, we basically rely on the linguistic semantic analysis as described in section 5.8.5, but we certainly

take advantage of the obvious lexical signs of plan and intention expressions. For example, verbs and phrases, such as “want”, “wish”, “plan”, “going to”, etc. (we defined a total of nineteen expressions similar to these ones) in a certain syntactic context and in the present or future tense clearly point at the person’s intention to do something. And vice versa, negative forms of verbs, such as “I don’t want”, “I have no wish to...”, “You don’t want to...” as well as verbs in the past tense serve as stop-words, and signal that this plan should be discarded and not taken into account since no system response is needed.

The fusion occurred when we identified a person moving to the library and we also detected the intention to find a book in a sentence such as this one: “I can find a book about it in the library”. Therefore, we could conclude the person wants to find a certain book in the library and the computer could provide assistance for this person to find it, giving the exact location of the book, such as the bookshelf and its relative position to other books.

The composition of many different tools and components was definitively a challenge, achievable due to the system that has been developed on top of the OI platform. We used JavaTM and C/C++ programs and OI allowed the communication between all of them without any particular change in the source code. The decision making part was strongly impacted by the poor results of the speech modality. However, we could provide two advices during the experiment: Nick’s phone number and the location of the book about French wines because, in both cases, the user intention was well recognized with the speech recognition tool and user’s movements were well tracked towards the telephone and the library.

We are using only open source software to compose each part of the fusion mechanism. Unfortunately, these tools follow the state of the art very slowly and we could not get results that are possible to expect from proprietary or from inaccessible tools developed by companies and recognized research centers. One of the strongest impacts of this slow evolution was in the speech recognition modality, where Sphinx could not provide precise results and C&C Tools were inaccurate on semantic analysis when ignoring interrogative sentences and other punctuations. On the other hand, the computer vision modality provided precise results about people’s positioning and movement directions, due to the rich framework OpenCV, whose output is presented in Figure 5.5. The computer vision system was executed in a laptop with an AMD Turion 64 2000 MHz processor, 1024 MB of RAM and a graphic card NVIDIA GeForce Go 6150 (256MB). The computer vision system is capable of providing the required information of all the five scenarios in real-time processing (25 frames by second). The system is robust to partial occlusions and it can detect and track both persons in any part of the scenario providing their position in all frames. The circles with the line inside is the motion direction (0 360 degrees) of each person, the pixels represent the Motion History of the human silhouettes, the large blobs are the estimated area

where each person is identified inside the scenario and the small squares are the position of each fixed object (the computer, the books and the telephone).

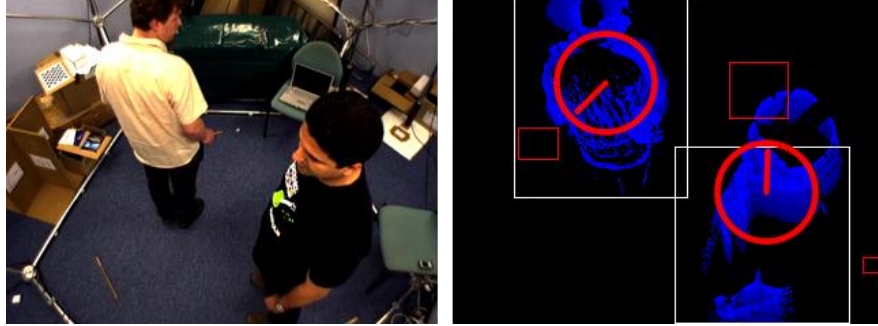


Fig. 5.5 Results of the computer vision system

5.12 Conclusion

This instance of the fusion mechanism contributed to validate the life cycle of the fusion system. Each channel represents a modality. Two devices were connected to capture signals, which are processed in the recognition phase. For each detection, the system creates a segment in the segmentation phase, delimiting useful data. Segments are processed and meanings are extracted from them. These meanings are annotated in the annotation phase. During the modality life cycle, an event-oriented architecture executes threads in parallel to perform high level fusion. Using the current proposal of the system, we could instantiate a fusion mechanism to:

- manage spatial relationships based on the fixed objects in the room;
- make semantic fusion of events not coinciding in time;
- achieve good results in speaker identification;
- synchronization between image and speech identification;
- create an open system to manage fusion between two or more modalities; and
- design the system so that each component can run in a separate machine due to the distribution mechanism interchanging data through a TCP/IP network.

However, we have more issues to solve in our future works. To name just a few, we should:

- implement an effective learning mechanism with long term memory to improve the recognition after several executions, by automatically increasing the training dataset;
- perform efficient decision making, even from information fragments, which could be achieved by giving more focus on complementarities and disambiguation of modalities;
- perform 3D video analysis in virtual worlds using the same strategy, but analyzing the behavior of virtual avatars in several situations; and
- add other modalities, e.g. eye gaze tracking, to evaluate the behavior of the mechanism and the scalability of the system.

Since the basis of the system is consolidated, these future works are oriented to increase the robustness of the solution, expand the possibilities of experimentation, and gradually improve the overall performance and reliability. However, that basis is not fixed and can be improved following the evolution of the field of multimodal applications, which is broadly discussed nowadays.

Chapter 6

Conclusion

We conclude this thesis summarizing the research and the contributions. According to our vision of future and the restrictions imposed by the scope of this research, we discuss future works in a short and long term perspective, as a continuation of the current contributions.

6.1 Summary of Contributions

We have defined in this thesis a multi-purpose, extensible and scalable framework for multimedia archiving to enhance the capacity of content description. This framework is supported by a pragmatic formal process that integrates different stages of multimedia processing. Its extensibility and multi-purpose design allowed its application on conceptually distinct problems. Its architecture enabled a scalable handling of multimedia content while storing, describing and retrieving.

Exploring the state of the art, we have identified gaps in multimedia archiving systems while supporting the segmentation and annotation of multimedia contents. We have noticed a rich literature of contributions offering support for multiple media types, more focused on storage capacity of the media database or on efficiency to extract media. When offering annotation capabilities, just a few went beyond the description of low-level features in order to be compared with Yasmim.

Considering the basic requirements, to the best of our knowledge we have not yet identified a work that addresses the contributions as covered by this research. We did not aim to cover all these requirements, but focus on the ones that would represent a distinguished contribution from the current state of the art. Following, we describe how we have addressed those requirements:

- **Multiple Media:** Yasmim is able to segment and annotate images, audios, animations, videos and 3D models. Within these media forms, there are also dozens of formats and the framework can virtually support most of them, inheriting this capability from the JavaTM platform, and not physically interfering in the file integrity. Even if the file is not supported by the JavaTM platform, the framework can simply store it for media clients developed in other technologies able to deal with special cases.
- **Feature Recognition:** this framework transfers the responsibility of feature recognition for the clients because this processing can be costly, thus impacting the performance of the system as a whole if done in the server (centralized). Therefore, the client applications presented in chapters 4 and 5 were responsible for recognition implementations, using several feature recognition techniques, such as automatic identification of user interface components, and recognition of multiple modalities for the purpose of fusion.
- **Segmentation:** in order to enhance the traditional representation of spatial, temporal and spatio-temporal segmentation, this framework implemented segments as vectorial structures, making it applicable not only for bitmap content but also for vectorial contents, such as graphics, animations and 3D, unifying its adoption on different media and segment types. In addition, segments can be mutually linked, creating a graph of connections and filling the gap of meanings floating in the limbo not yet segmented. It gives more freedom in comparison to approaches where segments are organized hierarchically or sequentially.
- **Annotation:** it provides a greater coverage of the descriptive formats that can be given to segments, which are: property, tagging, transcription, description, adhoc and domain concepts.
- **Distributed:** it provides REST web services widely available on the network that are accessible for clients developed in different languages, platforms and devices. In essence, every technology that has support for communication using sockets can access Yasmim's data and perform operations on the server, characterizing a modern approach of distribution, aligned with existing web architectures.
- **Multi-domain:** it allows the use of multiple ontologies to annotate segments and links. While observing the media, the user decides which ontology is more appropriate for the case, considering his/her domain of expertise.

Yasmim also upgrades the architecture of multimedia systems to a new paradigm, where every functionality is provided as web services, substantially increasing its applicability, which may vary from popular web browsers to complex industrial applications. Because those web services are resource-

oriented, media and data are accessible through URIs, using basic HTTP methods, allowing easy access even from mobile devices.

The most fundamental aspect is that the access to media descriptions is as easy as the access to the media in itself. It allows search engines to index media according to their strategies and goals, not relying on surrounding contents anymore. Therefore, in addition to its own search capabilities, Yasmim can also profit from search engines such as Google¹ and Bing².

We also have direct research contributions on the application of Yasmim. Chapters 4 and 5 are case studies, essentially distinct, using Yasmim to manage their media. The annotation of user interfaces is an alternative approach to several others that attempt to analyze user interfaces through the source code. Because new technologies for user interface implementations appear every day, it is practically impossible to keep the analysis of source code up-to-date. Using a simplified user interface model, together with segmentation and annotation techniques, screen shots seem to be the most comprehensive solution, covering even legacy systems and complex user interfaces [103].

The fusion of modalities were also a challenge and Yasmim fulfilled an important role, by managing the long term memory of the fusion mechanism. Every recorded video and audio were stored, segmented and annotated, becoming available for reasoning algorithms [66]. This research has motivated us to extend multimodal interaction to multimedia archiving and some experiments have already started [65], but are still in progress. OI components have been applied to implement multimodal interactions when performing four interactive tasks: 1) selection of segments, 2) vocal annotation, 3) query planning and 4) navigation. Since we are working with association of meanings and complex contents, we believe that this kind of application demands improved representativity of user intentions, which is sometimes difficult to represent using only keyboard and mouse.

6.2 Research Impact

The impact of the availability of Yasmim on the market would mainly benefit:

- *Scientific Community*: the research environment has been one of the main inspirations to develop Yasmim. It was thinking about the need for large datasets of media and how to organize them in an efficient way that inspired us to design this framework. This need came from challenging research projects from a local to an European level and we realized that a

¹ <http://www.google.com>

² <http://www.bing.com>

lab working with signal processing could count on Yasmim to store their research results in a long term perspective. Other multimedia labs would share the same vision and visualize potential applicability of the framework in their innovation process.

- *Industrial Companies*: the technology basis of this framework is open source, which allows its fast and wide adoption, which is strategically important for its dissemination. Yasmim can have a strong participation of researchers and entrepreneurs with the mission to coordinate, expand and integrate activities in the field of multimedia archiving. The quality of open source applications are increasing so fast that those companies are managing to reduce products' prices to keep the level of sales. The classical case about this scenario is the Microsoft Office Suite that is constantly changing their strategy because of the competition with OpenOffice, a free and open source alternative managed by Oracle, which has gained an important market share. Other companies just accept the open source movement not as a threat, but as an allied, building their solutions on top of high quality open source code and helping the community to improve it when possible and necessary for the natural evolution of their own applications. Making Yasmim the basis of industrial solutions would contribute to reduce overall costs and push the economy around the solution, implying in more opportunities and sharable knowledge bases.
- *Individuals*: this framework can be applied on the development of websites for the general public. Nowadays, individuals all over the web are relevant multimedia content contributors, and there is a lack of web sites supporting rich segmentation and annotation techniques.

6.3 Future Works

Future works reflect our vision about what can be extended over what has been done so far. We have short and long term expectations that can become part of our research plans in a near future.

6.3.1 Short Term

6.3.1.1 User Profile

The audio and visual aspects of multimedia contents register a lot of meanings that are perceived differently by people's diversity. Because of this natural and

abstract human interpretation, multimedia archiving systems, as many other categories of systems, are evolving on the direction of better considering those people who produce and consume media. There are plenty of techniques to analyze human aspects when interacting with systems. They rely on collected data about the user. For new users, almost nothing is known besides what is explicitly informed during the registration process. The amount of information is small because the registration must be short in order to not discourage the user. It limits the number of fields and questions, forcing the adoption of additional techniques to collect data from the user. A well known example of this scenario is the strategy used by the Amazon.com online store. The user preferences are collected according to his/her use of the system. After some minutes navigating on the website, it is possible to notice the emphasis given to contents related to what was previously seen. It keeps working even after leaving the website and going back later on. The personalization gradually improves as more traces of our preferences are left and more products are purchased. This content adaptation effectively pleases the user and contributes to a better user interaction, while helping the company to archive its goals. When applied on a multimedia archiving system, this strategy might gradually help users to consume and produce media efficiently. The more robust is the support for media content representation as well as the support for user profile representation, the more precise is the media archiving and retrieval results.

6.3.1.2 Advances on Digital Right Management

DRM is an essential piece on the multimedia archiving domain. Unfortunately, we could explore only the basics throughout this work due to the focus of the research. Beyond the usual copyright protection, DRM features could be extended to a lower granularity, assigning rights not only for the content as a whole, but also for segments and annotations. It avoids, for instance, fully presenting a movie containing violence to children, by taking away from them the right to access segments that delimit violent scenes. Annotations that are not appropriate for a specific geographic region, because they are written in another language, could be hidden in another country. This is also applicable for annotations that are created for different domains, where they can be hidden depending on the profile of the specialist.

6.3.2 Long Term

6.3.2.1 Social User Feedback

Taking the user feedback solution, we intend to give it a social perspective. As in many social networks, the participation of people generates considerable volume of data. These data can be analyzed, summarized and also used to call attention of those people who maintain the applications. This represents a new perspective into social networks for enterprise systems. Social software has been used to manage work activities [28], but have not yet been used as a channel for end-users to give their opinions about these systems as they perform their daily work. The opinion of end-users does matter because they are the real business performers. If they have a position about how their applications can be improved then they probably know how to become more productive, which is strategically interesting for the organization.

6.3.2.2 Query By Example

In some situations, the user has something in mind to find, but does not know exactly the right terms to search for. A common case is the search for audio content. Sometimes, the user knows the melody of a song, but does not know its name, neither the singer, band or parts of the lyrics. That is when query by example is applicable. In this case, the user can simply hum the melody and the resulting sound waves would be enough to identify similar songs. Another application would help people with disabilities to express themselves using gestures or movements to exemplify their intentions. For instance, a circular gesture with the hands would be interpreted as a ball and content annotated with ball or similar objects would be retrieved.

References

1. 1321, R.: The md5 message-digest algorithm. <http://tools.ietf.org/html/rfc1321> (1992)
2. Akman, V.: From discourse to logic: Introduction to model-theoretic semantics of natural language, formal logic and discourse representation theory. *Computational Linguistics* **21**(2), 265–268 (1995)
3. Antoniou, G., van Harmelen, F.: A Semantic Web Primer (Cooperative Information Systems). The MIT Press (2004)
4. de recherche Multimodale dans le contenu Audiovisuel, I.I.: Spécifications techniques du logiciel irma. Tech. Rep. 1-2009, Université catholique de Louvain (2009)
5. Blanning, R.W.: Data management and model management: a relational synthesis. In: ACM-SE 20: Proceedings of the 20th annual Southeast regional conference, pp. 139–147. ACM, New York, NY, USA (1982). DOI <http://doi.acm.org/10.1145/503896.503921>
6. Bloehdorn, S., Petridis, K., Saathoff, C., Simou, N., Tzouvaras, V., Avrithis, Y.S., Handschuh, S., Kompatsiaris, I., Staab, S., Strintzis, M.G.: Semantic annotation of images and videos for multimedia analysis. In: A. Gómez-Pérez, J. Euzenat (eds.) *ESWC, Lecture Notes in Computer Science*, vol. 3532, pp. 592–607. Springer (2005)
7. Bormans, J., Hill, K.: Mpeg-21 overview - version 5. Tech. Rep. 21000, ISO/IEC, Shanghai (2002)
8. Bouchet, J., Nigay, L.: Icare: A component-based approach for the design and development of multimodal interfaces. In: CHI '04 Extended Abstracts on Human Factors in Computing Systems, pp. 1325–1328. Vienna (2004)
9. Bulterman, D.C.A.: User-centered control within multimedia presentations. *Multimedia Syst.* **12**(4-5), 423–438 (2007)
10. C. Marshall, C., Rossman, G.: *Designing Qualitative Research*. Sage, London (1989)
11. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A unifying reference framework for multi-target user interfaces. *Interacting with Computers* **Vol. 15**(No. 3), 289–308 (2003)
12. Campbell, R.H., Islam, N., Madany, P.: Choices, frameworks and refinement. *Computing Systems* **5**(3), 217–257 (1992)
13. Chotmanee, A., Wuwongse, V., Anutariya, C.: A schema language for mpeg-7. In: E. Lim, S. Foo, C. Khoo, H. Chen, E. Fox, S. Urs, T. Costantino (eds.) *Digital Libraries: People, Knowledge, and Technology, Lecture Notes in Computer Science*, vol. 2555, pp. 153–164. Springer Berlin / Heidelberg (2010)

14. Cole, L., Austin, D.: Visual object recognition using template matching. In: Proceedings of Australasian Conference on Robotics and Automation (2004)
15. Corporation, I.: Open source computer vision library - opencv. <http://www.intel.com/technology/computing/opencv/index.htm> (2009)
16. Corporation, O.: Glassfish application server. <https://glassfish.dev.java.net/> (2010)
17. Corporation, O.: The java virtual machine specification. <http://java.sun.com/docs/books/jvms/> (2010)
18. Corporation, O.: Mysql database server. <http://mysql.com> (2010)
19. Curran, J., Clark, S., Bos, J.: Linguistically motivated large-scale nlp with c& c and boxer. In: Proceedings of the ACL 2007 Demonstrations Session, pp. 29–32 (2007)
20. Cusano, C., Santini, S., Schettini, R.: On the cooperative creation of multimedia meaning. In: 4th International conference on Semantics and digital media technologies. Graz, Austria (2009)
21. D., R., B., M.: A master-slaves volumetric framework for 3d reconstruction from images. In: Proceedings of the SPIE'07, vol. Vol. 6491. San Jose, USA (2007)
22. Dai, P., Tao, L., Xu, G.: Dynamic context driven human detection and tracking in meeting scenarios. In: A. Ranchordas, H. Araújo, J. Vitrià (eds.) VISAPP (Special Sessions), pp. 31–40. INSTICC - Institute for Systems and Technologies of Information, Control and Communication (2007)
23. Davenport, T.H.: Process Innovation: Reengineering work through information technology. Harvard Business School Press, Boston, MA, USA (1993)
24. Davis J., B.A.: The representation and recognition of action using temporal templates. Tech. Rep. 402, MIT Media Lab Technical Report (1997)
25. Doller, M., Kosch, H.: The mpeg-7 multimedia database system (mpeg-7 mmdb). Journal of Systems and Software **81**(9), 1559–1580 (2008)
26. Encyclopedia, W.T.F.: Nosql. <http://en.wikipedia.org/wiki/NoSQL> (2010)
27. Engel, R., Pfleger, N.: SmartKom: Foundations of Multimodal Dialogue Systems. Springer, Berlin, Germany (2006)
28. Erol, S., Granitzer, M., Happ, S., Jantunen, S., Jennings, B., Johansson, P., Koshmider, A., Nurcan, S., Rossi, D., Schmidt, R.: Combining bpm and social software: Contradiction or chance? Journal of Software Maintenance and Evolution: Research and Practice **22**(3) (2010)
29. Fayad, M., Schmidt, D., Johnson, R.: Building Application Frameworks: Object-Oriented Foundations of Framework Design. Wiley & Sons (1999)
30. Fayad, M.E., Schmidt, D.C.: Special issue on object-oriented application frameworks. Communications of the ACM **40**(10) (1997)
31. Fielding, R.: Architectural styles and the design of network-based software architectures. Ph.D. thesis, University of California, Irvine (2000)
32. Foundation, A.: Couch db project. <http://couchdb.apache.org> (2010)
33. Foundation, F.S.: Gnu lesser general public license - lgpl. <http://www.gnu.org/licenses/lgpl.html> (2007)
34. Fowler, M.: UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2003)
35. Gaitanis, K., Vybornova, O., Gemo, M., Macq, B.: Multimodal high level fusion of input commands as a semantic goal-oriented cooperative process. In: In 12th International Conference Speech and Computer (2007)
36. Gantz, J., Reinsel, D.: As the economy contracts, the digital universe expands. IDC (2009)
37. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition **5**, 199–220 (1993)
38. Hamano, J., Torvalds, L.: Git. <http://git-scm.com/> (2010)

39. Hamilton, A.G.: Logic for Mathematicians. Cambridge University Press, Cambridge, UK (1978)
40. Handschuh, S.: Semantic annotation of resources in the semantic web. In: Semantic Web Services: Concepts, Technologies, and Applications, chap. 5, pp. 135–155 (2007). DOI 10.1007/3-540-70894-4_5
41. Hockenmaier, J., Steedman, M.: Cggbank: A corpus of ccg derivations and dependency structures extracted from the penn treebank. *Comput. Linguist.* **33**(3), 355–396 (2007). DOI <http://dx.doi.org/10.1162/coli.2007.33.3.355>
42. Inc., G.: Google images. <http://www.google.com/images>
43. Inc., S.M.: Project grizzly. <https://grizzly.dev.java.net/> (2009)
44. Interconnection, I.I.P.S.O.S.: Lotos - a formal description technique based on temporal ordering of observational behavior. Tech. Rep. ISO/IS 8807, ISO Central Secretariat (1988)
45. Iqbal, R., Shirmohammadi, S., Saddik, A.E.: Saddik, a framework for mpeg-21 dia based adaptation and perceptual encryption of h.264 video. In: Proc. of SPIE/ACM MMCN (2007)
46. Johnson, R.E., Foote, B.: Designing reusable classes. *Journal of Object-Oriented Programming* **1**(2), 22–35 (1988)
47. Karat, J., Vanderdonckt, J., Karat, C.M., Blom, J.O. (eds.): Designing personalized user experiences in eCommerce. Kluwer Academic Publishers, Norwell, MA, USA (2004)
48. Kompatsiaris, Y., Hobson, P.: Introduction to semantic multimedia. In: Semantic Web Services: Concepts, Technologies, and Applications, chap. 1, pp. 3–13 (2008). DOI 10.1007/978-1-84800-076-6
49. Lalanne, D., Nigay, L., Palanque, P.A., Robinson, P., Vanderdonckt, J., Ladry, J.F.: Fusion engines for multimodal input: a survey. In: J.L. Crowley, Y.A. Ivanov, C.R. Wren, D. Gatica-Perez, M. Johnston, R. Stiefelwagen (eds.) ICMI, pp. 153–160. ACM (2009)
50. Lawson, J.Y.L., Al-Akkad, A.A., Vanderdonckt, J., Macq, B.: An open source workbench for prototyping multimodal interactions based on off-the-shelf heterogeneous components. In: Proceedings of the EICS'09. Pittsburgh, USA (2009)
51. Lawson, J.Y.L., Macq, B.: Openinterface platform for multimodal applications prototyping. In: ICASSP Show & Tell 2008. Las Vegas, USA (2008)
52. Lehman, J.F., Laird, J., Rosenbloom, P.: A gentle introduction to soar, an architecture for human cognition. In: In S. Sternberg & D. Scarborough (Eds), *Invitation to Cognitive Science*. MIT Press (1996)
53. Li, L.J., Socher, R., Li, F.F.: Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In: CVPR, pp. 2036–2043. IEEE (2009)
54. Lie, A., Grythe, K., Balasingham, I.: On the use of the mpeg-21 framework in medical wireless sensor networks. In: Applied Sciences on Biomedical and Communication Technologies, 2008. ISABEL '08. First International Symposium on, pp. 1–5 (2008). DOI 10.1109/ISABEL.2008.4712591
55. Limbourg, Q., Vanderdonckt, J.: Usixml: A user interface description language supporting multiple levels of independence. In: M. Matera, S. Comai (eds.) ICWE Workshops, pp. 325–338. Rinton Press (2004)
56. Limbourg, Q., Vanderdonckt, J.: Multi-path transformational development of user interfaces with graph transformations. In *Human-Centered Software Engineering: Software Architectures and Model-Driven Integration* **Vol. II** (2007)
57. Liu, J.S., Tseng, M.H., Huang, T.K.: Mediating team work for digital heritage archiving. In: JCDEL, pp. 259–268. ACM (2004)
58. López, F., Sanchez, J.M.M., García, N.N.: Cain-21: An extensible and metadata-driven multimedia adaptation engine in the mpeg-21 framework. In: T.S. Chua, Y. Kompatsiaris, B. Mérialdo, W. Haas, G. Thallinger, W. Bailer (eds.) *SAMT, Lecture Notes in Computer Science*, vol. 5887, pp. 114–125. Springer (2009)

59. catholique de Louvain, U.: Interface de recherche multimodale dans le contenu audiovisuel - irma. <http://www.tele.ucl.ac.be/view-project.php?name=IRMA> (2010)
60. Lux, M.: Caliph & emir: Mpeg-7 photo annotation and retrieval. In: W. Gao, Y. Rui, A. Hanjalic, C. Xu, E.G. Steinbach, A. El-Saddik, M.X. Zhou (eds.) *ACM Multimedia*, pp. 925–926. ACM (2009)
61. Martin, J.C.: Tycoon: Theoretical framework and software tools for multimodal interfaces (1998). URL citeseer.ist.psu.edu/martin98tycoon.html
62. Martin, J.C., Kipp, M.: Annotating and measuring multimodal behaviour - tycoon metrics in the anvil tool (2002)
63. Martínez, J.: Mpeg-7 overview - version 10. Tech. Rep. 15938, ISO/IEC, Palma de Mallorca (2004)
64. Mendonça, H.: Hildeberto's blog. <http://www.hildeberto.com>
65. Mendonça, H., Lawson, J.Y.L., Macq, B.: Interface of multimodal searching on multimedia archives. In: *IWSSIP 2010: 17th International Conference on Systems, Signals and Image Processing*, pp. 522–525 (2010)
66. Mendonça, H., Lawson, J.Y.L., Vybornova, O., Macq, B., Vanderdonckt, J.: A fusion framework for multimodal interactive applications. In: *ICMI-MLMI '09: Proceedings of the 2009 international conference on Multimodal interfaces*, pp. 161–168. ACM, New York, NY, USA (2009). DOI <http://doi.acm.org/10.1145/1647314.1647344>
67. of Michigan, U.: Soar. <http://sitemaker.umich.edu/soar/home> (2008)
68. Montero, F., Lopez-Jaquero, V.: Idealxml: An interaction design tool. In: *Proceedings of the CADUI'2006* (2006)
69. Mori, G., Paterno, F., Santoro, C.: Ctte: Support for developing and analyzing task models for interactive system design. pp. 797–813 (2002)
70. Multitel: Mediatic/3d media. <http://mediatic.multitel.be/platforms/3dmedia.html> (2010)
71. Natsev, A., Smith, J.R., Tesic, J., Xie, L., Yan, R.: Ibm multimedia analysis and retrieval system. In: J. Luo, L. Guan, A. Hanjalic, M.S. Kankanhalli, I. Lee (eds.) *CIVR*, pp. 553–554. ACM (2008)
72. Nielsen, J.: *Usability Engineering*. MIT Academic Press, Cambridge, MA (1993)
73. Nilsson, M.: Id3 tag version 2.3.0. <http://www.id3.org/id3v2.3.0.html> (1999)
74. Oliver, J.: *Jamie's Ministry of Food: Anyone Can Learn to Cook in 24 Hours*. Michael Joseph (2008)
75. OMG, O.M.G.: Model driven architecture. <http://www.omg.org/mda/> (2007)
76. OMG, O.M.G.: Unified modeling language. <http://www.uml.org> (2007)
77. OMG, O.M.G.: Business process modeling notation specification. <http://www.omg.org/spec/BPMN/1.2/> (2009)
78. Papageorgiou, H., Prokopoulos, P., Protopapas, A., Carayannis, G.: Multimedia indexing and retrieval using natural language, speech and image processing methods. In: *Multimedia Content and the Semantic Web*, pp. 279–297. John Wiley & Sons (2005)
79. Parekh, R.: *Principles of Multimedia*, 1 ed. edn. Tata McGraw Hill, New Delhi (2006)
80. Pfleger, S.L., Atlee, J.M.: *Software Engineering: Theory and Practice*, 4 edn. Prentice Hall (2009)
81. Pfleger, N.: Context based multimodal fusion. In: *Proceedings of the 6th international conference on Multimodal interfaces* (2004)
82. Poppe, C., Keukelaere, F.D., Zutter, S.D., Bruyne, S.D., Neve, W.D., de Walle, R.V.: Predictable processing of multimedia content, using mpeg-21 digital item processing. In: H.H.S. Ip, O.C. Au, H. Leung, M.T. Sun, W.Y. Ma, S.M. Hu (eds.) *PCM, Lecture Notes in Computer Science*, vol. 4810, pp. 549–558. Springer (2007)

83. Process, J.C.: Jsr 51: New i/o apis for the javatm platform. <http://jcp.org/en/jsr/detail?id=51> (2002)
84. Process, J.C.: Jsr 311: Jax-rs: The javatm api for restful web services. <http://jcp.org/en/jsr/summary?id=311> (2009)
85. Process, J.C.: Jsr 317: Javatm persistence 2.0. <http://jcp.org/en/jsr/summary?id=317> (2009)
86. Purchase, H., Naumann, D.: The tomus model of multimedia: an empirical investigation. In: Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications. Association for the Advancement of Computing in Education (2000)
87. Reynolds, D., Rose, R.: Robust text-independent speaker identification using gaussian mixture speaker models. *Speech and Audio Processing, IEEE Transactions on* **Vol. 3**(No. 1), 72–83 (1995)
88. Riehle, D.: Framework design: A role modeling approach. Ph.D. thesis
89. Rinaldi, F., Dowdall, J., Hess, M., Ellman, J., Zarri, G.P., Persidis, A., Bernard, L., Karanikas, H.: Multilayer annotations in parmenides. In: K-CAP2003 Workshop on Knowledge Markup and Semantic Annotation. ACM, Sanibel, Florida, USA (2003)
90. Russell, B.C., Torralba, A.: Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision* **77**(1–3), 157–173 (2008)
91. Russell, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*, 2 ed. edn. Prentice Hall, New Jersey (2002)
92. Sanchez, J.M.M., Koenen, R., Pereira, F.: Mpeg-7: The generic multimedia content description standard, part 1. *IEEE MultiMedia* **9**(2), 78–87 (2002)
93. Santini, S.: Multimedia semiotics. In: *Encyclopedia of Multimedia* (2006)
94. Schenk, S., Saathoff, C., Staab, S., Scherp, A.: Semaplorer - interactive semantic exploration of data and media based on a federated cloud infrastructure. *J. Web Sem.* **7**(4), 298–304 (2009)
95. Schmidt, D.C.: Guest editor's introduction: Model-driven engineering. *Computer* **39**(2), 25–31 (2006). DOI <http://doi.ieeecomputersociety.org/10.1109/MC.2006.58>
96. Schrijver, D.D., Neve, W.D., Wolf, K.D., Sutter, R.D., de Walle, R.V.: An optimized mpeg-21 bsd framework for the adaptation of scalable bitstreams. *J. Visual Communication and Image Representation* **18**(3), 217–239 (2007)
97. Shapiro, L.G., Stockman, G.C.: *Computer vision*. Prentice Hall (2001)
98. Sheppard, N.P.: On implementing mpeg-21 intellectual property management and protection. In: M. Yung, A. Kiayias, A.R. Sadeghi (eds.) *Digital Rights Management Workshop*, pp. 10–22. ACM (2007)
99. Shoham, Y., Leyton-Brown, K.: *Multiagent Systems*. Cambridge, New York (2009)
100. Slota, R., Kosch, H., Nikolow, D., Pogoda, M., Breidler, K., Podlipnig, S.: Mmsrs - multimedia storage and retrieval system for a distributed medical information system. In: *HPCN Europe*, pp. 517–524 (2000)
101. Sofokleous, A.A., Angelides, M.C.: Dcaf: An mpeg-21 dynamic content adaptation framework. *Multimedia Tools Appl.* **40**(2), 151–182 (2008)
102. Souchon, N., Vanderdonckt, J.: A review of xml-compliant user interface description languages. In: J.A. Jorge, N.J. Nunes, J.F. e Cunha (eds.) *DSV-IS, Lecture Notes in Computer Science*, vol. 2844, pp. 377–391. Springer (2003)
103. Sousa, K.S., Mendonça, H., Vanderdonckt, J.: A model-driven approach to align business processes with user interfaces. *J. UCS* **14**(19), 3236–3249 (2008)
104. Spaniol, M., Klamma, R., Waitz, T.: Mecca-learn: A community based collaborative course management system for media-rich curricula in the film studies. In: R.W.H. Lau, Q. Li, R. Cheung, W. Liu (eds.) *ICWL, Lecture Notes in Computer Science*, vol. 3583, pp. 131–143. Springer (2005)

105. Stanford University, D.o.C.S.: Protégé platform. <http://protege.stanford.edu> (2009)
106. Tidwell, J.: Designing Interfaces. O'Reilly, Beijing (2006)
107. Torralba, A., Fergus, R., Freeman, W.: Labelme: The open annotation tool. <http://labelme.csail.mit.edu/>
108. Town, C.: Multi-sensory and multi-modal fusion for sentient computing. *International Journal of Computer Vision* **71**(2), 235–253 (2007)
109. Trevisan, D., Nicolas, V., Macq, B., Nedel, L.: Medicalstudio: A medical component-based framework. In: Workshop de Informática Médica - WIM (2007)
110. Tsekeridou, S., Kokonozi, A., Stavroglou, K., Chamzas, C.: Mpeg-7 based music metadata extensions for traditional greek music retrieval. In: B. Günsel, A.K. Jain, A.M. Tekalp, B. Sankur (eds.) *MRCS, Lecture Notes in Computer Science*, vol. 4105, pp. 426–433. Springer (2006)
111. Vacura, M., Svátek, V., Saathoff, C., Franz, T., Troncy, R.: Describing low-level image features using the comm ontology. In: ICIP, pp. 49–52. IEEE (2008)
112. Vanderdonckt, J.: A mda-compliant environment for developing user interfaces of in-formation systems. In: In Proceedings of CAiSE'05, *Lecture Notes in Computer Science*, vol. 3520. Springer-Verlag, Berlin (2005)
113. Velte, T., Velte, A., Elsenpeter, R.C., Elsenpeter, R.: Cloud Computing: A Practical Approach. McGraw Hill Professional (2009)
114. Vybornova, O., Mendonça, H., Lawson, L., Macq, B.: High level data fusion on a multimodal interactive application platform. In: Proceedings of IEEE ISM'08. Berkeley, USA (2008)
115. W3C: Resource description framework - rdf. <http://www.w3.org/TR/PR-rdf-syntax/> (1999)
116. W3C, W.W.W.C.: Owl 2 web ontology language document overview. <http://www.w3.org/TR/2009/REC-owl2-overview-20091027/> (2009)
117. Walker, W.: Sphinx-4: A flexible open source framework for speech recognition. Tech. Rep. SMLI TR2004-0811, Sun Microsystems Inc. (2004)
118. Whittaker, J.A.: Exploratory Software Testing. Pearson Education, Boston, MA (2010)
119. Zou, X., Bhanu, B.: Tracking humans using multi-modal fusion. In: CVPR'2005, IEEE Computer Society Conference on. San Diego, USA (2005)
120. Zupan, B., Bohanec, M., Bratko, I., Demsar, J.: Machine learning by function decomposition. In: D.H. Fisher (ed.) *ICML*, pp. 421–429. Morgan Kaufmann (1997)

Appendix A

List of Publications

This research has been validated since 2008, through the publication of scientific papers in several international conferences and also in an international journal. The publications are organized according to the chapter that they are contributing for. Figure A.1 depicts those chapters and their respective publications, which are represented by numbers, according to the list in the sequence.

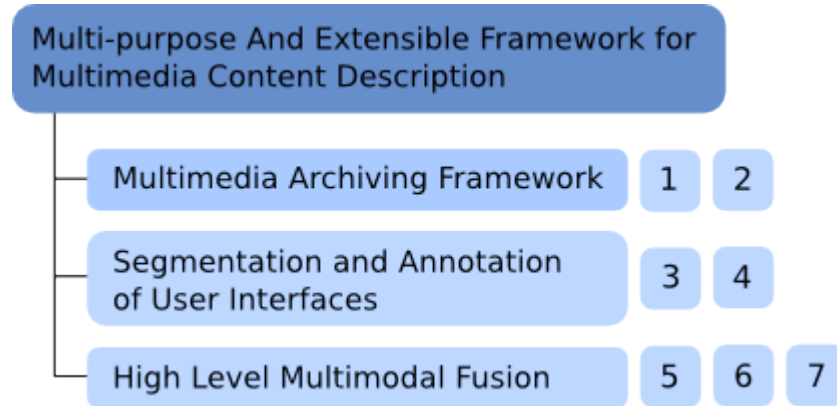


Fig. A.1 Reference Map

Chapter 3 has two related publications:

1. The paper published in the 17th International Conference on Systems, Signals and Image Processing - (IWSSIP'2010) introduces the framework as a whole, considering the types of segments and annotations. The experimentation was made with a Yasmim client that supports multimodal

interaction. This client enables the segmentation and annotation of media using advanced user interaction techniques.

Mendonça, Hildeberto; Lawson, Lionel; Macq, Benoit. **Interface of Multimodal Searching on Multimedia Archives**. In Proceedings of the IWSSIP 2010, Rio de Janeiro, Brazil, 2010. pp 522 - 525.

2. Yasmim would be a simple multimedia archiving framework if the work done for managing user interface models using web services had not been considered as the basis for Yasmim's distributed architecture. The obtained results were finally presented at the 1st International Workshop on User Interface Extensible Markup Languages - UsiXML'2010, and the evolution of that architecture was adopted by Yasmim to offer all its functionalities as web services, allowing heterogeneous applications to access a common multimedia repository.

Mendonça, Hildeberto; Sousa, Kenia; Vanderdonckt, Jean. **Service-Oriented Architecture for Supporting Collaborative User Interface Development**. In Proceedings of the 1st International Workshop on User Interface Extensible Markup Language UsiXML'2010, Berlin, Germany. June, 2010. pp 99 - 109.

Chapter 4, segmentation and annotation of user interfaces, is the first case study used to validate Yasmim. Two papers have built the basis for this application. They are:

3. The work published in the 3th IEEE International Workshop on Business-driven IT Management (BDIM'2008), reports the study of the impact that business processes may have over user interfaces of systems. The analysis of the impact helps to identify how processes are executed through information systems. An important result was that the feedback of the user while performing his/her task has a considerable value to improve the process and advanced and intuitive ways to collect this feedback are needed.

Sousa, Kenia; Mendonça, Hildeberto; Vanderdonckt, Jean. **Addressing the Impact of Business Process Changes on Software User Interfaces**. In Proceedings of the BDIM'08, Salvador, Brazil, 2008. pp. 11 - 20.

4. The idea was extended to consider two directions: the impact from the business process to the user interface and from the user interface to the business process. The traceability of the impacts was possible due to the use of formal models to represent processes, user interactions and user interfaces. After building this basis, it was possible to implement the segmentation and annotation of user interfaces to provide an intuitive way to collect the user feedback, which is the main contribution of chapter 4.

Sousa, Kenia; Mendonça, Hildeberto; Vanderdonckt, Jean. **A Model-Driven Approach to Align Business Processes With User Interfaces**. Journal of Universal Computer Science - J.UCS. Vol. 14, No. 19, 2008. pp. 3236 - 3249.

Chapter 5, high level multimodal fusion, is the second and more extensive case study in which Yasmim was applied. Three papers were published on this research. They are:

5. The work on multimodal fusion was performed during the eNTERFACE'08 workshop in Paris. During a whole month, five researchers were fully dedicated to implement this experiment. The paper published in the 13th International Conference Speech and Computer (SPECOM'2009) proposes an architecture for a system that needs to perform multimodal fusion, considering Yasmim as the storage of media, segments and annotations.

Vybornova, Olga; Mendonça, Hildeberto; Neigberg, Daniel; Gomez, David; Shen, Ao. **An Architecture for Multimodal Semantic Fusion**. 13th International Conference Speech and Computer - SPECOM'09, 2009. pp 123 - 129.

6. Afterwards, more research was done to consolidate the results of the eNTERFACE'08 workshop and the results were presented in a doctoral consortium in the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems.

Mendonça, Hildeberto; Macq, Benoit. **High Level Data Fusion On A Multimodal Interactive Application Platform**. In Proceedings of the EICS'09 Doctoral consortium, Pittsburgh, USA, 2009. pp 333 - 336.

7. Finally, the complete and mature research on multimodal fusion was presented in the 11th International Conference on Multimodal Interfaces in a special session dedicated to fusion engines.

Mendonça, Hildeberto; Lawson, Lionel; Vybornova, Olga; Macq, Benoit; Vanderdonckt, Jean. **A Fusion Framework for Multimodal Interactive Applications**. In Proceedings of the ICMI-MLMI 2009, Cambridge, USA, 2009. pp 161 - 168.

Appendix B

Access to Yasmim's Source Code

Yasmim is a open source software under GNU Lesser General Public License (LGPL) license [33]. It is a permissive license that allows the use of the code in other proprietary software, stimulating the economical exploitation of the project. LGPL is more suitable for libraries instead of programs. Since Yasmim is a library of services, this license is more appropriate for its purpose.

B.1 Requirements

Yasmim's source code is hosted in the GitHub Social Coding Website. The address of the project is: <http://www.github.com/htmfilho/Yasmim>. The code has read-only access through the website and read-write access using a Git client. Git is a distributed version control system for large software projects [38]. It gives a complete local history of the project and the merge of these distributed repositories with the central repository. It is considered one of the fastest version control system on the market and the one with the largest number of features available. We have chosen Git because Yasmim is intended to be developed in a distributed way, independent of the developers' location, which may demand a complex version control to meet different needs and trends.

Therefore, to have a copy of Yasmim's source code, the basic requirement is to have Git installed. Versions for several operating systems are available at this address: <http://git-scm.com/download>.

B.2 Getting a Working Copy of the Project

To get a working copy of the project, first it is necessary to create a local repository. Navigate until the place where the project source will be stored and use the following commands:

```
> mkdir yasmim
> cd yasmim
> git clone git://github.com/htmfilho/Yasmim.git
```

It downloads the full repository locally. To update this repository when there are updates available in the main repository, use the following command:

```
> git pull
```

B.2.1 Adding New Resources

The add command is used to add new resources to the repository. However, it is also used to add already versioned resources to the commit. It is useful when just some modified files should be committed (e.g. to solve a particular bug) before a bigger commit. It does not mean that every modified file of the bigger commit should be added manually. The command is:

```
> git add <file/directory path+name>
```

B.2.2 Committing

To commit changes to the branch use the following command line:

```
> git commit -a -m 'commit message.'
> git push origin master
```

Where master is the branch's name and the command `git push origin master` synchronizes the local repository with the remote repository.

B.3 Project Structure

The source code is organized as follows:

- `models` - UML and business process models files.
- `src` - this folder is the global source folder, but it contains only the configuration folder.
 - `conf` - contains configuration files describing how the application should be deployed on the application server.
- `yasmim-ejb` - sub-project that implements Yasmim's business logic to manipulate locations of resources, segments and annotations. This project does not manipulate files directly because operations are transactional and the time needed to process files might exceed the timeout of a transaction.
 - `setup` - configures the server with specific application's parameters.
 - `src` - contains the business logic source code and configuration files.
 - `conf` - configures the database connection pool access and identifies entity classes.
 - `java` - contains the business logic's source code organized in packages.
- `yasmim-war` - sub-project that implements Yasmim's web services interface. This layer manipulates files received from clients and retrieves files to clients. It also accesses the business logic to access data related with these files.
 - `src` - contains web services' source code and configuration files.
 - `conf` - configures the authentication and authorization of the web services, properties of the administration module and other minor properties of the web environment.
 - `java` - contains the web services' source code organized in packages.
 - `web` - contains the web files of the administration module.

Appendix C

Yasmim Deployment

This appendix explains how to configure and deploy Yasmim.

C.1 Requirements

Some basic software and frameworks are necessary to install Yasmim. They are:

- *JavaTM Development Kit 6 (JDK)*: this kit is used by developers to compile and execute JavaTM code. It is also needed by the application server because JSP pages are transformed in JavaTM Servlets classes that have to be automatically compiled to start providing web content. JDK is available at: <http://www.oracle.com/technetwork/java/index.html>.
- *JavaTM Runtime Environment 6*: or JavaTM Virtual Machine (JVM), executes compiled JavaTM classes (.class). Since the application server, its libraries and Yasmim are developed in JavaTM, they all need the JRE to run. JDK already contains a JRE to run recently compiled code, but it can also be installed separately. JRE is available for download at the same address as the JDK.
- *Glassfish Application Server V3*: is a middleware that runs and provides scalable services for applications. These services allow applications to support many concurrent users, connect to several data sources, interoperate with services provided by other middleware instances, expand the platform to scale the application, support several frameworks to give flexibility for developers and many other relevant features. Glassfish also accesses external web services, sends emails, provides framework for logging, user interface rendering, and manages multiple instances of the application server to optimize the use of available hardware, among other advantages.

- *MySQL 5.1 Database*: is a high performance database used to store references to resources, segments and security data. Only the business logic layer has access to this database, using the connection pool of the application server.
- *CouchDB 1.0*: is a high scalable document database used to store annotations. The business logic layer and the web service layer have access to this database, whose data are provided by REST web services.
- *Email Server*: is used by the application server to send emails. The ability to send emails is important for the system-user (e.g. alerts) and user-user (e.g. collaboration, discussion) communication. An email server provides asynchronous service, thus it can be available remotely.

C.2 Creating the Database

Yasmim uses MySQL database to store references to resources in the file system. We consider that MySQL 5.0 or superior is installed and running normally. We also consider that the path of the operating system, pointing to the bin folder where all MySQL commands are located, is configured. We are going to use the basics, avoiding specific tools that can accelerate the process, because it would demand more explanations than necessary. We start executing the command below to login in a MySQL section using the super user `root`.

```
> mysql -u root -p
```

It creates a client authenticated session to access MySQL. `-u` means that we are passing the user of the session in the command line and `-p` means that we want to type the password right after the command has been executed. The user `root` has enough rights to create the database, but we do not use it all the time. Once authenticated, we type the command below to create the database and a dedicated user for it:

```
mysql> create database yasmim;
mysql> create user yasmim_user identified by
      '(password)';
mysql> use yasmim;
mysql> grant all privileges on yasmim.*
to yasmim_user@%;
mysql> flush privileges;
```

The database is created, a new user is also created and finally we give all privileges for this user to operate the new database. To check if the database was appropriately created, we execute the following command:

```
mysql> show databases;
```

If the database yasmim is in the list then it was created successfully. We have to sign out from this root's session and open a new session for the new user to create the database structure, following the sequence of commands below:

```
mysql> quit;  
c:\> mysql -u yasmim_user -p  
mysql> use yasmim;  
mysql> source (file path)/database-create.sql;
```

The script.sql is available in Yasmim's source code, available at (project-location)/Yasmim/yasmim-ejb/src/conf. The user yasmimuser should be the user to access the application instead of the super user root.

C.3 Configuring Connection Pool

The connection pool manages the demand for database access, allowing the reuse of connections before destroying them. The database should be created and the MySQL driver correctly installed in the application server. We use Glassfish V3 as application server and, at this point, we consider that it is already installed and running normally. The necessary steps are:

1. We need the MySQL JDBC Driver, since Glassfish and its deployed applications are written in Java™. Drivers are available for download at <http://www.mysql.com/downloads/connector/j/>.
2. Download the driver file mysql-connector-java-<version>.jar and copy it to [gf-home]/glassfish/domains/domain1/lib/.
3. Restart Glassfish in order to make it load the new database driver.
4. Enter in the administration console at [http://\[server-name\]:4848/](http://[server-name]:4848/) and go to Resources/JDBC/Connection Pools.
5. Create a new connection pool with the name yasmimPool, select the resource type javax.sql.ConnectionPoolDataSource, select the database vendor MySQL and click on next.

6. Select the datasource classname `org.postgresql.ds.PGConnectionPoolDataSource` and inform the following additional properties:
 - a. `DatabaseName=yasmim`
 - b. `Password=*****`
 - c. `PortNumber=3306` (this is the default port but the server administrator can change it for security reasons)
 - d. `ServerName=[server-name—ip]`
 - e. `User=yasmim_user`
7. Click on *Finish* to save the new connection pool.
8. Go to the list of connection pools again and select the new one that has just been created.
9. Click on *Ping* to check if the connection was correctly configured. The message *Ping Succeeded* means that the connection is working fine.
10. In order to be able to use this connection pool in JEE applications, we have to create a JNDI name for it. Go to `Resources/JDBC/JDBC Resources`.
11. Click on *New* and set the JNDI Name `jdbc/yasmim`, select the connection pool created above and click on *OK* to finish.

C.4 Configuring Authentication Realm

The authentication realm is a framework that implements most of the complexity related to authentication and authorization, minimally interfering in the application implementation. Since we are using a JEE6 Application server, we are going to explore it using the security infrastructure provided by Glassfish.

C.4.1 JDBC Realm

This is a configuration entirely made in the application server. It may demand some time, but at least Yasmim is going to be free from the basic security complexity. Yasmim uses JDBC Realm because we stored the user data in a database, which could be a third party database (single sign-on

for multiple applications in the context of an organization) or the database of the application (users and groups embedded to be independent from the organizational context).

Figure C.1 proposes a data structure for users and groups. It follows basically the following principle: a group contains several users and a user can be part of several groups. We created an N-N relationship between the tables `access_group` and `user_account`, using the intermediary table `user_group`.

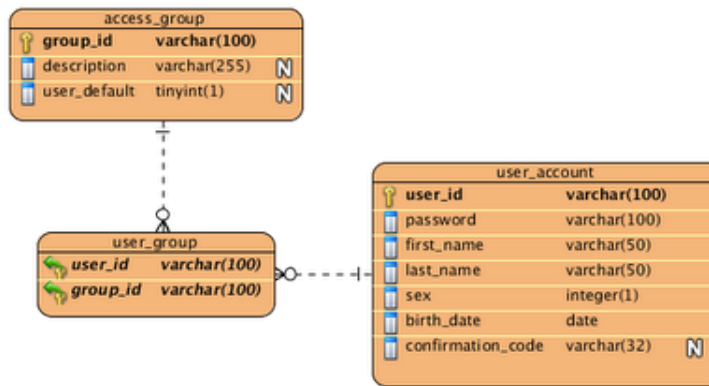


Fig. C.1 Database model of users and groups.

The script to create this structure is available in the repository. The beginning of the script creates the structure above in a MySQL database.

The following sql script considers the model above and they are used to populate the tables for test purposes. The first one inserts 2 users in the database: John Smith and Mary Allen. Their password is secret but it is not readable because a MD5 algorithm was applied on it. The realm needs encoded passwords because we have specified the MD5 [1] as the diggest algorithm in the realm configuration detailed later on. The column `confirmation_code` is not relevant now, so we defined it as null.

```

insert into user_account values
('johnsmith@acme.com','Xr4ilOzQ4PCOq3aQ0qbuaQ==',
 'John','Smith',1,'1980-02-09',null);
insert into user_account values
('maryallen@acme.com','Xr4ilOzQ4PCOq3aQ0qbuaQ==',
 'Mary','Allen',0,'1976-05-10',null);
  
```

Then, we insert the basic groups *admins* and *users*. The group *users* was defined as the default for registered users, but this attribute is not relevant either here.

```
insert into access_group values
    ('admins', 'Administrative Group', null);
insert into access_group values
    ('users', 'Default User Group', 1);
```

Finally, we associate the users with the groups. Mary Allen is part of the groups *admins* and *users* and John Smith is part of the group *users*. So, Mary can do everything John can do, but John cannot do everything that Mary can.

```
insert into user_group values
    ('admins', 'maryallen@acme.com');
insert into user_group values
    ('users', 'maryallen@acme.com');
insert into user_group values
    ('users', 'johnsmith@acme.com');
```

Follow the steps below to configure the JDBC Realm:

1. Go to the Glassfish admin console ([http://\[server-name\]:4848/](http://[server-name]:4848/)), navigate to Configuration / Security / Realms and press *New...*;
2. Name: yasmim-realm - it is used to refer to this realm in the application (tag realm-name in the file web.xml);
3. Class Name: com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm;
4. Properties specific to this Class (using the model above as a reference):
 - a. JAAS Context: jdbcRealm - identifier for the login module;
 - b. JNDI: jdbc/yasmim - the data source JNDI name pointing to the connection pool where the tables are available;
 - c. User Table: user_account - user table according to the model above;
 - d. User Name Column: user_id - column where the user name is stored;
 - e. Password Column: password - column where the password is stored;
 - f. Group Table: user_group - notice that the group table is not the access-group, but the intermediary table user_group. It is used to store further details about the groups;
 - g. Group Name Column: group_id - this is the foreign key to the access-group table, thus this is the id of the group;

- h. Digest Algorithm: MD5 - the password should not be saved as it is. It informs to the realm that the password was scrambled by the MD5 algorithm; and
- i. Encoding: Base64 - the encoding used by the MD5 algorithm.

C.4.2 Application Configuration

Now, we are going to configure the application to use the realm. We have to configure the files `web.xml` and `sun-web.xml`. An example of `web.xml` configured for this purpose is available in the Yasmim repository. Here, some comments about the configuration:

Since some pages are going to be protected, we need an access denied page to alert the user:

```
<error-page>
  <error-code>403</error-code>
  <location>/access_denied.xhtml</location>
</error-page>
```

On the application side, the access to resources of the system is given to roles, that can be assumed by users and groups. Roles are declared this way:

```
<security-role>
  <description/>
  <role-name>admin</role-name>
</security-role>
<security-role>
  <description/>
  <role-name>user</role-name>
</security-role>
```

This security constraint states that only users with the role *admin* can access the content of the `/admin` folder. These users can perform *GET* and *POST* http methods only.

```
<security-constraint>
  <display-name>Admin Constraints</display-name>
  <web-resource-collection>
    <web-resource-name>Administrators</web-res....>
  <description/>
  <url-pattern>/admin/*</url-pattern>
```

```

        <http-method>GET</http-method>
        <http-method>POST</http-method>
    </web-resource-collection>
    <auth-constraint>
        <role-name>admin</role-name>
    </auth-constraint>
</security-constraint>

```

This second constraint states that only users with the role *user* can access the content of the */media* folder. The tag *url-pattern* is flexible enough to consider several complex situations, but we may find some difficulty if the name of the files and folders are not well organized.

```

<security-constraint>
    <display-name>Users Constraints</display-name>
    <web-resource-collection>
        <web-resource-name>Users</web-resource-name>
        <description/>
        <url-pattern>/media/*</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
    </web-resource-collection>
    <auth-constraint>
        <role-name>user</role-name>
    </auth-constraint>
</security-constraint>

```

Now, we define which authentication method is presented to the user. We chose the most common one, which is the *FORM* method. *yasmim-realm* is the name of the realm we have just created. */login.xhtml* is the authentication page and */login_error.xhtml* is shown in case of authentication failure.

```

<login-config>
    <auth-method>FORM</auth-method>
    <realm-name>yasmim-realm</realm-name>
    <form-login-config>
        <form-login-page>/login.xhtml
        </form-login-page>
        <form-error-page>/login_error.xhtml
        </form-error-page>
    </form-login-config>
</login-config>

```

In addition to the *web.xml* file, the *sun-web.xml* file is used to associate the application roles with the user groups. It is a practical way to associate

roles to several users at the same time. The file is located in the *WEB-INF* folder, together with *web.xml*, and its content is shown below:

```
<sun-web-app error-url="">
  <context-root>/yasmim</context-root>
  <security-role-mapping>
    <role-name>admin</role-name>
    <group-name>admins</group-name>
  </security-role-mapping>
  <security-role-mapping>
    <role-name>user</role-name>
    <group-name>users</group-name>
    <group-name>admins</group-name>
  </security-role-mapping>
</sun-web-app>
```

In this case, the role *admin* is associated with the group *admins* and the role *user* is associated with the groups *admins* and *users*.

C.4.3 Implementation of the Authentication Page

There are, at least, three pages to consider: *login.xhtml*, *login_error.xhtml* and *access_denied.xhtml*. The *login.xhtml* page was declared in the *web.xml* file as the page that collects the authentication data from the user through a standardized form. The form contains the *username* and *password* fields and the submit command to post those data to the server. The action of the form and the name of each field are pre-defined. Look at the code below:

```
<form method="post" action="j_security_check">
  <h:panelGrid columns="2">
    <h:outputLabel for="j_username" value="E-mail"/>
    <input type="text" name="j_username"/>

    <h:outputLabel for="j_password" value="Password"/>
    <input type="password" name="j_password"/>

    <h:outputText value=" " />
    <h:panelGrid columns="2">
      <input type="submit" name="sbmt" value="Signin"/>
      <h:button outcome="index" value="Cancel"/>
    </h:panelGrid>
  </h:panelGrid>
```

```
</form>
```

This is a JSF 2.0 page with Facelets, but it can also be a simple HTML page. The important thing is the form action, which is `j_security_check`, and the username and password fields, which are named respectively as `j_username` and `j_password`. When the user submits the data, the web container handles it in a special way without interfering in the logic of the application.

According to the authentication and authorization data in the database and what was declared in the application configuration, once authenticated, Mary Allen has full access to the content of the folders *admin* and *media* and John Smith has access only to the content of the folder *media*.

C.5 Configuring JavaMail™

One of the main advantages of using an application server like Glassfish is to keep the application free of complex code, such as 1) manual control of database transactions; 2) database access configuration; 3) security authentication and authorization; 4) sending and receiving e-mail messages, among many other complexities that are non-functional requirements, consuming the time we would be spending on functional requirements.

An important differential of Glassfish V3 is its very rich and complete administration console. It is easy to use and to learn, which is one of the most important competitive advantages, since it contributes to reduce the maintenance cost. We are going to use the administrative console to configure a JavaMail™ resource, allowing Yasmim to send emails. Follow the steps below:

1. enter in the administrative console ([http://\[server-name\]:4848/](http://[server-name]:4848/));
2. go to *Resources / JavaMail™ Sessions*;
3. create a new JavaMail™ session and set the following mandatory properties:
 - a. JNDI Name: `mail/[email-account-name]`
 - b. Mail Host: `[smtp-server-address]`
 - c. Default User: the username to authenticate on the smtp server
 - d. Default Return Address: the address used by recipients to reply the message. Some servers require that this address should be the one used by the authenticated user to access his/her mailbox.

If the server does not request secure authentication, then the three steps above are enough to start using the new JavaMail™session, but a server without secure authentication is a very rare case nowadays. It is certainly needed to inform a password to login on the smtp server. In most cases, the server administrator also changes the default port of the smtp server, which forces us to explicitly inform the correct port. For these special needs we can use additional properties in the JavaMail™session. Follow the steps below:

1. Still on the JavaMail™session form, go to the Additional Properties section and add 3 more properties, which are:
 - a. mail.smtp.port: [port-number]
 - b. mail.smtp.auth: true
 - c. mail.smtp.password: *****
2. Click on *Save* to create the JavaMail™session.

The last step is how to use this new JavaMail™session in our applications to send emails. Using the JNDI name, we are going to inject the JavaMail™session in a Java™class, which could be a POJO of a pure web application, an EJB Session Bean, or any other type of class. See the code below for details:

```
public class UserAccountBsn {
    @Resource(name = "mail/[email-account-name]")
    private Session mailSession;

    public void sendMessage(UserAccount userAccount) {
        Message msg = new MimeMessage(mailSession);
        try {
            msg.setSubject("[app] Email Alert");
            msg.setRecipient(RecipientType.TO,
                new InternetAddress(userAccount.getEmail(),
                    userAccount.toString()));
            msg.setText("Hello " + userAccount.getName());
            Transport.send(msg);
        }
        catch(MessagingException me) {
            // manage exception
        }
        catch(UnsupportedEncodingException uee) {
            // manage exception
        }
    }
}
```

```
}
```

The `@Resource` annotation receives the JNDI name of the JavaMail™ session and injects an instance of the session in the variable `mailSession`. This variable is used within the `sendMessage` method to create a new `MimeMessage`. The content of the message is built and finally sent to the recipient by the method `Transport.send()`. The method receives as parameter an entity class representing an user registered on the application.

Using this feature, we avoid any additional implementation to add those parameters hard-coded or parameterized, saving a lot of time, simplifying the maintenance of the applications, and reusing existing resources naturally shared by the container.

C.6 Deploying the Application

After configuring the application server, it is time to deploy the application that makes use of all configured services. The development tool should be able to package a compiled version of the source code in a file *yasmim.ear*. This file is sent to the application server administrator, who proceeds with the deployment. The standard procedure is the following:

1. In the Glassfish administrative console, go to the section Applications > Enterprise Applications;
2. Click on *Deploy*;
3. Select the type *Enterprise Application (.ear)*;
4. Browse the file system for the file *yasmim.ear*; and
5. Click on *Ok*.

The application is ready to be used and its administration module is available at: `http://[server-name]:8080/yasmim`.