Ubiquitous collaborative multimedia capture of live
experiences toward authoring extensible interactive
multimedia documents

Andrey Omar Mozo Uscamayta

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Andrey Omar Mozo Uscamayta

Ubiquitous collaborative multimedia capture of live
experiences toward authoring extensible interactive
multimedia documents

Master dissertation submitted to the Instituto de
Ciências Matemáticas e de Computação – ICMC-
USP, in partial fulfillment of the requirements for the
degree of the Master Program in Computer Science
and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and
Computational Mathematics

Advisor: Profa. Dra. Maria da Graça Campos Pimentel

USP – São Carlos
June 2017

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

U84u

Uscamayta, Andrey Omar Mozo

Captura multimídia colaborativa ubíqua de
experiências ao vivo para autoria de documentos
multimídia interativos extensíveis / Andrey
Omar Mozo Uscamayta; orientadora Maria da Graça
Campos Pimentel. -- São Carlos -- SP, 2017.

129 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática Computacional)
-- Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2017.

1. Multimedia. 2. CSCW. 3. Authoring. 4. Capture.
5. Smartphone. I. Pimentel, Maria da Graça Campos,
orient. II. Título.

Andrey Omar Mozo Uscamayta

Captura multimídia colaborativa ubíqua de experiências ao vivo para autoria de documentos multimídia interativos extensíveis

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Maria da Graça Campos Pimentel

**USP – São Carlos
Junho de 2017**

ACKNOWLEDGEMENTS

I want to thank my family for the constant support at this phase of my live. My mother Maria, my grandfather Fortunato, my father Felipe, my sister Mary, my aunt Francisca and my relatives.

I want to thank Roy, Jonato and Juan, my dear friends in Peru. Thanks all my friends in Brazil. Even if you were close or far, you gave me the strength that I needed to complete this objective.

I thank this country and the ICMC-USP for receiving me and giving me the tools to complete this research. Specially, I want to thank my advisor, Prof. Maria da Graça Campos Pimentel, for the continuous support and guidance.

Finally, I thank CAPES and CNPq for their financial support.

RESUMO

USCAMAYTA, ANDREY OMAR M. **Captura multimídia colaborativa ubíqua de experiências ao vivo para autoria de documentos multimídia interativos extensíveis**. 2017. 129 p. Master dissertation (Master student Program in Computer Science and Computational Mathematics) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

A crescente importância de conteúdo multimídia gerado por usuários amadores exige pesquisas por modelos, métodos, tecnologias e sistemas que apoiem a produção multimídia. Apesar dos recentes resultados que permitem captura colaborativa em vídeo utilizando dispositivos móveis, existe uma lacuna no apoio à captura colaborativa de múltiplas mídias. O trabalho apresentado nesta dissertação propõe que a produção multimídia colaborativa ubíqua possa ser alcançada por usuários que realizem a captura de múltiplas mídias e de anotações utilizando o aplicativo móvel *CMoViA*. *CMoViA* também permite que o conteúdo gerado por esses usuários seja exportado para a plataforma *CI+WaC*, a qual permite editar e anotar documentos multimídia interativos. Essa proposta requer a extensão de trabalho recentes reportados na literatura: o modelo *I+WaC-IE* (*Interactors+WaC-Interaction Events*), a ferramenta *I+WaC-Editor* e a ferramenta *MoViA*. Assim, a aplicação *CMoViA* segue o modelo *CI+WaC-IE* proposto neste trabalho como extensão do modelo *I+WaC-IE*. A proposta foi avaliada por meio de estudo de caso realizado no domínio educacional, no qual estudantes capturam colaborativamente uma palestra.

Palavras-chave: Multimídia, CSCW, Autoria, Captura, Smartphone.

ABSTRACT

USCAMAYTA, ANDREY OMAR M. **Ubiquitous collaborative multimedia capture of live experiences toward authoring extensible interactive multimedia documents**. 2017. 129 p. Master dissertation (Master student Program in Computer Science and Computational Mathematics) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

The growing importance of multimedia content generated by ordinary users demands research for models, methods, technologies and systems that support multimedia production. Despite recent results allowing the collaborative capture of video via mobile devices, there is gap in supporting the collaborative capture of multiple media. In this dissertation we propose that ubiquitous collaborative multimedia production can be carried out by users who capture and annotate multiple media using the *CMoViA* mobile application. *CMoViA* also allows export the user-generated content to the *CI+WaC*, which allows them to edit the user-generated content in the form of interactive and extensible multimedia documents. The proposal demanded extending recent work reported in the literature, namely the *I+WaC-IE* (*Interactors+WaC-Interaction Events*) model, the *I+WaC-Editor* tool and *MoViA* tool. Hence, *CMoViA* follows the proposed *CI+WaC-IE* model. We discuss results from a case study, carried out in the educational domain, in which students collaboratively capture a lecture.

Keywords: Multimedia, CSCW, Authoring, Capture, Smartphone.

LIST OF FIGURES

Figure 1 – Capture & access multimedia production model (MARTINS; PIMENTEL, 2011)	36
Figure 2 – Multimedia annotations classification (MARTINS, 2014).	38
Figure 3 – I+WaC-Editor by Martins and Pimentel (2014): web tool for authoring and extension of multimedia documents. <i>top left</i> : media player component. <i>top right</i> : annotation visualization component. <i>bottom</i> : editing and annotation component.	50
Figure 4 – <i>MoViA</i> playback and navigation as presented by Cunha, Machado Neto and Pimentel (2013).	51
Figure 5 – Scenario of capture of an educational event by one student, with <i>post hoc</i> sharing.	52
Figure 6 – UML use case diagram that present the original main functions of <i>MoViA</i> and <i>I+WaC-Editor</i> in the context of multimedia production model.	53
Figure 7 – I+WaC-IE conceptual model by Martins (2014)	56
Figure 8 – Collaborative I+WaC-IE model extended from Martins (2014): novel elements indicated in bold font.	58
Figure 9 – Scenario of collaborative capture of an educational event by three students.	62
Figure 10 – UML use case diagram that present then main functions of <i>CMoViA</i>	63
Figure 11 – Traceability matrix of <i>CI+WaC-IE model</i> to <i>UML use case diagram</i>	66
Figure 12 – Traceability matrix of <i>CI+WaC-IE model</i> to functional and non functional requirements	67
Figure 13 – <i>CMoViA</i> architecture represented by a UML component diagram.	70
Figure 14 – <i>CMoViA</i> collaborative capture and visualization represented into an <i>UML sequence diagram</i>	72
Figure 15 – Traceability matrix of <i>CI+WaC-IE model</i> to the UML component diagram	74
Figure 16 – Traceability matrix of <i>CI+WaC-IE model</i> to the UML sequence diagram	75
Figure 17 – Conceptual data model of <i>CMoViA</i>	76
Figure 18 – ER model proposed for the <i>CMoViA API</i> application.	77
Figure 19 – Traceability matrix of <i>CI+WaC-IE model</i> over the data models of <i>CMoViA app</i> , <i>CMoViA API</i> and <i>CI+WaC</i>	83
Figure 20 – Explicit synchronization with <i>MoVIA2</i> (previous version)	88

Figure 21 – Representation of the synchronization information calculated for <i>CMoViA</i> when a new medium is captured, corresponding to Equation 6.2. . . .	89
Figure 22 – Interfaces used for media selection and application configuration in <i>CMoViA app</i>	91
Figure 23 – Interfaces used for context information in <i>CMoViA app</i>	92
Figure 24 – Collaborative capture scenerio with three smartphones where one of them uses a Bluetooth headset.	93
Figure 25 – <i>CI+WaC</i> reproducing a video and showing the different text annotations.	95
Figure 26 – <i>CI+WaC</i> reproducing a collaborative capture of 3 participants: the first captured video, the second captured photos and the third captured photos and audio using a Bluetooth headset.	96
Figure 27 – “ <i>Summer School on Computers in Education</i> ” visualization in I+WaC-Editor. View of the first mobile device used to capture the session. . .	97
Figure 28 – “ <i>Summer School on Computers in Education</i> ” visualization in I+WaC-Editor. View of the second mobile device used to capture the session: the image on the left was captured 10 seconds after the image captured by the first device shown in Figure 27.	98
Figure 29 – Likert responses average for each question (CUNHA; USCAMAYTA; PIMENTEL, 2016)	100

LIST OF SOURCE CODES

Source code 1 – Session document descriptor used for <i>CMoViA app</i>	77
Source code 2 – XML Schema of the document descriptor used for <i>CMoViA app</i> .	78
Source code 3 – <i>Session descriptor</i> description used for <i>CI+WaC</i>	79
Source code 4 – <i>Session descriptor</i> example generated with <i>CMoViA</i> , exported to <i>CI+WaC</i>	80
Source code 5 – List of available events in JSON	121
Source code 6 – JSON Event detail for existing {id}	122
Source code 7 – Detailed list of context information about the event with existing{id} in JSON	122
Source code 8 – Details of New event in JSON	123
Source code 9 – JSON object with capture and event detail for event existing {id}	123
Source code 10 – Capture object in join operation for a new participant in JSON . .	124
Source code 11 – Capture object in join operation for a new participant in JSON . .	125
Source code 12 – Form object that include a photo in binary to upload	125

LIST OF TABLES

Table 1 – Examples of interaction events captured during multimedia production.	41
Table 2 – Features in Capture & Access Applications	47
Table 3 – Example of new <i>Interactors</i> focused in the <i>author</i> concept	60
Table 4 – Examples of Interactors	127
Table 4 – Examples of Interactors	128
Table 4 – Examples of Interactors	129

LIST OF ABBREVIATIONS AND ACRONYMS

3C collaboration model	Communication, Coordination and Cooperation model
3C model	Communication, Collaboration and Coordination
ACM	Association for Computing Machinery
AndreEyA	Android EyA
API	Application Programming Interface
AVR	Automatic Video Remixing System
CI+WaC	Collaborative I+WaC
CI+WaC-IE	Collaborative Interactors+WaC-Interaction Events
CMoViA	Collaborative Mobile Video Annotation
CMoViA app	CMoViA application
CSCW	Computer-Supported Cooperative Work
CSS	Cascading Style Sheets
ER model	entity–relationship model
EyA	Enhance your Audience
EyApp	EyA application
GPS	Global Positioning System
HCI	Human–Computer Interaction
HD	High Definition
HDMI	High-Definition Multimedia Interface
HTML	HyperText Markup Language
HTML5	HyperText Markup Language version 5
HTTP	Hypertext Transfer Protocol
I+WaC-Editor	I nteractors+ WaC - E ditor
I+WaC-IE	Interactors+Watch and Comment Interaction Events
IAMmDocT	Interactor Algebra for Multimedia Document transformations
IBS	Instant Broadcasting System
ICMC	Instituto de Ciências Matemáticas e de Computação
ICTP	International Center for Theoretical Physics
IEEE	Institute of Electrical and Electronics Engineers
iOS	iPhone Operative System
JSON	JavaScript Object Notation

MIT	Massachusetts Institute of Technology
Mobicast	MOBile phone collaborative event CASTing
MoViA	M obile V ideo A notation
MoViA2	MoViA version 2
MVM	Mobile Vision Mixer
NCL	Nested Context Language
NTP	Network Time Protocol
PaaS	Platform as a Service
REST	Representational state transfer
SD	Secure Digital
SDK	Software Development Kit
ubicomp	ubiquitous computing
UGC	user-generated content
UML	Unified Modeling Language
URI	Uniform Resource Identifier
USB	Universal Serial Bus
USP	Universidade de São Paulo
WaC	Watch and Comment
WebNCL	Web presentation machine NCL
WiFi	Technology for wireless local area networking
WWW	World Wide Web
XML	Extensible Markup Language

CONTENTS

1	INTRODUCTION	23
1.1	Context	23
1.2	Research problem	25
1.3	Objective	26
1.4	Methodology and Results	27
1.5	Organization of the dissertation	29
2	BACKGROUND	31
2.1	Ubiquitous computing	31
2.2	Collaborative systems	33
2.3	User-generated content	34
2.4	Multimedia production process	35
2.5	Multimedia annotations	37
2.6	Interaction events	40
2.7	Final remarks	40
3	RELATED WORKS	43
3.1	Multimedia Capture Technology Research	43
3.1.1	<i>Educational events</i>	44
3.1.2	<i>Entertainment events</i>	45
3.1.3	<i>Video collaborative community</i>	46
3.1.4	<i>Summary</i>	46
3.2	Previous work from our group	46
3.2.1	<i>Interactors</i>	47
3.2.2	<i>I+WaC-Editor</i>	49
3.2.3	<i>MoViA</i>	50
3.2.4	<i>MoViA and I+WaC-Editor first integration</i>	52
3.3	Final remarks	53
4	CI+WAC-IE CONCEPTUAL MODEL	55
4.1	I+WaC-IE conceptual model by Martins (2014)	55
4.2	Extended Collaborative I+WaC-IE model	57
4.2.1	<i>Impact on Capture & Access</i>	58
4.2.2	<i>Impact on Interactors</i>	59

4.2.3	<i>Impact of “is a” relationship between Comment and MediaElement concepts</i>	60
4.3	Final remarks	60
5	CMOVIA AND CI+WAC-IE MODELS	61
5.1	Requirements gathering	61
5.1.1	<i>Scenario description</i>	62
5.1.2	<i>Functional and non-functional requirements</i>	64
5.1.2.1	<i>Functional requirements</i>	64
5.1.2.2	<i>Non-functional requirements</i>	65
5.1.3	<i>CI+WaC-IE model influences</i>	65
5.1.3.1	<i>Model influences over scenario</i>	66
5.1.3.2	<i>Model influences over requirements</i>	67
5.2	Software design	67
5.2.1	<i>Software development process</i>	68
5.2.1.1	<i>First cycle: MoViA2</i>	68
5.2.1.2	<i>Second cycle: CMoViA</i>	68
5.2.2	<i>Architecture</i>	69
5.2.3	<i>Prototype behavior</i>	70
5.2.4	<i>CI+WaC-IE model influences</i>	71
5.2.4.1	<i>Influences of the CI+WaC-IE model on the software architecture</i>	73
5.2.4.2	<i>Model influences over prototype behaviour</i>	75
5.3	Data model	76
5.3.1	<i>CMoViA API data model</i>	76
5.3.2	<i>CMoViA app data model</i>	77
5.3.3	<i>CI+WaC integration</i>	79
5.3.4	<i>CI+WaC-IE model influences over the data models</i>	82
5.4	Final remarks	82
6	CMOVIA IMPLEMENTATION	85
6.1	Collaborative MoViA	85
6.1.1	<i>CMoViA API</i>	85
6.1.2	<i>CMoViA app</i>	86
6.1.2.1	<i>Synchronized capture</i>	86
6.1.2.1.1	<i>Synchronization in MoViA2</i>	87
6.1.2.1.2	<i>Synchronization in CMoViA</i>	88
6.1.2.2	<i>Multimedia selection</i>	90
6.1.2.3	<i>Context information</i>	90
6.1.2.4	<i>Mobile accessories</i>	92
6.1.3	<i>CI+WaC web tool integration</i>	94

6.2	Experiments with users	95
6.2.1	<i>Previous recordings</i>	96
6.2.2	<i>Case Study</i>	98
6.2.2.1	<i>Usability Evaluation</i>	98
6.2.2.2	<i>Tests in real scenarios</i>	99
6.2.2.3	<i>Results</i>	99
6.3	Final remarks	100
7	CONCLUSION	103
7.1	Contribution and discussion	104
7.2	Limitations	105
7.3	Future Works	106
7.4	Final remark	107
	BIBLIOGRAPHY	109
	GLOSSARY	117
ANNEX A	CMOVIA API DESCRIPTION	121
ANNEX B	EXAMPLES OF INTERACTORS	127

INTRODUCTION

This chapter first presents the context in which the research reported in this dissertation is carried out. Next, the research problem and corresponding objective are discussed, followed by the methodology adopted and the research and corresponding results.

1.1 Context

Mobile and social media are changing how users create, share, and view content, as summarized by [Shenoy \(2013\)](#). Content generated by media has amateur users has been growing in the last years as a result of the usability, portability and popularization of mobile devices such as cameras, smartphones and tablets. Many people use their own mobile devices to generate documentary evidence about real live experiences in which they take part of — examples include events, concerts, trips, meetings and interviews as discussed by [Krumm, Davies and Narayanaswami \(2008\)](#). The increase in availability of mobile devices has potential to impact several applications domains — a case in point is Mobile Healthcare, as observed by researchers such as [Stankovic \(2016\)](#).

In the context of user-generated content, the growing relevance of multimedia content generated by amateur users motivates research related to models, methods, technologies and systems that support multimedia production – examples include support for the automatic summarization of multiple user-generated videos as reported by [Zhang, Zhang and Zimmermann \(2015\)](#), and for the automatic generation of photo collages as in the contribution by [Bianco and Ciocca \(2015\)](#).

As a contribution from discussions about the future of multimedia production, specialists recommend that multimedia authoring should take advantage of concepts and techniques from ubiquitous computing and remote collaboration as well as use multimedia elements in the capture of real live experiences, as summarized by [Rowe \(2013\)](#). Researchers

also recommend the support of mobile multimedia production and interactive video, as in the case of the research agenda discussed by [Juhlin *et al.* \(2014\)](#). Moreover, [Shenoy \(2013\)](#) observes that advances in multimedia research are associated with challenges that include mobile streaming, privacy, and new modes of interaction. In the context of ubiquitous computing, the multimedia production process should be transparent to users. In the context of collaborative systems, capture and enrichment of multimedia elements could be supported by collaborative tasks.

In the context of multimedia capture, even though researchers report results exploiting multiple media — such as paper and photographs as in the work by [Yeh *et al.* \(2006\)](#); paper and digital audio as presented by [Pearson, Robinson and Jones \(2015\)](#); typed text and electronic ink as proposed by [Ren, Li and Lank \(2014\)](#); audio, photos and information gathered from sensors available in mobile devices as investigated by [Kay *et al.* \(2012\)](#) — many researchers make use of *video* as the principal media element, as in the work by [Guimarães *et al.* \(2011\)](#), [Guimarães, Cesar and Bulterman \(2013\)](#), [Cunha, Uscamayta and Pimentel \(2016\)](#), [Cunha, Machado Neto and Pimentel \(2013\)](#), [Ojala *et al.* \(2014\)](#).

Considering both the state of art and the state of the practice, the main objective of collaborative video capture tools targets to the generation of a single video file that integrates the overall collaborative contents. The resulting video is generated by the computer processing of the original video elements obtained by the users participating in the capture process. The integrated video is a traditional linear video and, as such, does not contain any implicit or explicit references to the original video elements provided collaboratively by the users. The capture tools used include professional capture technology integrated in the rooms in which the capture take places as contributed by [Canessa, Fonda and Zennaro \(2007\)](#), [Viel *et al.* \(2013a\)](#). In the case of amateur users using mobile devices, the capture is usually achieved with the use of mobile applications.

In the context of capture and access applications, as surveyed by [Truong, Hayes *et al.* \(2009\)](#), the capturing of video for multimedia authoring has been investigate by many researchers, as in the work by [Brotherton and Abowd \(2004\)](#) and [Müller and Ottmann \(2000\)](#): in this this case the focus was on on user-generated content by capturing user-interactions with instrumented environments – the more transparently the better, as illustrated in the work by [Pimentel *et al.* \(2001\)](#). Focusing on mobile devices-based user-generating content in the context of capture and access applications demanding in instrumented environments, [Truong, Abowd and Brotherton \(1999\)](#) exploited the first generation of tablets (tablet-PCs) to allow the capture of images and ink obtained in an instrumented environment to be annotated synchronously by a group of students, who could enrich their own version of the captured content — this work focused on supporting the generation of personalized versions of public captured content and, then, did not

consider the sharing of contents among the students.

In the context of multimedia access, researchers from industry and academy have proposed tools for visualization, extension and enrichment of multimedia elements captured by end users, as in the contributions by [Sá, Shamma and Churchill \(2014\)](#) and [Bulterman, Cesar and Guimarães \(2013\)](#). The tools let users to perform tasks individually or collaboratively — the latter called collaborative multimedia authoring of user-generated content when resulting from the extension and enrichment of multimedia elements.

1.2 Research problem

The process of amateur multimedia production results from the demand of people that want to keep a multimedia record of their experiences. In their work, [Ojala *et al.* \(2014\)](#) classified those experiences as events of the following types: *a) sporting events* (soccer, volleyball, basketball, etc.), *b) musical events* (for example, concert, performance and recital) and *c) formal events* (such as classes and dissertation defenses). We observed that each type of event has its own features and particularities that have to be taken into consideration when capturing takes place.

Lectures, formal events in the classification by [Ojala *et al.* \(2014\)](#), is one example of live experience that can benefit from user-generated content: in this case, by students recording their live experience (during classes) using mobile devices. This, in fact, can also be cost effective as observed by [Canessa, Fonda and Zennaro \(2007\)](#). Nowadays, it is very common to see students using their own smartphones and tablets to record their classes with the permission of the professor. Generally students record in video or audio and in some cases, they take pictures of the blackboard. All these media resources are used, for example, as additional material for later review. In this case, students usually use the native applications available in their own mobile devices for recording videos, audio and images. As a result, the reviewing of the several captured media is made as separated processes — i.e., the student is forced to review each captured media individually and make the necessary connections, intrinsic to the live experience in which the media elements were captured.

Efforts have been proposed in the literature to allow one single user to record audio and images synchronously using a single mobile device, so the user can also review the corresponding media synchronously, as discussed by [Canessa *et al.* \(2014\)](#). In this case the media elements supported are audio and images: access to the captured media is individual and uses the original device employed for capturing. Explicitly exploiting ubiquitous computing concepts, the *personal audio loop* by [Hayes *et al.* \(2004\)](#) transparently captured (in fact, buffered) audio from one user to allow the recovering of recent information: this would be transparent and ephemeral user-generated content (audio only).

Sá, Shamma and Churchill (2014) have also proposed the collaborative recording of video, images or audio synchronously using mobile devices – the aim is that users who did not participate in the recording could visualize the capture content in real time by those capturing and streaming their media. In this case the media elements supported are video, audio and images, and the media reviewing could be individual or grouped on the web. The researchers proposed a high level architecture and corresponding low fidelity prototype, the latter evaluated by means of a simulation with single (non-collaborating) users.

From the representative works highlighted above, we observe that results reported in the literature support amateur user-generated content via mobile devices — achieved by users recording video or, in some cases, multiple media. In most cases, the capture activity results in a single video instead of an interactive multimedia document — an exception is the capture of audio and images which results in a corresponding (informal) multimedia document (with audio and images) without video. Other works using similar approaches are discussed in Chapter 2.

From the representative works above we can also observe a lack of support to some of the concepts associated with the recommendations by Rowe (2013) and Juhlin *et al.* (2014) as follows:

- collaboration should be supported by collaborative tasks at the time of capture and at the time of review;
- collaboration should be transparently supported by ubiquitous services;
- the user-generated content should result in a shareable and extensible interactive multimedia document.

Considering both recommendations and results reported in the literature, summarized above, in this work we aim to provide one answer to the following research question:

How to support ubiquitous collaborative multimedia production using user-generated content, captured via mobile devices, so as to result in interactive and extensible multimedia documents?

This research question guided the objective of our work, as made explicit in the next section.

1.3 Objective

The aim of this work is to propose an approach that supports the ubiquitous collaborative production of interactive and extensible multimedia documents by exploiting user-generated content captured via mobile devices.

Our approach to achieve this objective is detailed in the next section.

1.4 Methodology and Results

Previous works by our research group proposed methods, operators, and tools for collaborative authoring in themes of capture and access of multimedia content, as in the proposal of the Watch and Comment (WaC) paradigm by [Cattelan *et al.* \(2008a\)](#). The researchers proposed, among others, models and tools for the access and extension of multimedia documents in mobile platforms — for example the *MoViA* tool proposed by [Cunha, Machado Neto and Pimentel \(2013\)](#), and in web platforms — for example the *I+Wac-Editor* contributed by [Martins \(2014\)](#).

In order to support ubiquitous collaborative multimedia production by users empowered by mobile devices toward producing extensible interactive multimedia documents, we based our research on previous work from our group as follows:

- We extended the *I+Wac-IE* model by [Martins \(2014\)](#). The original work supported the modeling of user interactions with an interactive multimedia document so as to allow annotations as well as other editing operations.

Our extended model, named *CI+Wac-IE* model, allows the explicit identification of multiple authors and of their interactions with media and annotations. As a result, our extended model supports multimedia authoring carried out at the time of media capture by end users. This is possible because the extended model registers explicit collaborations carried out at the time of context-aware capture of multiple media, resulting in extensible interactive multimedia documents.

- We also extended the *MoViA* tool proposed by [Cunha, Machado Neto and Pimentel \(2013\)](#). The original mobile-based tool supported video recording locally in mobile devices and the enrichment of the video using multiple media-based annotations.

Our extended application, named *CMoViA*, allows collaborative context-aware capture of video as well as audio, images and bookmarks.

- [Martins \(2014\)](#) contributions included a web-based application called *I+WaC-Editor*, which allowed a user to manipulate multimedia documents via user interaction-based operations (*Interactors*) for the automatic and semi-automatic authoring and annotation of documents.

We integrated our *CMoViA* application with the application *I+WaC-Editor* so as to export the capture content to the *I+WaC-Editor* environment — this demanded that [Martins \(2014\)](#) extended his own tool which he then named *CI+WaC*. Hence, both multiple media content and any existing annotations captured by *CMoViA*

are made available in an environment supporting extensible interactive multimedia documents.

More specifically, the contributions of the work reported in this dissertation include:

- The extension of the *I+WaC-IE* model, named *CI+WaC-IE* model and detailed in Chapter 4, so as to include features for media element ownership demanded for support the collaborative captured events and collaborative authoring of interactive extensible multimedia documents.
- A set of new *Interactors* associated with the author of the media in the time of capture (Section 4.2.2).
- A specification for a collaborative multimedia document (Section 5.3.2) that supports collaborative multimedia capture based on *CI+WaC-IE* model.
- Definition of an explicit physical synchronization method using a mobile device's accelerometer (Section 6.1.2.1).
- A method of collaborative capture that includes the utilization of mobile accessories for a more ubiquitous user interaction (Section 6.1.2.4).
- A prototype, called *CMovia app*, based on the extension of *MoViA app* that let users to perform collaborative captures using different multimedia elements and various visualization approached for resulting multimedia documents (Section 6.1.2).
- A REST API prototype, *CMoViA API*, that supports both the collaborative capture with synchronized media from all clients applications, and the storage of context information about the capture (Section 6.1.1).
- A process to export the user-generated multimedia to the *I+WaC-Editor* (Section 6.1.3), with corresponding demand for extension of the original editor by his designer Martins (2014) himself. The process generates a new collaborative multimedia document that can be visualized in the web-based editor named, after the extension, *I+WaC-Editor*.

As a result of our research and development work described previously, we can answer the research question proposed earlier as follows:

Ubiquitous collaborative multimedia production can be carried out by users who capture and annotate multiple media using the CMOVIA mobile application and export the user-generated content to the CI+WaC, which allows them to edit the user-generated content in the form of interactive and extensible multimedia documents.

In the next section, we introduce the remaining chapters of this dissertation.

1.5 Organization of the dissertation

This dissertation is organized as follows. **Chapter 2** overviews the theoretical concepts used in the dissertation, which cover ubiquitous computing, collaborative systems, user-generated content, multimedia production process, multimedia annotations and interaction events.

Chapter 3 reviews related works and a summary of results contributed by our research group, which we use in our own research.

After summarizing the *I+WaC-IE* model proposed by [Martins \(2014\)](#), in **Chapter 4** we present our extension for the model, *CI+WaC-IE* model, and discuss its impact on collaborative capture scenarios.

In the light of the extended *CI+WaC-IE* model we propose, in **Chapter 5** we detail the design of the proof of concept *CMoViA* prototype we developed, which in turn was based on the *MoViA* application previously contributed by [Cunha, Machado Neto and Pimentel \(2013\)](#).

In **Chapter 6** we detail implementation features of the prototype *CMoViA*, and discuss results from a case study carried out in the educational domain.

In **Chapter 7** we summarize the contributions and limitations of the work reported in this dissertation, and briefly discuss future work.

BACKGROUND

This chapter introduces the main concepts related with the research presented in this dissertation: ubiquitous computing (Section 2.1), collaborative systems (Section 2.2), user generated content (Section 2.3), multimedia production process (Section 2.4), multimedia annotations (Section 2.5), and interaction events (Section 2.6).

2.1 Ubiquitous computing

Weiser (1991) defines ubiquitous computing (ubiquomp) as units of hardware and software connected by wireless technology. They communicate with each other with the objective of performing different tasks. It should be transparent for users: the more transparent, the more ubiquitous. The features that should be present in ubiquitous technologies are: 1) *cheap*, 2) *low-power hardware*, 3) *software for ubiquitous applications*, and 4) *a network* that connects all devices involved.

Application-driven research in ubiquitous computing has focus in three particular features: 1) *natural interfaces*, 2) *context-aware application*, and 3) *automated capture and access* — according to the classification proposed by Abowd and Mynatt (2000). These authors discuss their classification as follows:

1. **Natural interfaces.** For many years, users just have used peripheral devices¹ as interfaces to interact with desktop computers, laptops and even mobile devices. Ubiquitous computing claims that computer interfaces need to become more human natural. Because of that, computer interfaces were supported by other communication forms — for example handwriting recognition, speech recognition and gesture language. The mobile technology improved the interaction interface by the utilization of a multi-touch gesture language. The mobile device detect the multi-touch events over

¹ Peripheral devices such as mouses, keyboards, microphones and speaker.

the screen and use it to interact with the applications. However, the communication interfaces are still not natural enough to be completely transparent for users, this feature brings many challenges for academy and industry to improve them more every time.

There has been a continuous increase in research aiming at providing interfaces more natural to humans including speech recognition and movement detection. Specialized hardware such as Kinect and mobile devices (with sensors such as Global Positioning System (GPS), accelerometer, gyroscope and magnetometer) allow researches and developers to investigate alternative interfaces for communication. The main limitations in the use of this technology are power capacity, processing power and sensor precision.

2. **Context-aware applications.** It is important that applications obtain contextual information about the different types of interaction including user-user interaction, user-application interaction and user-device interaction. The design of applications can focus on answering the following five *W*'s of context-aware information:

- *Who*: Information about the people involved during all the duration of an interaction event;
- *What*: Interpretation about the human activities performed all along an interaction event;
- *Where*: Information about the physical location and the history of movements about the users all along an interaction event;
- *When*: Information about the time moments in which users engage in some interaction may help the understanding of the human activity;
- *Why*: Information about the user — for example, collect from sensors informing body temperature and heart rate — could help understanding “*why*” the user is doing a particular action in a specific time.

Most mobile devices such as smartphones, cameras and game consoles have different kind of sensors, including GPS, accelerometer, gyroscope and magnetometer. The use of data collected from sensors could help researchers and developers to gather context-related information. Applications then can store, analyze and use that information towards trying to understand or to infer human activity.

3. **Automated capture and access.** The importance of applications that capture daily events to make information available afterwards is related to the feature of “*easy to use*”. Activities to be captured can be of different types, and can be grouped into personal events or group events — for example vacations, birthday, parties, educational events, sport events and music events. Each type of event has its own

set of particular features. Because of that, the applications must provide access to the media elements recorded during the event and also try to capture context information according to the set of particularities of the event. Applications then should provide alternatives for visualization, editing or enrichment of the recordings for the users.

As a result of our research towards the objective of our work, we produced a prototype with features associated with the three classes discussed by [Abowd and Mynatt \(2000\)](#). We propose a natural interface by a re-engineering process oriented to the end-users attending regular sessions – for instance, students attending lectures in classrooms. The prototype was designed to collect and share context information between participants: in this case, information about the recordings and the user interaction. Moreover, the prototype is a solution that automates the capture and access tasks associated with the multimedia production process in the context of collaborative capture. Detailed in the Section [6.1.2](#).

2.2 Collaborative systems

In his review of Computer-Supported Cooperative Work (CSCW), [Grudin \(1994\)](#) reported that this area emerged in the 80's from the need researchers and practitioners had to learn how people work in small groups and organization, as well as how the computer-based technology supports work-group. At the beginning, researchers focused on supporting the isolated work of every person in a company, although some people could work in the same team following a common objective. The main challenge for small groups was how to improve the communication between members because they usually shared key goals. Considering that *CSCW* applications should anticipate possible frictions between members, researchers designed tools to maintain members communicating and aware of their objectives. One level above, the challenge of organization leaders was to improve the coordination between different groups, considering that each group could have different key goals. This led researchers to design *CSCW* applications aimed at dealing with conflicting goals by improving the coordination between the different groups.

Research in *CSCW* involves researchers from different disciplinary perspectives that discuss aspects of computer-based technology design for work group and the ways how people used it. [Ackerman \(2000\)](#) and [Schmidt \(1992\)](#) observe that collaborative system applications could have many limitations for the social-technical gap.² They recommend following recommendations from Human-Computer Interaction (HCI) research: designers should not force users to adapt to a specific technology, instead the technology should adapt to the ways groups work.

² Social-technical gap: mismatch between what is required socially and what developers can do technically.

In the context of multimedia production, collaborative work can support multimedia production in capture and extension stages. These stages could produce media elements that compose or enrich media documents in a collaborative way.

The literature reports collaborative approaches to author multimedia content which ranges from support to the generation of personalized versions of public captured content, as in the work by [Truong, Abowd and Brotherton \(1999\)](#), to integrating multiple information captured from colocated or distributed meetings, as in the proposals by [Streitz et al. \(1994\)](#).

In early research on computer-supported collaborative work, [Ellis, Gibbs and Rein \(1991\)](#) advocated “3C” areas which must receive attention from researchers and developers of software systems to support user-user interaction: Communication, Collaboration and Coordination (3C model). Inspired in their work, [Gerosa et al. \(2006\)](#) proposed a related Communication, Coordination and Cooperation model (3C collaboration model) for the development of collaborative systems.

The work we report in this dissertation aims at supporting collaborative capture and then demands attention to collaboration/cooperation, communication and coordination. As an example, students attending a class could *a) collaborate* so as to obtain a multimedia document as material for reviewing the class. Regarding *b) communication*: in the example, each student could share implicitly (*implicitly context information refers to a context-aware information captured automatically with non user interaction like for example the time when a recording started that were stored automatically in the video like complementary information.*) or explicitly (*explicitly context information refers to a context-aware information captured by a user interaction like for example when the user that is recording, makes a bookmark of an important moment of the event and use a tool or functionality in the capture application to register that moment.*) context information about his recording with the other students also capturing some aspect of the class — this could be unidirectional communication). Finally, in the aspect of *c) coordination*: the recording decisions such as position in the classroom and choosing which media to record are taken by students considering the features of their mobile devices and the (context-aware) information of the other recordings. Then, a group of users could collaborate by coordinating their efforts while communicating transparently or explicitly.

2.3 User-generated content

In the first years of the World Wide Web (WWW), users accessed content published on dedicated *Web servers* by authorized authors — the servers were managed by *Web masters* who were fluent in HyperText Markup Language (HTML). By the mid 90s, Ward

Cunningham launched the *Wiki Wiki Web*³ which soon was known simply as *Wiki*. Ward's aim was to provide “*It's a web of people, projects and patterns accessed through a cgi-bin script. It has a forms-based authoring capability that doesn't require familiarity with html*”, as he⁴ put it. Therefore, Ward's aim was to allow content to be created by users themselves. This approach caused a change in paradigm in the WWW which soon started to be called *Web 2.0*: content could be created by end-users first in the Wiki, and later in other platforms including the many types of Social Networks that abound nowadays.

By allowing end users to author content, the *Web 2.0* let users interact not only with the Web but also among themselves. Tools like wikis, blogs and forums allow users to generate a massive amount of information on the Web. In particular, the popularization of social networks and mobile technology allowed user-generated content (UGC) to include media elements such as audio, video, images and GPS locations, to name a few as illustrated in the contributions by Kuksenok, Brooks and Mankoff (2013) and Krumm, Davies and Narayanaswami (2008).

Rowe (2013) observes that the increase in multimedia *user-generated content* motivates research in models, methods, technology, systems that support the multimedia production of new content. Amateur collaborative capture proposed, supported by the work reported in this dissertation, fits in the *user-generated content* paradigm since it aims at supporting a group of users (e.g. students) to collaborate, coordinate and communicate so as to generate multimedia content in the form of video, audio, photos and bookmarks associated to live experiences in which they participate (e.g. classes) – the resulting content being shared in a Web repository in which the resulting content is made available as an interactive and extensible multimedia document.

2.4 Multimedia production process

The process of multimedia production used in professional video productions, like television and cinema productions, is comprised of four main stages: *pre-production* (planning), *production* (recording media), *post-production* (prepare the media for consumption) and *distribution* (final product publishing), as observed by several authors including Minneman *et al.* (1995), Nack (2005) and Kindem (2009). Krumm, Davies and Narayanaswami (2008) observed that the popularization of smartphones and tablets led end-users to become producers of amateur multimedia content. The process of amateur multimedia production adapts the professional process to deal with the varied limitations in each of the four original stages.

Nack (2005), Hardman *et al.* (2008), Cesar *et al.* (2008) propose different cus-

³ <<http://wiki.c2.com/?WikiHistory>> and <<https://en.wikipedia.org/wiki/WikiWikiWeb>>

⁴ <<http://c2.com/wiki/mail-history.txt>>

Figure 1 – Capture & access multimedia production model (MARTINS; PIMENTEL, 2011)

tomizations and adaptations to the traditional multimedia production process considering the professional and amateur multimedia production. They have extended the traditional process with the inclusion of enrichment and editing operations over the final product. As result, the enriched multimedia document becomes the new final product.

Martins and Pimentel (2011) proposed a six stage process named *capture & access multimedia production process*: this process takes into account that users pre-produce, produce, access, review, edit and enrich multimedia elements. They also defined user roles (amateur and professional users) and tasks for each stage of the process, as illustrated in Figure 1. The six stages of the *capture & access multimedia production process* are:

- **Pre-production.** Planning stage of *what* (media to capture), *how* (capture resources and process), *where* (capture locations), *when* (capture time and duration) and *who* (responsible people). Decision stage about themes, goals, events, actions and so on, required in capture stage. Also, it is a stage for selection and preparation of equipment, people, etc. In the case of collaborative capture, this stage is used by the participants to perform the coordination between them.
- **Capture.** Stage of live recording in which decisions made at *pre-production* stage are followed. This stage is in charge of taking care of synchronization of the media

streams involved in the recording process. Also, it is possible to capture other useful information about the event, for example: GPS coordinates, camera person information and sensors information.

- **Post-production.** Stage in charge of preparing the media captured to be consumed by users following the feature of *easily to consume*. In this stage the media elements can be analyzed and enriched by an authoring process (manual or automatic). The media elements, enriched in this stage, generate a multimedia presentation as final product.
- **Publishing.** Stage in which the media assets prepared in *post-production* stage, or enriched by end-users in the *extension* stage, are considered as final product. They are published for end-user consumption in distribution channels such Web servers, Web repositories, broadcasts channels and so on.
- **Access.** Stage in which multimedia content is consumed by end-users. End-users perform tasks such as visualization, review and navigation over the content. This stage can be used for audience analysis or for derivation of implicit or explicit ratings.
- **Extension.** Stage in which all media and metadata created or generated along the all stages of the process can be affected by authoring decisions by amateur or professional users. Users can enrich some content to generate new versions. These enriched version can becomes novel final products.

The amateur collaborative capture, proposed in this work, fits in the context of multimedia production process. As we saw before, capture & access multimedia production model were extended to support amateur and professional production, so we grouped together all the available tasks for amateur collaborative capture and visualization into the six stages of the multimedia production process, having in mind that this work focuses mainly in pre-production, capture and publishing stages.

2.5 Multimedia annotations

Early experiences regarding interactive multimedia authoring by capture information in instrumented environments reported efforts in the educational domain, as in the contributions by [Abowd et al. \(1999\)](#), [Brotherton and Abowd \(2004\)](#), as well as in the CSCW domain, as in the contributions by [Pedersen et al. \(1993\)](#) and [Streitz et al. \(1994\)](#) as well as in other meeting systems as surveyed [Yu and Nakamura \(2010\)](#). These contributions support multimedia authoring by annotating the user interaction with devices embedded the environment, or “linking by interacting”, as discussed by [Pimentel, Abowd and Ishiguro \(2000\)](#).

Figure 2 – Multimedia annotations classification (MARTINS, 2014).

Exploiting early tablet PCs, Goularte *et al.* (2004) proposed multimodal annotations on video with the M4Note system. Video annotations became a very popular feature in many commercial and research applications. On a daily basis different video annotations are available as subtitles in movies or annotations over YouTube⁵ videos. Different types of available annotations include text annotations (subtitles, comments, etc.), ink annotations (YouTube annotations), audio annotations, image annotations, video annotations, time annotations and so on. One of the aims of providing annotations on video is to allow third-party annotations, as in the work by Guimarães, Cesar and Bulterman (2010), Guimarães, Cesar and Bulterman (2012) and Teixeira *et al.* (2012).

Multimedia annotations, or annotations over digital content, became an important research topic, offering varied tool that support and facilitate the annotation making process. Martins (2014) defines multimedia annotations as “*information units about a media element⁶ or a document media⁷ generated along a multimedia production process.*”⁸ Figure 2 shows the classification by Martins (2014) of multimedia annotations in seven dimensions as described next:

1. **Function.** Annotations can be classified according to their function as: a) *Metadata annotations* when they have a descriptive role with the associated data; and b) *Con-*

⁵ Different ink annotations supported in Youtube <<https://www.youtube.com/watch?v=g2P-jxDhjCo>>

⁶ “*Media element*” like audio, video, image, text, ink and so forth.

⁷ *Document media* is a document compose for a set of synchronized media elements.

⁸ The multimedia production process used is the *Capture & access multimedia production model* proposed by Martins and Pimentel (2011)

tent annotations when they have a function of extension, clarification, comprehension, relation, etc.

2. **Form.** When annotations follow a predefined scheme, they are classified as *a) Structured*, otherwise they are *b) unstructured*. Considering their content, annotations can be classified as one *c) simple object* or as one *d) complex object*: the latter applied to annotations composed of several simple objects.
3. **Capture.** Annotations can be classified according to in which stage of the multimedia production process they were originated: *a) manual annotations* or explicit annotations, when they are created explicitly by user; *b) automatic annotations* or implicit annotations, when they are produced automatically by a capture device as in the case time or geolocation annotation; and *c) derived annotations*, when they are produced as a result of content analysis of the associated media or of other annotations.
4. **Abstraction.** Depending of their abstraction, annotations can be classified as: *a) Content-based annotations*, when they are product of some analysis of the associated media, as in the case of bookmarks related to the beginning of user speech identified by audio processing tools; and *b) Context-based annotations*, when they enrich the associated media with metadata collected by sensors such as GPS or accelerometers.
5. **Granularity.** Annotations can be classified as: *a) Global annotation*, when associated with a media element as a whole; *b) Fragment annotation*, when associated with an individual fragment of a media element; *c) Special fragment annotation*, when associated with a visual region of the media element (image or video); and *d) Temporal fragment annotation*, when associated with a time interval of the media element (video or audio).
6. **Coupling.** Annotations can be classified as *a) embedded annotation* (tightly coupled), when it can be aggregated in the same container of the corresponding media element; and *b) decoupled annotation* (loosely coupled), when it is aggregated in an external document or repository.
7. **Origin.** Annotations can be classified according to the stage in the multimedia production process in which they were created: *a) pre-production*, *b) capture*, *c) post-production*, *d) publishing*, *e) access*, *f) extension annotation*.

In our work we extended the *MoViA* tool originally proposed by Cunha, Machado Neto and Pimentel (2013), as detailed in Section 3.2.3. The original tool recorded video and let users to add different *multimedia annotations* over the video: audio, digital ink and text. In the work reported in this dissertation, we extended the capture feature of *MoViA* allowing

the collaborative capture of audio, photos, and bookmarks, as detailed in Section 6.1.2. Moreover, the process of authoring annotations was extended so as to allow the collaborative association of annotations to a session in a specific time, instead of with a specific media.

2.6 Interaction events

Martins, Vega-Oliveros and Pimentel (2011) introduce interaction event as *metadata produced in the different stages of the multimedia production process by the interaction between users, devices, media or the live session*. The interaction events are divided in three groups, *a) user-user interaction*, e.g. message exchange between users that are recording a live lecture; and *b) user-device interaction*, e.g. a user activating the mute option for the microphone while he is recording a soccer game. *c) user-media interaction*, e.g. an interviewer creating a bookmark in the moment when an interviewee started answering each of the questions in an interview.

According to Geyer, Richter and Abowd (2005), *interaction events* also can be categorized according to their time-stamp or their side-effect. The **time-stamped classification** can be *a) online*, when the interaction event occurs during the live activity (at the time of capture); and *b) off-line*, when the interaction event occurs after the capture activity has been concluded. The **side-effect classification** includes *a) explicit*, when the interaction event is created by the direct user interaction; and *b) derived*, when the interaction event is created by computer process of a specific media or the media document. For example, the message exchange between users during the capture (user-user interaction) is an explicit on-line interaction, and many bookmarks calculated by the automatic identification of silent moments in an interview video (user-media interaction) is a derived off-line interaction.

Table 1 presents examples of interaction events examples considering the different classifications, stage of the interaction according to the *Capture&Access multimedia production process*, location in which the interaction event is stored.

2.7 Final remarks

In this chapter we discussed concepts and results associated with the research reported in remaining of this dissertation. In the next chapter we review related work reported in the literature, including results from our research group which were used in the development of the research detailed in the remaining chapters.

Table 1 – Examples of interaction events captured during multimedia production.

Event	Classification	Stage	Origin	Description
strokeDrawn	explicit, off-line, user-device	extension	whiteboard log	a ink stroke has been drawn in a whiteboard
slideChanged	derived, off-line	post- production	whiteboard log	a switch of slides oc- curred in a whiteboard
messageSent	explicit, on-line, user-user	capture	chat log	a text message has been transmitted to other user
commentCreated	explicit, on-line, user-media	extension	annotation log	a comment over a slide or a media element has been created
displayPaused	explicit, off-line, user-device	access	player log	playback suspended
micMuted	explicit, on-line, user-device	capture	capture log	a microphone has been explicitly muted
silenceBegin	derived, off-line	post- production	audio analy- sis	a period of no sound has started

Event column represents the name of the interaction event, **Classification** column follows the classification of the interaction event described below, **Stage** column represents the action stage of the interaction event from *Capture&Access multimedia production process*, **Origin** column represents the location where the interaction event is stored and **Description** column shows a clarification of the interaction event.

RELATED WORKS

To carry out research towards solving a particular problem, it is necessary to learn from contributions reported by other authors working on the same or similar problem. In this chapter we first discuss related works (Section 3.1) and in Section 3.2 we present an introduction to tools, contributed by our research group, that we use in work we report in the remaining chapters.

3.1 Multimedia Capture Technology Research

We analyzed applications and technologies used in the process of multimedia production, in collaborative systems, in ubiquitous computing and in user-generated content as follows:

- The selected contributions reported in three digital libraries: ACM DL,¹ IEEE Xplore,² and Springer Link;³
- We defined a search string with the terms: multimedia capture, video capture, collaborative capture, capture technology, multimedia synchronization;
- We applied the search string to the abstract and title of the publications in the digital libraries;
- We obtained 134 articles as a result from the search;
- After reading the abstract of each article, we selected 35 publications as the most relevant for our work;
- We also considered works that report available commercial applications.

Different kinds of technologies for capture and access process were developed with the objectives that include reduction in cost and allowing good recordings by amateurs. Most selected publications focuses on the capture of educational and entertainment events.

¹ <<http://dl.acm.org/>>

² <<http://ieeexplore.ieee.org/Xplore/>>

³ <<http://link.springer.com/>>

3.1.1 Educational events

Researchers report alternatives for supporting the capture of live events such as work meetings, as in the seminal work by on ubiquitous computing [Weiser \(1991\)](#), which was followed by others such as [Pedersen et al. \(1993\)](#). In the educational context, the work by [Abowd et al. \(1996\)](#) was soon followed by others including [Bianchi \(1998\)](#), [Mukhopadhyay and Smith \(1999\)](#), [Müller and Ottmann \(2000\)](#), and [Bianchi \(2004\)](#). Nowadays many companies provide services for the automatic capture of lectures for later review, as employed for instance in the Massachusetts Institute of Technology (MIT), which has been using the services by [1beyond.com \(2015\)](#). Generic infrastructures for supporting the capture of experiences have also been proposed, as in the work by [Truong and Abowd \(2004\)](#) and [Pimentel, Baldochi Jr. and Cattelan \(2007\)](#). [Truong, Hayes et al. \(2009\)](#) present a survey on the use of ubiquitous computing for capture and access.

[Canessa, Fonda and Zennaro \(2007\)](#) proposed Enhance your Audience (EyA) for synchronized capture and web publication of the classes offered by the International Center for Theoretical Physics (ICTP). The *EyA* technology was developed for synchronized class capture scenario with low quality video, periodical High Definition (HD) photos and audio. They designed a box called *EyA Box*, located in the classroom. The *EyA Box* contains an Apple Mac mini, an USB camera, a webcam and an USB microphone. The ICTP has been using the EyA technology for recording and publishing their classes since 2007.⁴

The Android EyA (AndreEyA) tool, for the Android platform, and EyA application (EyApp) tool, for the (iPhone Operative System (iOS) platform, were proposed as extensions of EyA technology. They were focus supporting students recording their classes and use the capture content as reviewing material. The mobile tools let students capture periodical photos and audio, also the reproduction of the recordings as described by [Canessa et al. \(2014\)](#).

[Viel et al. \(2013b\)](#) propose a technology for multi-video lecture capture. It was composed of a instrumented classroom with cameras, electronic whiteboards and computers. The prototype was composed of three modules: *a) ClassRec*, for synchronized video capture transfer to a server by a message broker. *b) ClassGen*, that generate an interactive multi video with the videos captured with ClassRec. *c) player*, generate a multi video objects for HyperText Markup Language version 5 (HTML5) player or a Nested Context Language (NCL)⁵ interactive video for Web presentation machine NCL (WebNCL)⁶ player.

⁴ <http://sdu.ictp.it/index.html>

⁵ <http://www.ncl.org.br/>

⁶ <https://github.com/lince/webncl>

3.1.2 Entertainment events

Engström *et al.* (2012) introduced the Mobile Vision Mixer (MVM) System as a mobile tool for a group of amateur users co-produce and broadcast in real-time. As it was a professional recording, one member of the team has the role of director whose application shows the other four cameras recordings at the same time so the director decides which camera is active at broadcasting at any time. The system was composed of mobile cameras, Bambuser (a live streaming online service), a local MVM server and the Mobile mixer application. Later, Engström, Perry and Juhlin (2012) proposed the Instant Broadcasting System (IBS) as a mobile application (Movino) for video capture and a server application for mixer and broadcasting in real live. Similarly to their previous work, one member had the role of director.

Kaheel *et al.* (2009) proposed MOBILE phone collaborative event CASTing (Mobicast) as a technology for the collaborative capture of an event, integrating the individual videos together so as to broadcasting the result. The system is composed of a mobile application for both video capture and transfer the video in real time to a director, which in turn is a cloud service that generates the integrated video. To synchronize the different devices, the application uses a centralized server which employs the Network Time Protocol (NTP)⁷ to maintain synchronized the clocks from the server with the devices used for capture.

Sá, Shamma and Churchill (2014) designed the *Caleido* tool as a non functional iOS prototype — low fidelity prototype that help them to simulate a collaborative scenario. Their application focuses on video, audio and photo collaborative capture in real live scenarios. The design included the possibility of users changing the media being recorded where depending on their own capture and on the other users.

There are several commercial mobile tools for collaborative video capture. On example is *Streamweaver*,⁸ which let camera men, up to four points of view from an event, show their recording in a single screen divided in four. The system also provides visualization of the video generated for the iOS mobile and web platforms. Using a time-fixed scheduling approach, *Vyclone*⁹ creates a single video by alternating the points of view of each participant in a collaborative capture. For example, the tools plays the video from each participant's point of view alternating at each ten seconds, being the visualization possible for the web as well as for the Android, iOS and Windows Phone platforms.

⁷ The Network Time Protocol (NTP) is used to synchronize the time of a computer client or server to another server or reference time source, such as a radio or satellite receiver or modem.

⁸ <<http://streamweaver.com/>>

⁹ <<http://vyclone.com/>>

3.1.3 Video collaborative community

Nowadays, people already have used their own mobile devices to record partially or integrally events like concerts, shows, sport events and so on. Exploiting such a context, many authors proposed use these existing video to generate a video result called video mix or *mashup* – the proposals include using videos freely available in collaborative communities in which users can transfer their videos from a particular event.

Guimarães *et al.* (2011) and Bulterman, Cesar and Guimarães (2013) proposed *MyVideos* as a hybrid authoring tool that integrates and synchronizes, using audio fingerprints, videos from an event provided by a community. A web application allow users to make annotations towards identifying participants in the video and, as result, plays a video mixed focused on a specific point of view. The application was experimented in the context of relatives of a music player or singer participating in a school concert. Using videos provided by the collaborating community, *MyVideos* plays a video mixed with the better points of view focus on a given student.

Kennedy and Naaman (2009) proposed an approach for the organization and synchronization of a video collection of a concert captured by a collaborative community. Their prototype takes all videos and synchronizes them using audio fingerprints so as to reproduce all the videos at the same time.

Ojala *et al.* (2014) reported the Automatic Video Remixing System (AVR) as a system to support a collaborative community. At the end of a concert, each user transfers his personal video to the system which, then, generates a video remix with the best points of view from each contributed video. The resulting video is given back to the user as a reward.

3.1.4 Summary

Table 2 summarizes the features provided by different platforms employing the *capture & access* paradigm. Several of these features are important in a solution tackling the problem we focus in the work we report in this dissertation. We also based our research on previous work from our group which we summarize in the next section.

3.2 Previous work from our group

In order to support ubiquitous collaborative multimedia production by users empowered by mobile devices toward producing extensible interactive multimedia documents, we based our research on previous work from our group. The *Interactors model* (Section 3.2.1) was the base for the web-based multimedia authoring tool I+WaC-Editor (Section 3.2.2), MoViA (Section 3.2.3) was proposed as a recording and authoring mobile tool.

Table 2 – Features in Capture & Access Applications

Reference	Application	Devices		Media				Capture		Access	
		W	M	V	A	I	B	C	S	Vi	An
Canessa, Fonda and Zennaro (2007)	EyA	✓		✓	✓	✓			✓	✓	
Kaheel et al. (2009)	Mobicast	✓	✓	✓				✓	✓	✓	
Kennedy and Naaman (2009)	prototype			✓					✓	✓	
Guimarães et al. (2011)	MyVideos	✓		✓					✓	✓	✓
Engström et al. (2012)	MVM	✓	✓	✓				✓		✓	
Engström, Perry and Juhlin (2012)	IBS	✓	✓	✓				✓		✓	
http://vyclone.com/	vyclone	✓	✓	✓				✓		✓	
Viel et al. (2013b)	prototype	✓		✓					✓	✓	✓
http://streamweaver.com/	streamweaver	✓	✓	✓				✓		✓	
Canessa et al. (2014)	AndrEyA		✓		✓	✓			✓	✓	
Sá, Shamma and Churchill (2014)	Caleido		✓	✓	✓	✓		✓	✓	✓	
Ojala et al. (2014)	AVR	✓		✓					✓		
Cunha (2014)	MoViA		✓	✓			✓			✓	✓
Martins (2014)	I+WaC-Editor	✓		✓						✓	✓

Column **Devices** shows devices (**W**: web applications, **M**: mobile applications) used for the applications in the stages of capture and access. Column **Media** shows possible media elements used in capture (**V**: video, **A**: audio, **I**: images or photos and **B**: bookmarks). Column **Capture** shows if the application allowed collaboration (**C**) and synchronization (**S**) in the capture stage. Column **Access** shows if the application provides functionality for visualization (**Vi**) and the possibility for users to make annotations (**An**).

3.2.1 Interactors

In early work [Pimentel et al. \(2005\)](#) and [Cattelan et al. \(2008b\)](#), respectively, proposed and formalized operators capturing user interactions with pen and touch-based devices to generate corresponding dynamic multimedia documents. Generalizing that work, [Vega-Oliveros, Martins and Pimentel \(2011\)](#) defined *Interactors* as “operators based on the interaction of a user with some media” and proposed “a model of operators to generate interactive multimedia documents from captured media.” These multimedia operators support visualization, enrichment and formalization of multimedia documents oriented to amateur users.

The *Interactors* can be also divided in two groups. *Group 1* is composed of editing operators, which can be used for expansion (to generate more media elements) or filtering (media elements elimination). These operators focus on supporting manual authoring. *Group 2* is composed of operators for enrichment, visualization and browsing operations.

These operations are the focus in manual or automatic authoring, as formalized by [Martins \(2014\)](#)

[Martins, Vega-Oliveros and Pimentel \(2011\)](#) and [Martins \(2014\)](#) propose the *Interactors model* including concepts such as:

1. **Multimedia document** or *multimedia session* that aggregates all media captured in a single presentation.
2. **Interaction event** defined in Section 2.6.
3. **Multimedia session** is a “*non-empty set of media streams, synchronized and co-related, over which at least one kind of interaction event can be associated*”.
4. **Interactor** is “*an operator that is applicable to a specific type of medium and consists in a set with at least one interaction event.*” The taxonomy defines four classes of interactions related to user-media interaction:
 - *time*, timestamped to the start of a session captured.
 - *attributes*, media features collected in the capture process like color, thickness in ink strokes, etc.
 - *action*, possible user actions like drawing, erasing, muting, etc.
 - *position*, boundary limits of the interaction event over a surface, for example Cartesian coordinates of an ink stroke.

The *Interactors* were grouped in function at the media element that is used in the interaction event, as follows:

- **Inkteractors**

Interactors obtained by the processing features of user-ink interactions (e.g. a user design a stroke in his table). The main features are grouped in: *a) time-based*, used to filter or expand the media elements based in time constraints; *b) attribute-based*, use attributes of pen strokes like color, thickness and so on; *c) action-based*, user interaction with capture system like drawing, erasing and so forth; and *d) position-based*, user-ink interaction represented as a set of points in Cartesian coordinates.

- **AudioInteractors**

Interactors obtained by content-based analysis of user speech. The main features are grouped in: *a) time-based*, obtained by detecting patterns on digital audio file such as silence moments, spoken moments, etc; *b) attribute-based*, digital audio attributes (e.g. noise, frequency, amplitude) used to detect interest points; and *c) action-based*, capture moments associated with actions such as when the user activates the mute function of the microphone.

- **TextInteractors**

Interactors obtained by textual user-user interaction. The main features are grouped in: *a) time-based*, obtained by detecting time interval between message exchange; and *b) attribute-based*, text message attributes like font type and color.

- **BoardInteractors**

In scenarios when users employed board-like surfaces such as tablets or electronic whiteboards to deliver a presentation, the interactors are associated at these boards. The main features are grouped in: *a) time-based*, obtained by monitoring user-image interaction in capture stage; and *b) attribute-based*, slides attributes used to detect images of interest (e.g. a particular slide).

- **VideoInteractors**

Interactors obtained by identifying user interactions with streams of visual content like slides, photos or video. The main features are grouped in: *a) time-based*, obtained by monitoring user-image interaction for example moments when a point in the video was watched; and *b) attribute-based*, which use image attributes like type, size and so on, to detect images of interest.

Examples of each type of *Interactors* are given in Annex B.

3.2.2 I+WaC-Editor

The **Interactors+WaC-Editor** (I+WaC-Editor) is a web-based multimedia authoring tool that focuses on providing support for the extension of multimedia documents: one of the stages of the *capture & access multimedia production process* illustrated in Figure 1 on page 36.

Martins (2014) proposed the web tool as a proof-of-concept of concepts proposed in his doctoral thesis. A main aspect of Martins (2014) original work is the support to anonymous authoring and annotation and, as a result, anonymous implicit collaboration. In the work reported in this dissertation, the original model by Martins (2014) was extended to allow enriching the authoring process with the explicit identification of the authors in both individual and collaborative multimedia authoring. This extension is detailed in Chapter 4.

The *I+WaC-Editor* integrated the *Interactor operators* detailed by Martins and Pimentel (2014) and the features of the authoring tool *WaCTool* proposed by Cattelan et al. (2008a). The *I+WaC-Editor* was conceived as a playback and enrichment tool. Figure 3 show the tool composed of a media player component, an annotation visualization component and an editing and annotation component. *I+WaC-Editor* was implemented with native web technology (HTML5, CSS and Javascript).

Figure 3 – I+WaC-Editor by [Martins and Pimentel \(2014\)](#): web tool for authoring and extension of multimedia documents. *top left*: media player component. *top right*: annotation visualization component. *bottom*: editing and annotation component.

Figure 3 shows a screen of the *I+WaC-Editor* video visualizer. It shows the main video and other related media. The bottom part of figure shows several timelines with comments. A comment is a text annotation associated with an time interval over the main video. At the time of playback, the presentation of the comments follows behaviors as follows: *a) default*, the comment will be shown during the corresponding interval; *b) pause*, the player will stop at the beginning time of this comment; *c) loop*, at the end of the interval, the player will restart at the start of the interval; and *d) skip*, the player will skip until the ending time of the interval. These behaviors are defined in the *I+WaC-IE model* by [Martins \(2014\)](#), discussed in Section 4.1 on page 55.

3.2.3 MoViA

The **Mobile Video Annotation (MoViA)** tool is an Android application aimed at supporting the collaborative annotation of videos. MoViA users can record, playback, navigate and enrich video with collaborative annotations and share the annotations with other users. Although MoViA was conceived to allow the capture of any kind of event, the case studies presented by [Cunha \(2014\)](#) in her master dissertation illustrate educational events.

The user of *MoViA* demands previous authentication using *Google+ Sign-in* which allows associating the author with the media elements (video and annotations).

While recording a video, a user can use the “*mark*” option offered by *MoViA* to associated *bookmarks* to the corresponding time instant; the bookmarks are represented as text annotations at the time of playback.

When the application, a user can select one video from a list with all videos recorded with the device to playback the video or add an annotation. The annotations supported by

Figure 4 – *MoViA* playback and navigation as presented by Cunha, Machado Neto and Pimentel (2013).

MoVia are text annotations, audio annotations and ink annotations. These annotations are linked to specific time interval of a video element, the text and ink annotations play over the video. In the case of audio annotations, the video is stopped while the audio is playing.

Figure 4 (a) shows a screenshot of *MoViA* playing a video with text and ink annotations. The figure includes:

- a timeline-based navigation bar;
- labels which indicate the reproduction time and the duration of the video;
- a play/stop button;
- an annotation area with a text field and a button to add a text-based annotation;
- an on/off ink annotation button that stops the video playback to allow the user to add an ink-based annotation;
- an on/off audio annotation button that also stops the video playback to allow the to record an audio annotation;
- a button to activate the navigation mode, which allows navigation based on a list of available annotations as shown in Figure 4 (b);
- a button to activate the sharing annotations mode, which allows sharing the annota-

tion using the default sharing applications installed in the device;

- a drop-down button which allows selecting annotations from a given author;
- a field that shows the content of an annotation present in the timeline at the time of the corresponding playback.

The original MoViA tool allowed users to share annotations, which would make sense if users had previously shared the corresponding video. This means that, even though a user could capture a video and annotate it, the annotations and the video should be shared as separate entities. To provide an alternative for users to share the video and their annotations, a first integration of the MoViA and I+WaC-Editor was carried out as discussed next.

3.2.4 MoViA and I+WaC-Editor first integration

To provide users of MoViA tool with a web-based platform in which they could share their videos and annotations, a first integration of the MoViA and I+WaC-Editor was carried out. The integration was aimed at supporting a scenario of recording and publish a video in a educational event, as the one shown in Figure 5.

*John is a student that wants to record a class. John takes his smartphone and open MoViA app. He selects **High Quality** video option and type the name of the event. He starts recording and make some bookmarks in the important moments (according to his own personal consideration). At the end of the class, John finishes the recording and decides to playback it back to confirm that everything was captured correctly. John uses the export annotations functionality and sends the annotations with the original video to the webmaster responsible for the I+WaC-Editor. A webmaster carries out an exporting process running batch scripts in the server. Finally, the webmaster gives access to the recording in I+WaC-Editor for John and notifies him. John plays back his recording with the annotations in I+WaC-Editor and, then, he shares it with his colleagues.*

Figure 5 – Scenario of capture of an educational event by one student, with *post hoc* sharing.

Source: Elaborated by the author.

The scenario of multimedia production shown in Figure 5 can be supported by the integration of original version of MoViA (recording and making annotations over video) and of the I+WaC-Editor (playback the video recorded with MoViA), as illustrated in the *UML use case* diagram (Figure 6). The diagram shows the main functions that the original applications supports: the focus in the capture and the enrichment of a single video medium.

Figure 6 – UML use case diagram that present the original main functions of *MoViA* and *I+WaC-Editor* in the context of multimedia production model.

3.3 Final remarks

In this chapter we reviewed related work and results from our research group which were used in the development of the research detailed in the the remainder of this dissertation. While discussing adopted approaches, these works provided both requirements and guidance for our research towards supporting the ubiquitous collaborative production of interactive and extensible multimedia documents while exploiting user-generated content captured via mobile devices.

CI+WAC-IE CONCEPTUAL MODEL

[Martins \(2014\)](#) proposed two models in the context of his doctoral research in amateur-oriented multimedia authoring, a *formalization of an extended graph-based temporal layout model* and a *document annotation and algebra for document transformation model*. The first model, named Interactors+Watch and Comment Interaction Events (I+WaC-IE), abstracts the concepts of annotation and interaction event, created or collected in the multimedia production process stored in a document media that had been previously created according to the *Watch & Comment paradigm* ([CATTELAN et al., 2008a](#)).

After introducing the *I+WaC-IE* model by [Martins \(2014\)](#), this chapter presents an extension of the original model so as to expand its scope to the capture phase in which the original media is recorded.

4.1 I+WaC-IE conceptual model by [Martins \(2014\)](#)

In the work by [Martins \(2014\)](#), the main concept of the *I+WaC-IE* model is the use of annotations to associate an interaction event with a media document. The annotations are associated with the media document as media elements part of the media document. It is the reason why his model allows “*annotations of annotations*”.

In *I+WaC-IE* model, a single annotation can be associated with a specific media element, a group of media elements or a media document. The annotations are associated with at least one interaction event. Also, the annotations comprise the concepts: *a) group concept*, which classifies the annotations in common shared attributes; *b) Comment concept*, which refers to an arbitrary media resource that share all the features of any annotations (being an annotation, it can participate as a media element in the visualization); *c) Effect concept*, which defines an interactive temporal effect on an annotation in the visualization (this is a virtual edition operation and does not modify the document).

Figure 7 – I+WaC-IE conceptual model by Martins (2014)

Moreover, the model by Martins (2014) defines the following dimensions, which allow structuring and associating the concepts involved in the multimedia authoring process as illustrated in Figure 7:

1. **Media Dimension:** The *media dimension* is composed of a media document that contains one or many media elements. All media elements are synchronized in a timeline that the media document should provide. A media element can be a video, digital ink, text, audio or image. The *I+WaC-IE model* does not impose a document format, and the document format should allow addressing fragments of the media document and its media elements.
2. **Interaction Dimension:** The *interaction dimension* is composed of the type of agent that participate in the interaction event and an interaction type in which these agents engage. This agent can be a human or an object. The human is any participant and the object is any entity relevant for the event. In the context of interaction event, the interaction between the agents can fit into the interaction event classification where a human can be a user, and an object can be an equipment or media. The type of interaction between the agents can be classified as nonverbal or verbal.
3. **Temporal Dimension:** The *temporal dimension* composed of original time (occurrence of the captured media in the real world) and derived time (corresponds

to the temporal scope of the resulting media). Both, original and derived time are represented by a single interval. This temporal dimension is represented by a timeline.

4. **Spatial Dimension:** The *spatial dimension* is composed of original space (spatial framing of an interaction event in the real world) and derived space (spatial representation or rendering of the interaction event in the visual media elements).

It can be observed, in the original model, that there is no concept that allows the identification of the author associated with the authoring tasks at the time that the original media was captured. It can also be observed that the model by Martins (2014) assumed that annotations and other interactions can be made by one or more *Human* or *Object* agents on previous existing media. In other words, by focusing on annotations over existing media, the original model does not take into consideration the authors of the original media, which is consistent with *Watch & Comment paradigm* (CATTELAN *et al.*, 2008a) as originally aimed by Martins (2014).

The main objective of the work reported in the remainder of this chapter is to extend the original model by Martins (2014) so as to expand its scope to the capture phase in which the original media is recorded. As a result, the extended model will also support the authors involved in the capture phase.

4.2 Extended Collaborative I+WaC-IE model

In this work, the Collaborative Interactors+WaC-Interaction Events (CI+WaC-IE) model is proposed as an extension of *I+WaC-IE* model by Martins (2014). The main objective of the extension is to allow the model to support collaborative capture scenarios by supporting the identification of the authors involved in the capture phase.

Figure 8 shows the extensions of the original model by extending the original model shown in Figure 7 with a new concept and four new relationships: the extensions are highlighted in bold font in the figure.

In summary, the extensions are as follows:

- A new concept **Author** to identify authors of annotations, documents and media elements;
- Three new relationships named **has author**, which allows associating authors with *Annotation*, *Document* and *MediaElement*;
- A new relationship **is a** which allows specifying that *Comment* concept is also a *MediaElement*.

The impact of the proposed extensions are discussed in the following sections.

Figure 8 – Collaborative I+WaC-IE model extended from Martins (2014): novel elements indicated in bold font.

4.2.1 Impact on Capture & Access

This work proposes the inclusion of the concept of **Author** as the person who originates, generates or creates a specific multimedia content. The *Author* concept was connected with the concepts of *Annotation*, *Document* and *MediaElement* with relationships named “**has author**”. These associations define explicitly the property of the concepts previously mentioned.

The proposed extension has the following impacts in the capture and access phases of the multimedia production process:

- **Capture:** This extension makes it possible the existence of a relationship *has author* between the *Document* and the *MediaElement* concepts with *Author* concept.

The association of the *Author* concept with the *MediaElements* concept each time that a participant generates a medium at capture time allows that information that identifies the author is stored and linked with the medium. As a result, by using the identification of authors that participated in the capture of the basic *MediaElements* as well as the resulting *Document* concept, it is possible to identify not only who participated in the authoring process but also if the capture was collaborative or not.

- **Access:** The original model the *InteractionEvent* has a relationship (*has agent*)

with the *Agent* concept that are the participants of an interaction event: the model states that the agent could be a *Human* or an *Object* concept. As a result, one can not assume that the *Human* is the author of an annotation. Since in the capture stage the extension allows the identification of the author, as detailed above, every time a participant captures an explicit interaction event, as an annotation, the identification of the author is stored and linked with the *Document* or *MediaElement*. Moreover, the extension allows the identification of many participants who generate bookmarks (text annotation) independently from one another. Each bookmark contains the unique identification of the author in the timeline.

4.2.2 Impact on Interactors

The proposed extension has also a significant impact in the *interactors model*, since it is now possible to use the different *interactors* along with the **Author** attribute. This can be observed in Table 3, in which illustrates new alternatives of operators which can take advantage of the *Author* concept as attribute of the original operators in the context of collaborative work. A table with the original operators is in Annex 4.

It is important to observe that, as shown in the Category column in Table 3, because the extended model defines that **Author** is now a concept, it can be applied as an attribute to all Interactors.

The Interactor Algebra for Multimedia Document transformations (IAMmDocT) proposed by Martins (2014) uses the *Interactors* model for automatic and semi-automatic authoring. The proposed extension extends the original set of *Interactors* by specifying the **Author** of the actions associated with both automatic and semi-automatic authoring.

Category	Features	Name	Description
All Interactors	time	filterByAuthorValue	list of time moments when an author or group of authors are associated with media elements
		filterByMainAuthor	list of time moments when the main author (author of the Document) is associated with media elements
		startMediaPlaybackByAuthor	list of time moments when the media of a specific author has started reproduction
		endMediaPlaybackByAuthor	list of time moments when the media of a specific author has finished reproduction

Category	Features	Name	Description
	action	recordedMomentsByAuthor	list of time moments when media elements have been recorded by an given author
		noRecordedMoments	list of time moments which are not associated with any author

Table 3 – Example of new *Interactors* focused in the *author* concept

4.2.3 Impact of “is a” relationship between *Comment* and *MediaElement* concepts

The relationship “is a” between the concepts *Comment* and *MediaElement* was created as an explicit clarification that a *Comment* of an *Annotation* is also *MediaElement*. Before this clarification, a *Comment* could be considered as additional information about an *Annotation* without a definition about their structure and could generate confusion about the differences between a *comment* and a *text annotation* of a particular *MediaElement*. With the explicit relationship, a *Comment* that is a product of user-media interaction over an arbitrary media resource and his nature is also a media resource, such as audio, video, text, ink and so on. This benefits users reviewing and annotating documents.

4.3 Final remarks

As presented in Section 2.4, the *capture & access multimedia production process* specifies six general stages. Also, the original *I+WaC-IE* model had focus on user-event interactions which were materialized with the *Annotation*. Then, annotations made by users while watching a video were in accordance with the *Watch & Comment paradigm* discussed by Cattelan *et al.* (2008a). As a result, the model allowed a collaborative extension over a basic *Document* and its corresponding *MediaElements*. The model assumed that these basic *Document* and *MediaElements*, products of the capture stage, were generated by an anonymous author because, in the model, the author information was not relevant for the focus of the original model.

The extension presented in this chapter expands the scope of the original model to the capture phase in which the original media is recorded. In the next two chapters we present, respectively, the design and the implementation of a proof-of-concept prototype that applied the extended model to collaborative authoring achieved by allowing a group of users capturing media using mobile devices.

CMOVIA AND CI+WAC-IE MODELS

The extensions presented in the previous chapter expand the scope of the original model by [Martins \(2014\)](#) so that it is possible to identify authors involved in the capture phase – the capture phase in which the original media is recorded.

In order to demonstrate how the extended model can be applied in the support of collaborative authoring, in this chapter we describe its use in the design of a proof-of-concept prototype which allows a group of users to collaboratively author a multimedia document by capturing multiple media using mobile devices.

This chapter is organized as follows. Section [5.1](#) contextualizes a user scenario for a multimedia capture, besides identifying functional and non-functional requirements for the scenario. Section [5.2](#) details the software design proposed to the prototype. It includes the software development process, the architecture and the prototype behavior. Section [5.3](#) details the data model of the components of the prototype.

5.1 Requirements gathering

[Sommerville \(2006\)](#) defined the Requirement engineering as “*the process of finding out, analyzing, documenting and checking the services and constraints about what a system should do*”. Following the Requirement engineering, we study and analyze a case of collaborative multimedia capture, the existed applications and researches that tried to solve partially or completely that scenario. With all of that, we propose a scenario description along with an UML use case diagram. Both helped us to highlight the existed differences with the originals *MoViA* and *I+WaC-Editor*. Finally, we identify functional and non-functional requirements that were developed in the prototype proposed.

John and Jane are students that want to record their math class. John takes his smartphone and open CMoViA app. He selects the video option, creates a new event called ``geometry'' and starts recording. A short time later, Jane arrives to the class, takes her smartphone and opens CMoViA app. She realizes that the event already exists. She watches the context information about the class recording and observes that John has already been recording in video. So, she decides just take photos plus audio because she thinks that it is very important have good photos of the math in the board. She joins the event and start recording. John and Jane make bookmarks every important moment under they personal consideration to review in the future. At the end of the class, both students finish the recordings and play them back to verify the quality of the capture. They then decide to export their recordings to CI+WaC. Mark, another student, asks them to watch the recordings. At home, John uses his laptop to login into CI+WaC, gives permission and share the event with Mark. The event contains both recordings synchronized with the bookmarks. Mark uses his laptop to login into CI+WaC and to playback the recording. He realizes that the audio recorded by Jane is better, so he decides to playback again but this time using John's video with Jane's audio.

Figure 9 – Scenario of collaborative capture of an educational event by three students.

Source: Elaborated by the author.

5.1.1 Scenario description

At the moment, when we took the originals *MoViA* and *I+WaC-Editor*, we realized that they solved partially the problems of the scenario described before. Even, they dealt with the tasks of recording and publishing applied in educational event, the tasks for share and synchronization could take so much effort or high costs. This problem came from the semi-automatic integration (the media elements taken with *MoViA*, must interchanged manually) and the non collaborative capture (in a scenario of two people recording an event, the synchronization must be done manually or computer processing of any video feature, for example, using fingerprints).

The scenario, used by the original applications, focused in a single student recording video and making bookmarks in a classroom with *MoViA* app. At the end, the student can replay the video, adding the different annotations and share the video and text annotations with a person that upload manually to *I+WaC-Editor* for visualization and enrichment of the single video recording. This scenario was described in Section 3.2.4

We proposed a scenario of a collaborative capture in an educational event, where three students — John, Jane and Mark — record a math lecture. The scenario is described in the Figure 9. We represent the scenario into a UML use case diagram (Figure 10). We

Figure 10 – UML use case diagram that present then main functions of *CMoViA*.

The dark gray use cases have existed in the originals (*MoViA* and *I+WaC-Editor*), but we modified or extended them. The light gray use cases are new ones, created for the collaboration context. The use cases shown in white already existed in the original use cases.

Source: Elaborated by the author.

identify three roles in the diagram, the *CMoViA app* user, the *CI+WaC* user and the *CI+WaC* user manager. The different functions (use case) performed by each user were grouped following the stages from multimedia production process. The diagram exhibit that both applications (*CMoViA app* and *CI+WaC*) perform tasks of visualization and enrichment with annotations, but only *CMoViA app* focus on collaborative capture tasks.

The UML use case diagram (Figure 10) presents use cases in different background color: white, dark gray and light gray. 1) The white ones are use cases without modification, they already existed in the original version of the applications and we just reused them. 2) The dark gray ones are existed use cases that we modify or extend for the collaborative scenario. 3) The light gray ones are new use cases that we create in the context of this work. The new UML use case diagram highlights the functions for collaborative work and exportation.

Comparing the two UML use case diagrams (Figure 10 and Figure 6), we observe that the new UML use case diagram includes formal phases for the pre-production and publish into the context of multimedia production process. The pre-production phase manages the tasks for collaborative recording and the publish phase automatizes the publishing with the tasks for automatic exportation, mixing and synchronization. This phases were not considered in the scope of the originals *MoViA* and *I+WaC-Editor* because they were contextualized on a single recording scenario for *MoViA* and a single main video for *I+WaC-Editor*.

5.1.2 *Functional and non-functional requirements*

Sommerville (2006) classified the requirements as user or system requirements to define the responsibilities and actions that they are committed to do. As we have contextualize in an educational scenario the user requirements use the *student* as user and the system requirements use any component (*CMoViA app*, *CMoViA API* and *I+WaC-Editor*) of the prototype as system.

Based on the objectives, problems detected, literature review and our own experience in collaborative capture, we identify a group of functional and non-functional requirements

5.1.2.1 *Functional requirements*

1. The student should be able to create a new collaborative event just with the event name.
2. The student should be able to watch the available events (events in recording process, in real time).
3. The student should be able to watch the context information (whom were/are recording, which media they are recording and also a photo preview of their own

visual recordings) about an available event.

4. The student should be able to select the media that he wants to record.
5. The student should be able to join into an available event.
6. The student should be able to recording a real live experience by the media selected.
7. The student should be able to stop and replay the recording as desired.
8. The student should be able to make bookmarks of his personal important moments while he is recording the event.
9. The student should be able to select, playback and browsing any of his recordings.
10. The student should be able to make annotations while he is playing his recording.
11. The student should be able to use a Bluetooth headset accessory as alternative for audio recording.
12. The student should be able to use a remote shutter accessory as alternative for making bookmarks or stopping the recording.
13. The student should be able to configure the application (camera quality, interval time for periodical photos, use of Bluetooth headset accessory for recording).
14. The student should be able to export explicitly or automatically his recording to *CI+WaC*.
15. The student should be able to playback the collaborative capture (his own recording joined with the other students recordings) in *CI+WaC*.
16. *CMoViA app* should perform an automatic exportation only when the mobile is in charge and has WiFi Internet connection.
17. In playback with *CI+WaC*, the student should be able to interchange between all the same type media captured.

5.1.2.2 Non-functional requirements

1. The synchronization latency should be not grater of 1 second.
2. *CMoViA app* should not present more than four screens for starting a new recording.
3. *CMoViA app* should require confirmation before finish a recording.
4. In exporting and synchronization, the communication interface between the mobile application and the server should be generic so, other clients could participate and integrate their own recordings as long as they respect the communication interface.

5.1.3 *CI+WaC-IE model influences*

In the chapter 4, we discuss about the changes that the addition of the *Author* concept in the *CI+WaC-IE* model. Now, the model supports authoring and capture tasks in a collaborative scenarios. So, in this section we analyse the influence that the model has over the scenario proposed and the set of requirements.

The *CI+WaC-IE* model exhibits that many concepts are specifications of other concepts — the concepts of *Human* and *Object* are specification of the concept of *Agent*. The traceability matrices show that many concepts of the model have no influence on any component. This is because visualization and authoring are out of the scope of the work in this dissertation.

5.1.3.1 Model influences over scenario

Section 5.1.1 detailed a scenario of collaborative capture and visualization. This scenario was represented in a *UML use case diagram* (Figure 10). With the objective of verify the influence of the *CI+WaC-IE* model has over the collaborative scenario, we designed a Traceability matrix (Figure 11) that highlights the correspondences of each concept of the model with the components of the *UML use case diagram* (actors and use cases grouped following the multimedia production process). The matrix shows the strong influence that the *Author* and *MediaElement* concepts have over the different elements of the *UML use case diagram*, because they are present in the majority of the elements.

		UML CMoViA																												
		Actors			Pre-production					Capture					Publish		Access				Extension									
		Student in Movia	Student in I+WaC	Manager	Configure the application	join event	create event	select a collaborative event	select medium to record	watch context information	recording media	periodic photos	video	audio	finish recording	make bookmarks	export to I+WaC	mix sessions	playback	select media	browsing	interact with recording	view content	make audio annotation	make an ink annotation	make a text annotation	add annotation	perform nan effect	make a comment	
C-I+WAC-IE Model	Media dimension	Document				↳	↳										↳	↳	↳	↳	↳									
		MediaElement							↳		↳	↳	↳	↳	↳	↳	↳	↳	↳	↳	↳	↳	↳	↳	↳	↳	↳	↳		
		Video							↳		↳		↳																	
		Ink																								↳				
		Text							↳		↳			↳		↳											↳			
		Audio							↳		↳														↳					
	Annotation	Image							↳		↳	↳																		
		Author	↳	↳			↳	↳		↳	↳	↳	↳	↳		↳	↳	↳			↳	↳	↳	↳	↳	↳	↳	↳		
		Annotation																			↳	↳	↳	↳	↳	↳	↳	↳	↳	↳
		Comment																											↳	↳
		Effect																											↳	
		Interaction dimension	Group																											
	InteractionEvent																							↳	↳	↳	↳	↳		
	Agent			↳																			↳		↳	↳	↳	↳		
	Human			↳																					↳	↳	↳	↳		
	Object																													
	Space dimension		InteractionType																				↳		↳	↳	↳	↳		
		nonverbal																												
		verbal																												
	Time dimension	SpatialEntity																								↳				
		PresentationSpace																							↳					
		ValidSpace																												
	Time dimension	TemporalEntity																									↳			
		PresentationTime																									↳			
		ValidTime																												

Figure 11 – Traceability matrix of *CI+WaC-IE* model to *UML use case diagram*

Source: Elaborated by the author.

5.1.3.2 Model influences over requirements

Section 5.1.2 compiled a set of functional and non functional requirements that a mobile application for collaborative capture and visualization should include. To verify the influence that the *CI+WaC-IE model* has over the different requirements, we designed a Traceability matrix (Figure 12) that highlights the correspondences of each concept of the model with the different requirements. The matrix shows the strong influence that the *Author*, *Document* and *MediaElement* concepts have over the different requirements, because they are present in the majority of the elements.

			Requirements																				
			Functional																	Non-functional			
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	1	2	3	4
C-I+WAC-IE Model	Media dimension	Document	→	→			→		→		→		→		→		→	→					→
		MediaElement			→	→		→	→	→	→	→	→		→	→	→					→	
		Video																					
		Ink																					
		Text								→				→									
		Audio											→										
		Image																					
	Annotation	Author	→		→		→	→		→			→	→		→	→	→					→
		Annotation								→		→	→	→									
		Comment																					
		Effect																					
		Group																					
		InteractionEvent								→		→	→										
	Interaction dimension	Agent								→		→	→	→									
		Human																					
		Object											→	→									
		InteractionType								→		→	→										
		nonverbal																					
		verbal																					
	Space dimension	SpatialEntity																					
		PresentationSpace																					
		ValidSpace																					
	Time dimension	TemporalEntity																					
		PresentationTime																					
		ValidTime																					

Figure 12 – Traceability matrix of *CI+WaC-IE model* to functional and non functional requirements

Source: Elaborated by the author.

5.2 Software design

This section describes the software development process (Section 5.2.1), architecture (Section 5.2.2), a description of the prototype's behavior in a collaborative capture (Section 5.2.3) and an analysis of the influence that the *CI+WaC-IE model* has over the software design.

5.2.1 Software development process

Our work follows the interactive development process. The requirements defined in Section 5.1.2 were implemented, tested and evaluated in real scenarios of collaborative capture. This process was repeated in an interactive and incremental way, improving the existing requirements and proposing new ones. The prototype developed in this work had two main cycles and produced a stable version at the end of each cycle.

5.2.1.1 First cycle: MoViA2

MoViA version 2 (MoViA2) was the product generated at the end of the first cycle. *MoViA2* is an extension of *MoViA*. It was inspired in the context and problems from collaborative mobile capture of educational events. In the scenario in which a small group that want to make a multimedia collaborative capture and they are located in a remote place with limited WiFi and mobile network access. We proposed a mobile alternative using *MoViA2*. It proposed: 1) *an explicit physical synchronization* using the accelerometer sensor (Section 6.1.2.1.1), 2) *an interchange of multimedia session documents* between the members of the group using Bluetooth, 3) *union of different multimedia sessions* with the objective of create a collaborative session document and 4) *a multimedia document player*. *MoViA2* was used in four real scenarios in the class room context (graduate class, summer school, summer course and undergraduate end course project, as detailed in Section 6.2.1.

MoViA2 exhibited limitations: 1) *the synchronization of more than 4 devices demands one person making the physical **shake event** holding all devices*, which can be difficult for the average user; 2) *long waiting times when large media files are involved in the interchange*, *MoViA2* uses a Bluetooth network to share the media elements so the media interchange is limited by bandwidth; and 3) *it was impossible to include a new participant into the collaborative capture after it had been initiated* because the synchronization needs to be done at the beginning.

5.2.1.2 Second cycle: CMoViA

CMoViA was the product generated at the end of the second cycle. *CMoViA* is composed by: 1) *CMoViA app* component, that is an extension of *MoViA2* and 2) *CMoViA API* component, that is responsible for the synchronization and management the context information. *CMoViA* solved the limitations of *MoViA2*, described before, with a different synchronization mode (details in Section 6.1.2.1.2). *CMoViA* highlighted the importance of context information for an opportunistic multimedia collaborative capture. *CMoViA* also was test in real scenarios and we proposed an usability test case.

5.2.2 Architecture

We evolved the applications previous developed by members of our research group. In the case of *I+WaC-Editor* application, we debated in the group about the influence of the requirements over the *I+WaC-Editor*. As a result, [Martins \(2014\)](#) extended his architecture to attend the new requirements. He also implemented the corresponding features for the web application, which was then called *CI+WaC*. In the case of *MoViA* application, we used the application sources and the research results offered by [Cunha \(2014\)](#). After analyzing and testing the applications, we decided to continue with the architecture style, extending the architecture and implementing the demanded extensions. The extended mobile application is called *CMoViA*.

Figure 13 represents the architecture proposed in this work. We identify the three main components that take part of our functional prototype. The design exhibits the components with different background colors: 1) the light gray are components that we designed and created for this research; 2) the dark gray ones are existed component that we redesign and extend; 3) the transparent or white ones are existed or original components that we did not modify at all.

- *CMoViA app* component interacts with the components *CMoViA API* and *CI+WaC API* with HTTP Requests. *CMoViA app* component has four sub-components, 1) *media recording* component redefined the original capture process from only video recording to multimedia recording (video, audio, periodic photos and bookmarks); 2) *recording configuration* component defines parameters for recording: the default quality values for camera recoding, default periodic time for taking photos and mobile accessory configuration; 3) *media playback* component has a set of tasks for: a) the visualization of media elements, the creation and visualization of annotations (*annotation* component); b) manage the drawing area for the design of digital ink annotations (*drawing area* component) and c) manage the operations for navigation of the media elements in visualization (*navigation* component); 4) *collaborative operation* component has a communication task with the *I+WaC API* component, it makes operations like: a) authentication using Google+ Sign-in (*Google+ authentication* component); b) creation and synchronization of collaborative sessions (*session synchronization* component) and c) single exportation to *CI+WaC* component (*session exportation* component).
- *CMoViA API* component offers a set of operations for management of synchronization and context information. The clients access to the operations for collaborative capture using HTTP Requests.
- *CI+WaC API* component is a sub-component of *CI+WaC* component that interacts with *I+WaC-Editor* component with an interface. This interface take cares of the

Figure 13 – *CMoViA* architecture represented by a UML component diagram.

The dark gray components have existed in the originals (*MoViA* and *I+WaC-Editor*), but we in this work we modified or extended them. The light gray use cases are new ones, created for the collaboration context. The white use cases have existed in the originals and have no modification.

Source: Elaborated by the author.

obtaining and storing the data corresponding the captured session. *CI+WaC API* component has a *union session* sub-component that uses a synchronization token to mix sessions to perform a collaborative capture. Once this process is concluded, users can visualize the individual recordings and also edit and annotate the multiple media with the *I+WaC-Editor* component.

This component does not exhibit the details that the original *I+WaC-Editor* by Martins (2014). We designed high level components with the main features corresponding to the scope of this work. Hence, is it important to observe that Martins (2014) was responsible for the modifications in the implementation of the *I+WaC-Editor* that our proposed extension required.

5.2.3 Prototype behavior

CMoViA is composed by the interaction between three applications — *CMoViA app*, *CMoViA API* and *CI+WaC* — that were created and extended with the objective of

perform an opportunistic ad-hoc collaborative capture. Figure 14 presents an overview of the interaction between the three applications using an UML sequence diagram. For this scenario, we have considered only related tasks for capture and visualization. So, the tasks for authoring or extension phases were not included.

The UML sequence diagram exhibits five lifelines, two *CMoViA app* users (*john: CMoViA app* and *jane: CMoViA app*), one *CMoViA API*, one *CI+WaC* and one *mark :CI+WaC*. The last lifeline represents anyone that wants to playback the collaborative session, using *I+WaC-Editor*. We detailed the messages of the sequence diagram into **functions** or **tasks** of the UML use cases diagram from Figure 10 in the context of the multimedia production process.

- **create event** use case takes part of the *Pre-production* stage, represent the interaction between *john: CMoViA* to *CMoViA API* by the messages from 1 to 3.
- **join event** use case also takes part of the *Pre-production* stage, represent the interaction between *jane: CMoViA* to *CMoViA API* by the messages from 6 to 13.
- **recording** use case take part of the *Production* stage, represent the interaction between *john: CMoViA app* to *CMoViA API* by the messages 4,5 and *jane: CMoViA app* to *CMoViA API* by the messages 14,15.
- **finish recording** use case also take part of the *Production* stage, represent the interaction between *john: CMoViA app* to *CMoViA API* by the message 16 and *jane: CMoViA app* to *CMoViA API* by the message 17.
- **export to I+WaC** use case take part of the *Publishing* stage, represent the interaction between *john: CMoViA app* to *CI+WaC* by the message 18 and *jane: CMoViA app* to *CI+WaC* by the messages 19,20. As *jane: CMoViA app* uploads her session after *john: CMoViA*, in the message 20, *CI+WaC* perform the union of the two sessions.
- **playback** use case take part of the *Access* stage, represent the interaction between *mark :CI+WaC* to *CI+WaC* by the message 21.

5.2.4 CI+WaC-IE model influences

In Chapter 4 we discussed the changes that the addition of the *Author* concept in the *CI+WaC-IE* model. Given that the extended model supports capture and authoring in collaborative scenarios, in this section we analyse the influence that the model has on the software design of the prototype.

The *CI+WaC-IE* model exhibits that many concepts are specifications of other concepts — the concepts of *Human* and *Object* are specification of the concept of *Agent*. The traceability matrices showed that many concepts of the model have no influence

Figure 14 – CMoViA collaborative capture and visualization represented into an *UML sequence diagram*.

Source: Elaborated by the author.

with any component. This is because 1) the architecture diagram showed only high level components related with the scope of this work, and 2) the sequence diagram took an specific scenario for capture and visualization that did not take care of all features of authoring.

5.2.4.1 Influences of the *CI+WaC-IE* model on the software architecture

Section 5.2.2 proposed an architecture for a multimedia collaborative capture and visualization system. We designed a Traceability matrix (Figure 15) which exhibits the influence that the concepts of the *CI+WaC-IE* model has over the different components of the architecture. The matrix shows the strong influence that the *Author*, *Annotation*, *Document* and *MediaElement* concepts have over the components for capturing, visualization and authoring. The components for **capturing** (*CMoVia app*, *session synchronization*, *session exportation*, *media recording* and *union session*) take care of the recording of *MediaElement* by *Author* which are part of a multimedia *Document*. The components for **visualization** (*navigation* and *media playback*) take care of the replay of the *MediaElements* taken by *Authors* which are part of a multimedia *Document*. The components for **authoring** (*annotations*, *drawing area*, *annotation operations* and *annotations visualization*) take care of the *Annotations* produced by *Authors*.

			Architecture													
			CMoViA API	CMoViA app							I+WaC Editor					
				google+ authentication	session synchronization	session exportation	recording configuration	annotations	drawing area	navigation	media recording	media playback	session management	annotation operations	annotation visualization	union session
C-I+WAC-IE Model	Media dimension	Document	↪		↪	↪				↪	↪	↪	↪	↪		↪
		MediaElement	↪		↪	↪		↪	↪	↪	↪	↪	↪	↪	↪	↪
		Video														
		Ink						↪	↪							
		Text						↪								
		Audio														
		Image														
	Annotation	Author	↪	↪	↪	↪		↪		↪	↪	↪		↪	↪	↪
		Annotation	↪			↪		↪	↪					↪	↪	
		Comment														
		Effect														
		Group														
		InteractionEvent						↪						↪		
	Interaction dimension	Agent						↪								
		Human														
		Object														
		InteractionType						↪								
		nonverbal														
		verbal														
	Space dimension	SpatialEntity							↪					↪		
		PresentationSpace							↪					↪		
		ValidSpace							↪							
	Time dimension	TemporalEntity											↪	↪		
		PresentationTime											↪	↪		
		ValidTime														

Figure 15 – Traceability matrix of *CI+WaC-IE model* to the UML component diagram

Source: Elaborated by the author.

			UML sequence diagram																	
			Actor Obj			Messages														
			John	Jane	Marc	1) select media	2) create event	4) start recording	5) upload preview	6) get event list	8) get participants of event	10) get preview photo	12) select media	13) join event	14) start recording	15) sent photo preview	16,17) left event	18,19) upload session	20) mix sessions	21) playback
C-I+WAC-IE Model	Media dimension	Document					↳			↳	↳			↳			↳	↳	↳	↳
		MediaElement				↳		↳	↳		↳	↳	↳	↳	↳	↳	↳	↳	↳	↳
		Video																		
		Ink																		
		Text																		
		Audio																		
		Image																		
	Annotation	Author	↳	↳	↳	↳	↳	↳		↳			↳	↳		↳	↳	↳		
		Annotation																		
		Comment																		
		Effect																		
		Group																		
		InteractionEvent																		
	Interaction dimension	Agent																		
		Human																		
		Object																		
		InteractionType																		
		nonverbal																		
		verbal																		
	Space dimension	SpatialEntity																		
		PresentationSpace																		
		ValidSpace																		
	Time dimension	TemporalEntity																		
		PresentationTime																		
		ValidTime																		

Figure 16 – Traceability matrix of *CI+WaC-IE model* to the UML sequence diagram

Source: Elaborated by the author.

5.2.4.2 Model influences over prototype behaviour

Section 5.2.3 detailed the behavior of the *CMoViA* application in a scenario of collaborative capture and visualization. We designed a Traceability matrix (Figure 16) which exhibits the influence that the *CI+WaC-IE model* has on the different elements (objects and messages) of the UML sequence diagram. The matrix shows the strong influence that the *Author*, *Document* and *MediaElement* concepts have on the different requirements. The UML sequence diagram shows only the behavior of capture and visualization and does not consider authoring. As a result, the *Annotation* concept has no influence on any element of the UML sequence diagram.

5.3 Data model

We proposed a conceptual model for the *CMoViA app* as illustrated in Figure 17. The model is based on a multimedia collaborative capture. We identified three main actors: 1) *the event*, the occurrence that was recorded, 2) *the author* each one of the recorders that participate in the event, and 3) *the media* the medium used for recording(audio, images, video or bookmarks). *Capture* is the main entity because it relates the main actors described before.

CMoViA is composed of three components. They are: *CMoViA API* (Section 6.1.1), *CMoViA app* (Section 6.1.2) and *CI+WaC* (Section 6.1.3). In this section we detail the model design used for each component based on the conceptual data model (Figure 17).

Figure 17 – Conceptual data model of *CMoViA*

Source: Elaborated by the author.

5.3.1 CMoViA API data model

The *CMoViA API* handles the synchronization and the management of context information demanded in collaborative multimedia capture session. We designed an entity–relationship model (ER model), shown in Figure 18, which corresponds to the features described before. The *ER model*, which follows the conceptual model presented in Figure 17, is composed of five entities:

- *event entity* stores the information about the collaborative capture sessions.
- *author entity* stores information about participants, this entity is used for authorization.
- *capture entity* stores the information about captures that the different participants perform in the collaborative capture session, this entity is the connection between event and author entities
- *media entity* contains default information about the different mediums that *CMoViA app* can capture (video, photo, audio, audio of Bluetooth headset, bookmarks)
- *capture_has_media entity* is the connection between capture and media entities, it stores the information about each individual capture (author, event and media used).

Figure 18 – ER model proposed for the *CMoViA API* application.

5.3.2 CMoViA app data model

The *CMoViA app* takes care about multimedia capture tasks. The previous version *MoViA2* managed also the combination of single recordings and multimedia visualization. Following the conceptual model shown in Figure 17, we proposed a *session document descriptor* as data model.

The *session document description* is a XML document that contains the description of a single or collaborative capture. It is the data model used for *CMoViA app*, the Source code 1 shows an example about the capture in video of one of the recorders. This document is independent of the kind of synchronization used in the capture. We use *Simple XML* framework to encode and decode the XML document. Source code 2 represents the XML Schema of the session document description.

Source code 1 – Session document descriptor used for *CMoViA app*

```

1: <?xml version="1.0" encoding="UTF-8"?>
2: <event>
3:   <creationTime>185</creationTime>
4:   <duration>84</duration>
5:   <name>HCI class2015-06-11-20-15-29</name>
6:   <resourceList class="br.usp.icmc.movia.video.recorder.CameraPictureActivity\
   $7">
7:     <resource>
8:       <author>bruce.interm</author>
9:       <mediaList class="java.util.ArrayList">
10:        <media>
11:          <duration>80</duration>
12:          <name>189</name>
13:          <path>bruce.interm-189.mp4</path>
14:          <startedTime>189</startedTime>

```

```

15:         <type>VIDEO</type>
16:     </media>
17: </mediaList>
18: </resource>
19: </resourceList>
20: <serverStartDate>1434064072</serverStartDate>
21: </event>
22:

```

The root element is called *event*, it contains sub elements for event description like: 1) *name* that is a text that represent the name of the event; 2) *creation time* is a non negative number that represents the interval time in seconds since the synchronization was made (*MoViA2*) or the collaborative event was created (*CMoViA*); 3) *duration* is a non negative number; 4) *server started date* is a non negative number that represent the time where the event was created in Unix Timestamp format¹, also it is used as a primary key to identify that different session document description belongs to the same collaborative session at the moment of the union and finally 5) *resource list* that contains a set of sub elements called *resource*.

The *resource* element represents every medium that an specific person has recorded, it contains two sub elements: 1) *author* is the name or nickname of the author recorder and 2) *media list* that contains a set of sub elements called *media*.

The *media* element represent the detailed information about a media captured like: 1) *name* is optional and represent the name of the media, 2) *started time* is a non negative number that represent the interval time since the synchronization was made (*MoViA2*) or the collaborative event was created (*CMoViA*), 2) *duration* is a non negative number, 3) *type* is one value of a predefined constant list, the constants were *VIDEO*, *AUDIO* and *IMAGE*, finally 4) *path* is an string that represent the physical location of the medium into the mobile device.

The *serverStartDate* element is the token shared between all participants of a specific collaborative capture, the subsection explains the union session process. The *session_token* element from Source code 3 is stored into *serverStartDate* element our document description in Unix Timestamp format (Source code 2).

Source code 2 – XML Schema of the document descriptor used for *CMoViA app*

```

1: <?xml version="1.0" encoding="UTF-8"?>
2: <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" attributeFormDefault="
    unqualified" elementFormDefault="qualified">
3:   <xs:element name="event">
4:     <xs:element type="xs:int" name="creationTime" />

```

¹ “Timestamp is a system for describing points in time, defined as the number of seconds elapsed since midnight proleptic Coordinated Universal Time (UTC) of January 1, 1970, not counting leap seconds.” Source <<http://unixtimestamp.50x.eu/about.php>>

```

5:     <xs:element type="xs:int" name="duration" />
6:     <xs:element type="xs:string" name="name" />
7:     <xs:element type="xs:int" name="serverStartDate" />
8:     <xs:element name="resourceList">
9:         <xs:attribute type="xs:string" name="class" />
10:        <xs:element name="resource">
11:            <xs:element type="xs:string" name="author" />
12:            <xs:element name="mediaList">
13:                <xs:attribute type="xs:string" name="class" />
14:                <xs:element name="media">
15:                    <xs:element type="xs:int" name="duration" />
16:                    <xs:element type="xs:string" name="name" />
17:                    <xs:element type="xs:string" name="path" />
18:                    <xs:element type="xs:int" name="startedTime" />
19:                    <xs:element type="xs:string" name="type">
20:                        <xs:simpleType>
21:                            <xs:restriction base="xs:string">
22:                                <xs:enumeration value="VIDEO" />
23:                                <xs:enumeration value="AUDIO" />
24:                                <xs:enumeration value="IMAGE" />
25:                            </xs:restriction>
26:                        </xs:simpleType>
27:                    </xs:element>
28:                </xs:element>
29:            </xs:element>
30:        </xs:element>
31:    </xs:element>
32: </xs:element>
33: </xs:schema>

```

5.3.3 CI+WaC integration

The *session descriptor* document already existed in the original *I+WaC-Editor*. It is a *JSON* object which is used for visualization and multimedia authoring over a single video. This document was based in two main entities, the *media elements* and the *timelines*. A *timeline* includes sub elements with *events*. The *events* are described as text annotations. Each element has an own description like begin time, end time, begin session time, end session time, file name, label, comment, etc. Source code 3 shows the detailed description of a *session descriptor* document.

Source code 3 – *Session descriptor* description used for *CI+WaC*

```

1: {
2:     "id": STRING,
3:     "status": STRING,
4:     "origin": STRING,
5:     "begin": NUMBER,
6:     "end": NUMBER,
7:     "title": STRING,
8:     "type": ENUM,
9:     "session_token": NUMBER,

```

```

10:     "date_created": NUMBER,
11:     "media_elements": {
12:         "MEDIA_ELEMENT_1": {
13:             "begin": NUMBER,
14:             "end": NUMBER,
15:             "session_begin": NUMBER,
16:             "session_end": NUMBER,
17:             "filename": STRING,
18:             "type": ENUM
19:         }
20:         (...)
21:         "MEDIA_ELEMENT_N": {
22:             (...)
23:         }
24:     },
25:     "timelines": {
26:         "TIMELINE_1": {
27:             "label": STRING,
28:             "date_created": NUMBER
29:             "events": {
30:                 "EVENT_1": {
31:                     "begin": NUMBER,
32:                     "end": NUMBER,
33:                     "session_begin": NUMBER,
34:                     "session_end": NUMBER,
35:                     "comment": STRING,
36:                     "label": STRING,
37:                     "agents": [
38:                         {
39:                             "role": "author",
40:                             "label": STRING
41:                         }
42:                         (...)
43:                     ]
44:                 }
45:                 (...)
46:                 "EVENT_N": {
47:                     (...)
48:                 }
49:             }
50:         }
51:     },
52:     "owner": [STRING],
53: }
54:

```

Both *CI+WaC* and *CMoViA* applications extended and re-utilized document for communication and interchange document in the exportation of a capture. Source code 4 shows an example of a *session descriptor* generated with *CMoViA app*. The capture could be single or collaborative capture. *CMoViA app* took care to adapt the *session document descriptor* (data model of *CMoViA app* Source code 1) for *session descriptor* (data model of *CI+WaC* Source code 3).

Source code 4 – *Session descriptor* example generated with *CMoViA*, exported to *CI+WaC*

```

1: {
2:   "parents":[ ],
3:   "status":"active",
4:   "origin":"movia",
5:   "date_created":1434050993,
6:   "type":"movia",
7:   "session_token":1434050714,
8:   "begin":1434050993,
9:   "id":"bruce.interm_1434050993",
10:  "title":"MoViA: hora2015-06-11-16-29-52",
11:  "date_recorded":1434050993,
12:  "timelines":{
13:    "bruce.interm_1434050993_text":{
14:      "label":"MoViA Annotations by bruce.interm",
15:      "date_created":1434050993,
16:      "taxonomy_class":"wac",
17:      "events":{
18:        "text100":{
19:          "behavior":"default",
20:          "agents":[
21:            {
22:              "role":"author",
23:              "label":"bruce.interm"
24:            }
25:          ],
26:          "session_end":4,
27:          "date_created":1434051002,
28:          "session_begin":0,
29:          "comment":"Beginning capture of bruce.interm",
30:          "end":1434050997,
31:          "begin":1434050993
32:        }
33:      },
34:      "date_updated":1434051002
35:    }
36:  },
37:  "media_elements":{
38:    "hora2015-06-11-16-29-52":{
39:      "filename":"bruce.interm-10.mp4",
40:      "session_end":52,
41:      "session_begin":1,
42:      "type":"video",
43:      "end":1434051045,
44:      "begin":1434051003
45:    }
46:  },
47:  "date_updated":1434051002,
48:  "date_archived":1434050993,
49:  "owner":[
50:    "bruce.interm"
51:  ],
52:  "end":1434051045
53: }

```

In a multimedia collaborative capture scenario, each participant with *CMoViA app* exports their recording using this document. The change to identify if a session descriptor belongs a collaborative capture is with the element `session_token` (element of the session descriptor in Source code 3). It should be created for the *CMoViA API* (collaborative capture) or the *MoViA2* (single or collaborative). When *CI+WaC* identifies a collaborative session. It creates a new session. It merges all the media elements and the annotations from the single captures.

We masked the model of the *session descriptor* into the conceptual model (Figure 17). The entities of *Capture* e *Media* are explicitly represented in the *session descriptor* in the context of single recordings. The entity *Author* included in the element with key *owner*, the key of the *media_elements* list and also in the description of *timeline/events/agents*.

5.3.4 CI+WaC-IE model influences over the data models

Section 5.3 exhibits the data model (Figure 17) used in the collaborative capture and visualization system applied to the three components of *CMoViA*. we designed a Traceability matrix (Figure 19) that highlights the influence that the *CI+WaC-IE model* has over the principal elements of each data model.

The *CMoViA app* and the *CMoViA API* are in charge of capture and visualization phase. Because of that, the matrix confirmed that by the strong influence of the *Media dimension* has over the data model of *CMoViA app* and *CMoViA API*. Also the matrix shows that the data model of *CI+WaC* was influenced with the other dimensions besides *Media dimension*. It was because, the data model of *CI+WaC* was focus into visualization and authoring.

5.4 Final remarks

In this chapter, we described and detailed a scenario for collaborative capture, requirements, architecture, behavior and data model about the prototype. Along with that, we used traceability matrices that helped us to identify the influence of the *CI+WaC-IE model* over the artifacts of software analysis and software design associated with the prototype in a scenario of collaborative capture and visualization.

The traceability matrices exhibit the influence that the concept of *Author* and the concepts inside of the *media dimension* have over the *CMoViA app* and the *CMoViA API*. They confirm the influence that those concepts have in the capture tasks.

The traceability matrices also show the influence of the other dimensions besides the concept of *Author* and the concepts inside of the *media dimension*. This is because

			Data Model															
			CMoViA API				CMoViA app					CI+WaC						
			event	capture	author	media	event	-resource	--author	---media	----mediaType	id	session_token	owner	media elements	\-type	timelines	\-events
C-I+WAC-IE Model	Media dimension	Document	↪	↪			↪	↪						↪				
		MediaElement		↪		↪		↪		↪				↪				
		Video				↪				↪					↪			
		Ink																
		Text				↪				↪							↪	↪
		Audio				↪				↪					↪			
		Image				↪				↪					↪			
	Annotation	Author			↪				↪		↪		↪					↪
		Annotation															↪	↪
		Comment															↪	
		Effect																
		Group																
		InteractionEvent															↪	
	Interaction dimension	Agent																↪
		Human																
		Object																
		InteractionType															↪	
		nonverbal																
		verbal																
	Space dimension	SpatialEntity																
		PresentationSpace																
		ValidSpace																
	Time dimension	TemporalEntity															↪	
		PresentationTime															↪	
		ValidTime															↪	

Figure 19 – Traceability matrix of *CI+WaC-IE model* over the data models of *CMoViA app*, *CMoViA API* and *CI+WaC*.

Source: Elaborated by the author.

the *CI+WaC* focuses in visualization and in semi-automatic and automatic authoring. Visualization tasks are related with the *media dimension* and authoring tasks are related with the other dimension of the *CI+WaC-IE model*.

This chapter shows the importance and applicability that the concept of *Author* has over every stage on the multimedia production process.

CMOVIA IMPLEMENTATION

Chapter 4 presented the extended *CI+WaC-IE* model which is used in the context of collaborative capture, visualization and authoring. Chapter 5 details the application of the *CI+WaC-IE* model into the analysis and design of a software solution. This chapter presents details of the implementation and of the functionality of the *CMoViA* application.

This chapter is composed by the details of *CMoViA* (Section 6.1) — *CMoViA API* (Section 6.1.1), *CMoViA app* (Section 6.1.2) and *CI+WaC* (Section 6.1.3) —, tests in real scenarios (Section 6.2) and a discussion (Section 6.3).

6.1 Collaborative MoViA

The Collaborative Mobile Video Annotation (*CMoViA*) tool is a collaborative multimedia capture tool that focus on opportunistic *ad hoc* capture by amateur users for multimedia collaborative capture of real live events. The tool composed by a front-end mobile application (*CMoViA app*), a back-end application (*CMoViA API*) and a integration application with *CI+WaC* web application.

6.1.1 *CMoViA API*

The *CMoViA API* is a REST¹ web application. It was developed using the Ruby on Rails framework² and a Oracle database³. *CMoViA API* is running over Heroku cloud application platform⁴. We decided to use the *Heroku free plan* because it offers a Platform as a Service (PaaS) that includes all technical requirements for running a web application

¹ REST application is also called RESTful web application.

² <<http://rubyonrails.org/>>

³ <<https://www.oracle.com>>

⁴ <<https://www.heroku.com/>>

and it is available in a free URI. The *CMoViA API* is available online in <https://radiant-oasis-3197.herokuapp.com/event>.

For implementation, we were supported with the version control system *Git*⁵, and the sources are stored into a *GitHub*⁶ repository (<https://github.com/omar2562/movia-server-rails>).

CMoViA API is an application that manages the context and synchronization information about the collaborative captures. *CMoViA API* and *CMoViA app* establish communication between them using HTTP Requests over a REST architecture. Ruby on Rails lets that *CMoViA API* was developed under the Model View Controller paradigm.

The *CMoViA API* provides a unique resource called *event*. The interaction with the API is made with interchange objects in JSON and XML formats using the HTTP methods (GET, POST, PUT and DELETE). The *event* resource offers eight different operations that the *CMoViA app* uses in collaborative capture. The operations are: 1) Consult events, 2) Specific event details, 3) Participants of an event, 4) Create new event, 5) Join existing event, 6) Left an event, 7) upload photo preview, 8) Recover photo preview. The Annex A shows each operation in detail of *CMoViA API description*.

6.1.2 CMoViA app

The CMoViA application (CMoViA app) is a front-end application or mobile app. I was implemented for Android platform. We took the original *MoViA* application developed by Cunha (2014) in context of her master dissertation. We extended it with features for collaborative synchronous capture and context information. *MoViA app* was implemented using *JAVA*⁷ 1.6, *Android SDK*⁸ 5.0, the native APIs and *Simple XML*⁹ framework to manage XML files. We used the version control system *Git* and the sources are stored in *GitHub* repository (<https://github.com/omar2562/MoVIA2>).

We identify main features that *CMoViA app* manages in the context of multimedia collaborative capture. The four main feature are: 1) Synchronized capture, 2) Multimedia selection, 3) Context information and 4) Mobile accessories. Those features are detailed in the next sections.

6.1.2.1 Synchronized capture

In a collaborative capture context, the literature review reports different ways to resolve this problem. 1) One possible solution is perform a synchronization with the help of a

⁵ <https://git-scm.com/>

⁶ <https://github.com/>

⁷ <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

⁸ Android Software Development Kit <http://developer.android.com/sdk/>

⁹ <http://simple.sourceforge.net/>

cloud server. The application performs the synchronization before capture a medium.

a) In some cases the device internal clock is synchronized with the server clock. So, all media captured are synchronized because the recorded time at meta-data information is synchronized with the server clock. b) In other cases, the application calculates the synchronization information based on the server time and device time. The synchronized time calculated is stored as meta-data information into the media captured as in the work reported by Leite *et al.* (2008), Kaheel *et al.* (2009), Viel *et al.* (2014). 2 The other solution is perform a synchronization the media elements (audio or video) using computer processing of fingerprints from the audio of the media elements. The application could use matching algorithms to detect matches between the different audio streams, as in the work by Kennedy and Naaman (2009), Haitsma and Kalker (2003), Guimarães, Cesar and Bulterman (2013).

6.1.2.1.1 Synchronization in MoVIA2

MoVIA2, the previous version of *CMoViA app*, was designed for a scenario of a small group (two or three users) capturing a real event but, the users do not have Technology for wireless local area networking (WiFi) or mobile network access. The devices — smartphone or table — of every user has a different time (they are not synchronized). We proposed an *explicit physical synchronization*. We needed to identify the same instant of time for every device even the time was not the same following the clock of the device. We decided to identify a *shake event*, which is short and fast movement of the device (i.e., a *shake movement*). We used the accelerometer of the device to identify the *shake event* and the corresponding *synchronization time*. When *MoVIA2* detects a *shake event*, it uses that time to calculate the time of a capture in function of the *synchronization time*.

Figure 20 shows the synchronization process of *MoVIA2*. At the beginning, we set up at the applications in each device to be ready for synchronization. We tap over the button “Shake synchronizer” in the screen. The button take a red color that means the applications is ready to synchronization (Figure 20(a)). After that, we take all the smartphones with our hands and perform a *shake event* (Figure 20(b)). In this very moment, *MoVIA2* stores the *synchronization time* and change the button “Shake synchronizer” to green color. It is a feedback for a OK synchronization (Figure 20(c)). Finally, at recording, every time that *MoVIA2* captures a new medium, the application calculates the interval time between the *synchronization time* and the recording time of the new medium.

As a result of having used the *interval time* related with the *synchronization time* (i.e. the shake movement), all the media elements captured with a synchronized device are synchronized. After we mix the session of every device with *MoVIA2*, we can confirm the synchronization was successful at visualization.

We identify some problems with this synchronization method:

Figure 20 – Explicit synchronization with *MoViA2* (previous version)

- The synchronization method is limited to the maximum number of devices that one person can hold at the same time;
- In cases of multiple smartphones and tables, holding devices of different sizes may difficult the *shake event*;
- After performing the *shake event*, the user has to verify that all devices was synchronized. If one of the devices does not recognize the *shake event*, the process has to be repeated;
- In some cases the synchronization presented a delay of 1 second.

6.1.2.1.2 Synchronization in CMoViA

CMoViA app uses the support of a server (*CMoViA API*) to perform the synchronization. The media elements captured for each participant of the collaborative capture session should be synchronized with the other participants.

For the synchronization, we calculate the *media server time diff* (ΔT_{msd}). The Equation 6.1 shows that the *media server time diff* is the time difference between the *media captured time* (T_{mc}) and the *server time* (T_s), where a) the *media server time diff* is the time interval since the collaborative session begun until the application notify that it started the capture of a medium; b) the *server time* is the instant of time when one participant created the collaborative session, it was stored at *CMoViA API* and c) the *media captured time* is the instant of time when *CMoViA app* start to capture a new

medium.

$$\Delta T_{msd} = T_{mc} - T_s \quad (6.1)$$

Having in mind that each device (*CMoViA app*) and the cloud server (*CMoViA API*) could have different times, different time zone and so on. We do not try to synchronized the clock of each device, we need to calculate which is the interval time in the server and also the interval time in the mobile device. Both times added, represent the *media server time diff*.

The Equation 6.2 shows that the *media server time diff* is composed by the *server joining time diff* (ΔT_{sjd}) plus the *media captured time diff* (ΔT_{mcd}), where a) the *server joining time diff* is the time interval since the *server time* until the *join time* (T_j), b) the *join time* is the instance of time when a new participant joins into the collaborative session stored in *CMoViA API* — even the time stored are different for the clock of both, the server and the mobile device, they mark the exact time instance — and *CMoViA app* using their own clock time, c) the *media captured time diff* is the time interval from *join time* until *media captured time*.

$$\begin{aligned} \Delta T_{msd} &= \Delta T_{sjd} + \Delta T_{mcd} \\ \Delta T_{sjd} &= T_j - T_s \\ \Delta T_{mcd} &= T_{mc} - T_j \end{aligned} \quad (6.2)$$

Figure 21 shows graphically the synchronization into *CMoViA API* and *CMoViA app*. We store the *media server time diff* (seconds) as metadata related with the media captured into the session document description. This synchronization process is making by HTTP requests between *CMoViA API* and *CMoViA app*.

Figure 21 – Representation of the synchronization information calculated for *CMoViA* when a new medium is captured, corresponding to Equation 6.2.

After exporting the recordings, *CI+WaC* mixes the session and creates a new one collaborative session. we can confirm the synchronization was successful at visualization. This synchronization method allow to perform an opportunistic collaborative capture, because the users can join and left the recording anytime they want.

6.1.2.2 Multimedia selection

The literature review reports (Section 3.1) that many researchers use the video as principal media element. The popularization of the video became more popular because nowadays, smartphones and tablets offer high definition camera and they are not so expensive (accessible for good part of the population). This technological feature give some problems: 1) users that likes to record in HD the photos and HD videos have a limited device storage of the device, 2) even the popularization of cloud storage servers, the users have to deal with high prices and low bandwidth mobile networks. Researchers started working with photos and audio as well as video in multimedia production, and explore scenarios where the photos and audio are as important as video; examples include the work by [Guimarães et al. \(2011\)](#), [Guimarães, Cesar and Bulterman \(2013\)](#), [Cunha, Uscamayta and Pimentel \(2016\)](#), [Cunha, Machado Neto and Pimentel \(2013\)](#), [Ojala et al. \(2014\)](#). .

CMoViA app included the multimedia capture feature with audio, photos, video and bookmarks, also the utilization of smartphone accessories improve and facilitate the capture of audio and bookmarks.

The amateur users have the chance to choose the media that they want to capture. Before, the users create a collaborative session or join into an existing collaborative session, the application shows a list of the different medium combination to record as shown in Figure 22(a). The user can choose between: only video, photo and audio, only photos, only audio, only bookmarks.

Many smartphones and tables have high quality cameras but depending of the context or storage limitations, many users could prefer record low quality video. The utilization of accessories, like Bluetooth headset, could be very useful in some context to improve the audio quality. The configuration about the quality recording used by the camera, utilization or not of a Bluetooth headset and time selection for periodical photos is configured in a configuration screen, as shown in Figure 22(b) shows, the configuration is for the device, so it will affect any recording made by the smartphone or tablet.

Even the bookmarks are not medium, following the *CI+WaC-IE model*, we consider it as a media element as the other media (audio, video, images). The bookmarks are represented as text annotations with the objective to mark an specific time as important. The selection of *bookmarks* was though for users that are only interested in capture personal relevant moments in the real event (multimedia authoring).

6.1.2.3 Context information

The ubiquitous computing highlight the importance of context information that can be captured from users in specific actions (Section 2.1). The mobile application provides some context information to users that want to be participants of a collaborative event

Figure 22 – Interfaces used for media selection and application configuration in *CMoViA* app

capture. Before users join into a collaborative session, the application presents an interface with context information (start time, end time, participant name, icons representing the media recorded, icon representing if the recording was ended or not) about each participant that is or was part of the capture session, as Figure 23(a) shows. Figure 23(b) shows that *CMoViA* app provides a picture preview about the session captured (only for video or picture recording). It shows up when the user taps over context information of an active participant in the list.

This context information gives us an idea about the recording session. The new participant will take a decision about which media to use to join the session, besides an idea about the position to locate himself in the event place with the objective to try to cover all aspects and profiles about the event captured (e.g. in a math class, a new participant realizes that the other participants are recording the class using low definition video and a Bluetooth headset for audio used by the professor, but the new participant believes that it is really important to have pictures of the board where the professor explains theorems, solves problems, etc., so he will choose only photos and locate in front of the board). The decision will be taken in function of personal interests, technological or storage device limitations (e.g. the low quality of my smartphone camera will not let me take good photos of the

(a) List of all participants in the collaborative capture event

(b) Photo preview of the user *diana.interm* that is recording photo and audio

Figure 23 – Interfaces used for context information in CMoViA app

math written in the board) and context information about the session.

6.1.2.4 Mobile accessories

The popularization of smartphones and tables also create a huge range of mobile accessories. The accessories could include any software or hardware that was not designed as part of the original mobile device, it works like an complement for an specific task. The market offers different accessories like cases (designed to attach to, support, or otherwise hold the mobile device), mass storage (SD cards, Wi-Fi SD), chargers and external batteries, adapters (Micro USB to HDMI cables, etc.) and wireless accessories. The wireless accessories make the most of wireless technology to connect the specific hardware with the mobile device. Bluetooth became the most used technology for mobile accessory providers. Nowadays the market offers headphones, earphones, headset, smart-watches, selfie sticks, remote shutters, keyboards, glass screen enlarger, music box, etc. Finally there are smart clothing and wearable accessories created by researchers for lifestyle monitoring solutions, as illustrated in the work by [Caon et al. \(2015\)](#), [Neto et al. \(2016\)](#).

Figure 24 – Collaborative capture scenerio with three smartphones where one of them uses a Bluetooth headset.

In the capture sessions in real scenarios, we identify three main problems related with: 1) audio quality, 2) discomfort holding the mobile device in long period times and 3) loose attention in a bookmark capture.

1. Our first experience in a real scenario was in a graduate class *Math topics in data analyzing I and II* offered for the ICMC-USP. The main problem in this capture was the low sound of the audio of the professor because the mobile devices were located far from the professor in specific places in the class room for the objective of capture the all board . We though in a little force solution, use an specific mobile device for audio recording close or into professor clothes, but this solution imply wasting and mobile device just to record audio and possible discomfort for the professor. Then we though in a microphone solution using a Bluetooth headset. The low cost and wearable features of the accessory make us develop a feature of the possibility of use a Bluetooth headset as microphone instead of use the mobile device microphone. If a participant of a collaborative capture wanted to use their Bluetooth headset, he have to connect the accessory and configure the application in the configuration screen (Figure 22(b)). Figure 24 shows an collaborative capture scenery where one mobile device uses a Bluetooth headset.
2. In our local tests, we observed the discomfort faced by users holding a mobile device for a long period of time. The tripod accessory was the logic solution, even though

at first we created handmade supports recycling plastic bottles and Styrofoam. After that we acquired tripods and remote shutters.

3. The remote shutter communicates with the mobile device through Bluetooth. It was created for taking remote photos, the user presses a real button in the remote shutter to take the photo. The remote shutter is composed by two buttons for take photos, one button for Android devices and the other button for iOS devices. We decided to adapt this feature and reuse the two buttons, one button to stop the capture and the other to make bookmarks. The remote shutter let users pay attention at the presentation and also the possibility of make bookmarks.

6.1.3 CI+WaC web tool integration

I+WaC-Editor is a Web-based multimedia authoring tool that focuses on extension of multimedia documents. Martins (2014) developed it in the context of his Doctoral Thesis. *I+WaC-Editor* was originally conceived to be a playback and enrichment tool focus in a principal video media. Cunha (2014) includes a text annotations export feature into *MoViA* tool with the generation of a JSON object. Finally this document is exported (for example, by e-mail and Bluetooth. Even, the export process exists, it was a semi-automatic process because the video and the text annotations were sent to the webmaster of *I+WaC-Editor* and make the upload to the tool. This process had to be improved by the propose of an automatic exportation process.

Inspired in collaborative communities, where different users share their videos and receive in change a video mixer. We propose complete the automatic exportation process in the scenery where the participants of *CMoViA app* capture multimedia element and export the media and the text annotations (also the bookmarks) to *I+WaC-Editor* that let everyone playback the collaborative session. The users can interchange the different media following their own needs.

In this work, Martins (2014) extended *I+WaC-Editor* providing a REST web service for the exportation process, named Collaborative I+WaC (CI+WaC). It uses a JSON document with the session description. The REST web service is composed by four simultaneous operations:

1. The *open* operation starts a new ingestion session. The client application obtains a valid *Google+ Sign-in* access token and send it as a parameter of the operation. After that, the server returns an new upload session token that will be required in each next steps of the process.
2. The *descriptor* operation uploads the JSON descriptor (structure in Source Code 3) of the ingestion session.
3. The *mediaElement* operation uploads a media element. The application client —

CMoViA app is a client, not necessarily the only client — has to perform this operation for each media element of the capture.

4. The *close* operation, notify the finalization of the ingestion session. The server perform the union of sessions and make it available to other users. The union of sessions does not perform any synchronization — the client applications are in charge of perform the synchronization. The server just mix the document descriptor as one and creates a timeline for each ingestion session with their annotation text. The session union is possible for the *session_token* element that is created by *CMoViA* API and it is share between the participants.

The playback interface of *CI+ WaC* also was extended. Figure 25 shows the original interface used for individual session. Figure 26 shows the new interface oriented into collaborative capture, where (1) is the part that shows photos of an specific user, (2) is the part that play video of an specific user, (3) is the part with three combo box for choose the media for an specific user, (4) playback controls and (5) shows one timeline with text annotations or bookmarks of each participant in the collaborative capture.

Figure 25 – *CI+ WaC* reproducing a video and showing the different text annotations.

6.2 Experiments with users

Every time that we finished a group of requirements, we tested the application in real scenarios for educational events. We simulate collaborative recordings using smartphones, tablets and mobile accessories (Bluetooth headsets, tripods and remote shutters).

Figure 26 – *CI+WaC* reproducing a collaborative capture of 3 participants: the first captured video, the second captured photos and the third captured photos and audio using a Bluetooth headset.

6.2.1 Previous recordings

MoViA2 — previews version of *CMoViA app* — lets students can perform collaborative recording with the physical explicit synchronization and also the possibility of photo and audio recording beyond video recording. We make four main collaborative recordings using *MoViA2* with the objectives of testing the application and identifying the need for improvements.

- We captured the graduate class “*Math topics in data analyzing I and II*” offered by ICMC-USP, the second semester of 2014. The students of that course had access to the collaborative recording in *I+WaC-Editor*, as review material. This course was used for analysis and compares the functionalities of the *MoViA2* and *AndrEya*. The mobile devices used for the recording were a tablet with *AndrEya* (audio and periodic photo recording), a smartphone (audio and periodic photo recording) and a tablet (only periodic photo recording) with *MoViA2*. The main problem detected in this recording was the low sounds that impede the full understanding of the class review, solved after with the utilization of Bluetooth headsets.
- We captured the defenses by students presenting their undergraduate the “*Final Project*” for their courses in the ICMC-USP in the second semester of 2014. The collaborative recording was also available on *I+WaC-Editor*. Two mobile devices and a Bluetooth headset recorded six defenses with *MoViA2*, one smartphone recording periodic photos plus audio with the Bluetooth headset (attached to the speaker shirt

like a microphone) and one tablet recording only photos. After playing back this recordings, we realized the importance of having synchronized video recording that was not implemented yet and confirmed that the Bluetooth headset led to good results in the quality of audio.

- In the summer of 2015, we captured two summer school courses, the (“*Summer School on Computers in Education*” and the “*ACM ICPC programming summer school*”), which were made available on *I+WaC-Editor*. For the recording of both courses, we used one smartphone recording periodic photos plus audio with the Bluetooth headset, one tablet recording only photos and another tablet recording video in low quality. The main problems detected were: 1) it was not impossible to include a new participant to the collaborative recording once the recording was initiated; 2) the synchronization was limited to three devices; and 3) the capturing of a bookmark annotation could distract the attention of the student. Figure 27 shows an example of the visualization in the *I+WaC-Editor*.

Figure 27 – “*Summer School on Computers in Education*” visualization in *I+WaC-Editor*. View of the first mobile device used to capture the session.

CMoViA app described in Section 5.2.3 implements their features based on the problems detected in whole work. We employed *CMoViA* in many different scenarios like classes, dissertation and theses defense practices, and also in the study case discussed next.

Figure 28 – “Summer School on Computers in Education” visualization in I+WaC-Editor. View of the second mobile device used to capture the session: the image on the left was captured 10 seconds after the image captured by the first device shown in Figure 27.

6.2.2 Case Study

The case study was divided in two parts. The first part was a usability evaluation where the participants test *CMoViA app* alone. The second part was testing of *CMoViA app* in two different classes (real scenario). We recruited five graduated student volunteers in age range from 25 to 35 years old (they did not receive payment for participation). All participants were part of the whole case study except one of them that could not participate in the last collaborative test. We provide mobile devices and accessories for each participants. We have created a fictitious *google account* for each mobile device, so we preserve the anonymity of the participants.

6.2.2.1 Usability Evaluation

The usability evaluation pursues the objectives: *a) introduce CMoViA app*, *b) detect usability problems* and that *c) the participants get familiar with the application* in real collaborative situation. With it, the participants had gotten expertise using the app, which it was necessary for the next real scenario tests. We designed a set of tasks with the objective of simulate a collaborative recording scenario. The test required two smartphones, handmade supports and a laptop. The laptop was used to play a lecture video that simulates an educational event. Each participant performed the test individually in around 20 minutes. We used the *Think Aloud protocol* to collect usage data and a post-interview to collect opinions about the *CMoViA app*. We also recorded in video the usability test and the interviews in audio.

6.2.2.2 Tests in real scenarios

The tests involved two collaborative recordings of educational events. The objective was to observe and analyse the collaborative recording of a real class using multiple devices distributed in a classroom.

- The first test in a real scenario was performed in the “*Special Topics*” course, where all volunteers have already enrolled. The volunteers were free to choose their physical location into the classroom and also the media that they will used for the recording. However, we asked to the volunteers, use the following criteria for the media choosing: *a)* the medium chosen by the other participants in the capture, *b)* the quality of the medium chosen by the other participants and *c)* a personal choice. We have observed the volunteers actions during the collaborative capture. At the end of the recording, we request the participant answer some questions about their own experience in the text.
- The second test in a real scenario was performed in an offering of the “*Human-Computer Interaction*” course, where all volunteers were already enrolled. One of the volunteers did not participate on this experiment. The volunteers used the same criteria, in the medium chosen, of the first test in a real scenario. At the end of the experiment, the volunteers exported their recordings from *CMoViA* to *I+WaC-Editor*. After that, we gave access to playback the collaborative recording and provided a questionnaire about their experience. We used five points scale in the *Likert questions* to ask about their opinion about the capture. The five questions were 1) “I thought it was easy to perform the recording”, 2) “I thought it was easy to find out what the other participants were recording.”, 3) “I’m satisfied with the result of the collaborative recording.”, 4) “I’m able to visualize the slides with quality.” and 5) “I am able to follow the speaker.”. The questionnaire also contained open ended questions so participants could explain their answers.

6.2.2.3 Results

The usability evaluation exhibited that the *CMoViA* app was easy to learn, easy to use and also it has no major usability problems. It was because, the volunteers did not have problem to complete the set of tasks proposed. We identified minor usability errors: 1) one screen did not follow the correct flow, 2) the volunteers did not understand the low quality icon represented by “LQ”, 3) the volunteers did not understand the “camera preview” function and 4) one student said that he does not like to record presentations because he had gotten distracted in the recording and he could not pay the attention wanted. The student also said that he would like to only add temporal bookmarks and someone else record.

Figure 29 – Likert responses average for each question (CUNHA; USCAMAYTA; PIMENTEL, 2016)

Nonetheless, the real collaboration test scenario revealed a lot of issues that we could not identify in the usability tests. The first test in a real scenario exhibited some problems: 1) the volunteers were worry about losing or disturbing the presentation so they were in hurry to start recording, 2) the volunteers were unsure between “create” or “join” the event, 3) even they know about the context information screen, one of them asked aloud which media the others had chosen because, he thought that it was the “faster way” (all participants already know each other), 4) two participants choose the media without considering the context information, 5) on volunteer notices that another volunteer did not share a camera preview. We realized that most volunteers were more concerned in their individual capture and did not wanted to analyze the context information to decide which medium they will use.

The second test in a real scenario shows more positive results. The volunteers said that they were very interested in reviewing the material recorded because some of them were exponents and they received feedback form the professor and the audience. Because of that, they were more careful int the medium chosen for recording.

Figure 29 shows the responses averages for the *Likert questions*. In the questionnaire, the responses about the capturing tasks were fairly positive (4.4 and 3.8). It means that although the recording had many drawbacks, the participants thought that CMoViA facilitated the conduct of the collaborative recording. Even, the interfaces for visualization were very simple, the responses about the visualization in *I+WaC-Editor* were around neutral and negative (2.8, 2.6 and 3.2). The participants reported that they would like to manipulate the zoom for the video and images. They also reported that *I+WaC-Editor* could provide better options to control multiple audio sources. However, as we did not explore the visualization stage further, that result was expected.

6.3 Final remarks

This chapter presented the CMoViA tool, its most important functionalities and the tests with real users. This chapter also details how the CMoViA tool corresponds to the *CI+WAC-IE* model, presented in Chapter 4, and its own software design, detailed in Chapter 5). In particular, it was dicussed the importance of defining an explicit author in

a collaborative multimedia capture to generate an interactive and extensible multimedia document.

The tests in real scenarios highlighted the needs and problems that a collaborative recording presents. It help us to review the tool and show the features detailed in this chapter. Some interesting functionalities of *CMoViA* are detailed, including *a)* explicit physical synchronization method, *b)* choosing multiple media for recording, *c)* uses of context information to support decisions for recording, and *d)* uses of mobile accessories. This functionalities support a collaborative multimedia capture and help us to get better captured media.

The *CI+WaC* tool, extended by [Martins \(2014\)](#) based on his previous work, provides an interesting alternative to visualization of collaborative captures. It give the visualization control to the users who can choose which media to use according to their needs.

CONCLUSION

In the first chapter of this dissertation we introduced the context and associated gap we focus in our investigation, which led us to state the following research question:

How to support ubiquitous collaborative multimedia production using user-generated content, captured via mobile devices, so as to result in interactive and extensible multimedia documents?

This question led us to define the following objective for our research:

The aim of this work is to propose an approach that supports the ubiquitous collaborative production of interactive and extensible multimedia documents by exploiting user-generated content captured via mobile devices.

In the chapters that followed, we reviewed the literature which allowed us to extend previous work with respect to the *I+WaC-IE* model and associated *Interactors* as well as the *MoViA* mobile tool. As a result, we proposed the *CI+WaC-IE* model and implemented the *CMoViA* tool. Considering these results, we can present the following answer to our original research question:

*Ubiquitous collaborative multimedia production can be carried out by users who capture and annotate multiple media using the *CMoViA* mobile application and export the user-generated content to the *CI+WaC*, which allows them to edit the user-generated content in the form of interactive and extensible multimedia documents.*

In this chapter we review our contribution and its limitations, and point to future work.

7.1 Contribution and discussion

Towards achieving the results we report in this dissertation, our work contributes with previous research efforts as following:

- **CI+WaC-IE extended model**

The original *I+WaC-IE model*, proposed by Martins (2014), supported the anonymous collaborative enrichment of interactive multimedia documents via end-user annotations. In the context of the original media to be annotated, the model assumed the existence of a multimedia document containing media elements captured by a single unknown person because the identification of the author was not necessary. In the context of multimedia authoring, in our work we identify the agent (human or object) responsible for the interaction event since we cannot assume that the agent is also the author of the annotation. So, we proposed the *CI+WaC-IE model* as an extension of the original model: the main difference is that a new *Author* concept was added to the original model. The new concept allows identifying not only the author responsible for the original media elements and the original media document in a session with collaborative capture, but also the author responsible for annotations in the context of multimedia authoring.

- **New *Interactors* focused on *Author* concept**

The inclusion of the new *Author* concept in *CI+WaC-IE model* allows the explicit identification of the author responsible for the creation of annotations, media elements and multimedia document. It also allowed us to define a new set of *Interactors* related with the *Author* concept. This can enhance both automatic and semi-automatic authoring tasks.

- **Multimedia collaborative document that follows the *CI+WaC-IE model***

We propose a XML schema for a multimedia document that supports collaborative capture following the *CI+WaC-IE model*. It is called *session document description*. This document is used by the *CMoViA app* as data model in both individual and collaborative scenarios of capture.

- **Propose method for physical movement-based synchronization**

We proposed an alternative method to using wireless networks to connect devices: the method can be employed in scenarios with lack wireless infrastructure. The physical movement-based synchronization using the accelerometer sensor from mobile devices to perform a “shake event” results in the devices being synchronized at the time of the movement.

- **Promote the utilization of mobile accessories to facilitate or improve the capture and authoring tasks**

We integrated a remote shutter and a Bluetooth head set in the capture features of the *CMoVia* app. Even though mobile devices offer a diverse group of sensors, the use of low cost mobile accessories can help obtaining better results in the multimedia production process. For example, a remote shutter can facilitate an authoring task and a Bluetooth head set can improve the quality of captured audio. It is important to analyze the different features that the mobile accessories offers and adapt or fit in a specific task of multimedia production process by the help of software development.

- **CMoVia tool**

Towards producing a proof-of-concept prototype supporting the *CI+WaC-IE* model with the extensions we propose, we implemented the *CMoVia* tool by extending the previous *MoViA* app. The extension supports opportunistic collaborative multimedia capture and visualization of the collaborative multimedia document generated in the mobile devices and also in the web. This extension involved:

- **CMoVia API.** The *CMoVia* API implementation is a back-end application used to support the synchronization and manage the context information in a collaborative capture following the *CI+WaC-IE* model.
- **CMoVia app.** The *CMoVia* app is a mobile application used to support collaborative capture following the *CI+WaC-IE* model. It also provides a visualization mode in the mobile device of both individual or collaborative sessions.

The analysis of the test with real users and our own test recordings demonstrate that some students want to have a record of their classes a material for review, but some are not excited about taking part in collaborative capture or authoring. The main reason offered by the students is that participating in such recording could be distracting and they prefer focusing on the class. In this work, we tried to facilitate the capture process and also give solutions that support an ubiquitous capture with the objective of reducing the possible distraction and provide a useful consulting material.

7.2 Limitations

Even though collaborative capture can be applied to many kinds of events, we delimited our work to educational events because we had more opportunities to record events of this type.

The explicit physical synchronization has a limitation on the quantity of smartphones that a person can hold in his hands to make the shake synchronization.

This work presents technological limitations for the Android SDK version used for development, like the restriction to use the camera driver by other thread while it is recording. This restriction did not let us take real time photo preview, so we solved it by taking the photo preview before start the recording. It is possible that in future versions of the technology this will not be a problem, or another technology can be used.

7.3 Future Works

The work reported in this dissertation can be extended so as to offer further support to the *Capture&Access multimedia production process*, including:

- **Pre-production and capture**

There is an opportunity to analyze the nature of the other kinds of event like sport events or musical events to propose changes or specifications oriented to the particular context of the event. Mobile device offers different types of sensors such as accelerometer, gyroscope, magnetometer, GPS and so on, that an application of collaborative capture can store and add value to the multimedia document. The use of any contextual information in interaction events depends of a specific scenario. Also, in particular capture scenarios in which the use of specific sensors is demanded, the adoption of mobile accessories such as smart clothing and wearable accessories could support or improve the multimedia capture.

In the HCI context, there is an opportunity to propose new natural interfaces for mobile devices to create a real time bidirectional communication channel among the users collaborating in the capture of media. The objective is to capture more details about the event or even to define a user with the role of director that orchestrates the capture process. It is interesting to note that any communication can enrich the multimedia document because it could also be stored as annotations of user-user interaction events.

- **Post-production and extension**

The multimedia elements captured could be used for the generation of new media elements like 3D reconstruction, and the possibility to extend the authoring to include 3D annotations. Also, there is a demand for research leading to the production of automatic multimedia mixers (equivalent to video mixers) that would compute a summary of the event considering all capture multimedia elements and user annotations.

- **Access and publishing**

In the HCI context, there is a demand for proposing new natural interfaces for mobile devices to facilitate multimedia visualization in cases in which many different recordings and media elements were captured.

There is also a demand for analyzing publishing alternatives such as live streaming technologies.

7.4 Final remark

[Shenoy \(2013\)](#) observed that the future of research in multimedia systems will be impacted new trends, such as mobile and social, which are changing how users create, share, and view content. [Shenoy \(2013\)](#) also observes that these advances are associated with challenges that include mobile streaming, privacy, and new modes of interaction.

Inspired on researchers such as [Weiser \(1991\)](#), [Truong, Abowd and Brotherton \(1999\)](#), [Abowd, Mynatt and Rodden \(2002\)](#), [Goularte *et al.* \(2004\)](#), [Cattelan *et al.* \(2008a\)](#), [Guimarães *et al.* \(2011\)](#), [Bulterman, Cesar and Guimarães \(2013\)](#), [Viel *et al.* \(2013b\)](#), [Martins \(2014\)](#), [Zhang, Zhang and Zimmermann \(2015\)](#) and [Cunha, Uscamayta and Pimentel \(2016\)](#), our work focuses in exploiting new modes of interaction in mobile devices towards changing the way users create, share and view interactive multimedia content.

BIBLIOGRAPHY

1BEYOND.COM. *MIT Captures Lectures with Multiple Cameras and No Staff*. 2015. <http://1beyond.com/news/mit-captures-lectures-with-multiple-cameras-and-no-staff>. Citation on page 44.

ABOWD, G.; PIMENTEL, M. da G.; KERIMBAEV, B.; ISHIGURO, Y.; GUZDIAL, M. Anchoring discussions in lecture: An approach to collaboratively extending classroom digital media. In: INTERNATIONAL SOCIETY OF THE LEARNING SCIENCES. **Proceedings of the 1999 Conference on Computer support for collaborative learning**. [S.l.], 1999. p. 1. Citation on page 37.

ABOWD, G. D.; ATKESON, C. G.; FEINSTEIN, A.; HMELO, C.; KOOPER, R.; LONG, S.; SAWHNEY, N.; TANI, M. Teaching and learning as multimedia authoring: The classroom 2000 project. In: **Proceedings of the Fourth ACM International Conference on Multimedia**. New York, NY, USA: ACM, 1996. (MULTIMEDIA '96), p. 187–198. ISBN 0-89791-871-1. Available: <http://doi.acm.org/10.1145/244130.244191>. Citation on page 44.

ABOWD, G. D.; MYNATT, E. D. Charting past, present, and future research in ubiquitous computing. **ACM Trans. Comput.-Hum. Interact.**, ACM, New York, NY, USA, v. 7, n. 1, p. 29–58, Mar. 2000. ISSN 1073-0516. Available: <http://doi.acm.org/10.1145/344949.344988>. Citations on pages 31 and 33.

ABOWD, G. D.; MYNATT, E. D.; RODDEN, T. The human experience [of ubiquitous computing]. **IEEE Pervasive Computing**, v. 1, n. 1, p. 48–57, 2002. Citation on page 107.

ACKERMAN, M. S. The intellectual challenge of cscw: The gap between social requirements and technical feasibility. **Hum.-Comput. Interact.**, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, v. 15, n. 2, p. 179–203, Sep. 2000. ISSN 0737-0024. Available: http://dx.doi.org/10.1207/S15327051HCI1523_5. Citation on page 33.

BIANCHI, M. Automatic video production of lectures using an intelligent and aware environment. In: **Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia**. New York, NY, USA: ACM, 2004. (MUM '04), p. 117–123. ISBN 1-58113-981-0. Available: <http://doi.acm.org/10.1145/1052380.1052397>. Citation on page 44.

BIANCHI, M. H. AutoAuditorium: a Fully Automatic, Multi-Camera System to Tele-vise Auditorium Presentations. In: **Joint DARPA/NIST Smart Spaces Technology Workshop**. [S.l.: s.n.], 1998. p. <http://www.autoauditorium.com/nist/autoaud.html>. Citation on page 44.

BIANCO, S.; CIOCCA, G. User preferences modeling and learning for pleasing photo collage generation. **ACM Trans. Multimedia Comput. Commun. Appl.**, ACM, New York, NY, USA, v. 12, n. 1, p. 6:1–6:23, Aug. 2015. ISSN 1551-6857. Available: <http://doi.acm.org/10.1145/2801126>. Citation on page 23.

BROTHERTON, J. A.; ABOARD, G. D. Lessons learned from eclass: Assessing automated capture and access in the classroom. **ACM Trans. Comput.-Hum. Interact.**, ACM, New York, NY, USA, v. 11, n. 2, p. 121–155, Jun. 2004. ISSN 1073-0516. Available: <<http://doi.acm.org.ez67.periodicos.capes.gov.br/10.1145/1005361.1005362>>. Citations on pages 24 and 37.

BULTERMAN, D. C. A.; CESAR, P.; GUIMARÃES, R. L. Socially-aware multimedia authoring: Past, present, and future. **ACM Trans. Multimedia Comput. Commun. Appl.**, ACM, v. 9, n. 1s, p. 35:1–35:23, Oct. 2013. Citations on pages 25, 46, and 107.

CANESSA, E.; FONDA, C.; TENZE, L.; ZENNARO, M. EyApp & AndrEyA - Free apps for the automated recording of lessons by students. **International Journal of Emerging Technologies in Learning**, v. 9, p. 31–34, 2014. ISSN 18688799. Citations on pages 25, 44, and 47.

CANESSA, E.; FONDA, C.; ZENNARO, M. Eya system: Automated audio-video-slide recordings. In: . [S.l.: s.n.], 2007. (Conference ICL2007). Citations on pages 24, 25, 44, and 47.

CAON, M.; CARRINO, S.; MUGELLINI, E.; LANG, A. R.; ATKINSON, S.; MAZZOLA, M.; ANDREONI, G. Smart garments and accessories for healthy lifestyles. In: **Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers**. New York, NY, USA: ACM, 2015. (UbiComp/ISWC'15 Adjunct), p. 623–626. ISBN 978-1-4503-3575-1. Available: <<http://doi.acm.org/10.1145/2800835.2809434>>. Citation on page 92.

CATTELAN, R. G.; TEIXEIRA, C.; GOULARTE, R.; PIMENTEL, M. D. G. C. Watch-and-comment as a paradigm toward ubiquitous interactive video editing. **ACM Trans. Multimedia Comput. Commun. Appl.**, ACM, v. 4, n. 4, p. 28:1–28:24, Nov. 2008. Citations on pages 27, 49, 55, 57, 60, and 107.

CATTELAN, R. G.; TEIXEIRA, C.; RIBAS, H.; MUNSON, E.; PIMENTEL, M. Inkteractors: Interacting with digital ink. In: **Proceedings of the 2008 ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2008. (SAC '08), p. 1246–1251. ISBN 978-1-59593-753-7. Available: <<http://doi.acm.org/10.1145/1363686.1363973>>. Citation on page 47.

CESAR, P.; BULTERMAN, D. C.; GEERTS, D.; JANSEN, J.; KNOCHÉ, H.; SEAGER, W. Enhancing social sharing of videos: Fragment, annotate, enrich, and share. In: **Proceedings of the 16th ACM International Conference on Multimedia**. New York, NY, USA: ACM, 2008. (MM '08), p. 11–20. ISBN 978-1-60558-303-7. Available: <<http://doi.acm.org/10.1145/1459359.1459362>>. Citation on page 35.

CUNHA, B. C. **Captura da interação para autoria e compartilhamento multimídia em dispositivos móveis**. Master's Thesis (Master's Thesis) — INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO - USP, 2014. Citations on pages 47, 50, 69, 86, and 94.

CUNHA, B. C.; MACHADO NETO, O. J.; PIMENTEL, M. d. G. Movia: A mobile video annotation tool. In: **Proc. Symposium on Document Engineering**. [S.l.]: ACM, 2013. (DocEng '13), p. 219–222. Citations on pages 11, 24, 27, 29, 39, 51, and 90.

- CUNHA, B. C.; USCAMAYTA, A. O. M.; PIMENTEL, M. d. G. C. Opportunistic recording of live experiences using multiple mobile devices. In: **Proceedings of the 22Nd Brazilian Symposium on Multimedia and the Web**. New York, NY, USA: ACM, 2016. (Webmedia '16), p. 99–102. ISBN 978-1-4503-4512-5. Available: <http://doi.acm.org/10.1145/2976796.2988164>. Citations on pages 12, 24, 90, 100, and 107.
- ELLIS, C. A.; GIBBS, S. J.; REIN, G. Groupware: Some issues and experiences. **Commun. ACM**, ACM, New York, NY, USA, v. 34, n. 1, p. 39–58, Jan. 1991. ISSN 0001-0782. Available: <http://doi.acm.org/10.1145/99977.99987>. Citation on page 34.
- ENGSTRÖM, A.; PERRY, M.; JUHLIN, O. Amateur vision and recreational orientation:: Creating live video together. In: **Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work**. New York, NY, USA: ACM, 2012. (CSCW '12), p. 651–660. ISBN 978-1-4503-1086-4. Available: <http://doi.acm.org/10.1145/2145204.2145304>. Citations on pages 45 and 47.
- ENGSTRÖM, A.; ZORIC, G.; JUHLIN, O.; TOUSSI, R. The mobile vision mixer: A mobile network based live video broadcasting system in your mobile phone. In: **Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia**. New York, NY, USA: ACM, 2012. (MUM '12), p. 18:1–18:4. ISBN 978-1-4503-1815-0. Available: <http://doi.acm.org/10.1145/2406367.2406390>. Citations on pages 45 and 47.
- GEROSA, M. A.; PIMENTEL, M.; FUKS, H.; LUCENA, C. J. P. de. Development of groupware based on the 3c collaboration model and component technology. In: **Proceedings of the 12th International Conference on Groupware: Design, Implementation, and Use**. Berlin, Heidelberg: Springer-Verlag, 2006. (CRIWG'06), p. 302–309. ISBN 3-540-39591-1, 978-3-540-39591-1. Available: http://dx.doi.org/10.1007/11853862_24. Citation on page 34.
- GEYER, W.; RICHTER, H.; ABOWD, G. D. Towards a smarter meeting record–capture and access of meetings revisited. **Multimedia Tools Appl.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 27, n. 3, p. 393–410, Dec. 2005. ISSN 1380-7501. Available: <http://dx.doi.org/10.1007/s11042-005-3815-0>. Citation on page 40.
- GOULARTE, R.; CAMACHO-GUERRERO, J. A.; INACIO, V. R.; CATTELAN, R. G.; PIMENTEL, M. G. C. M4note: a multimodal tool for multimedia annotations. In: **Web-Media and LA-Web, 2004. Proceedings**. [S.l.: s.n.], 2004. p. 142–149. Citations on pages 38 and 107.
- GRUDIN, J. Computer-supported cooperative work: History and focus. **Computer**, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 27, n. 5, p. 19–26, May 1994. ISSN 0018-9162. Available: <http://dx.doi.org/10.1109/2.291294>. Citation on page 33.
- GUIMARÃES, R. L.; CESAR, P.; BULTERMAN, D. Personalized presentations from community assets. In: **Proc. Brazilian Symposium on Multimedia and the Web**. [S.l.]: ACM, 2013. (WebMedia '13), p. 257–264. Citations on pages 24, 87, and 90.
- GUIMARÃES, R. L.; CESAR, P.; BULTERMAN, D. C. Creating and sharing personalized time-based annotations of videos on the web. In: **Proceedings of the 10th ACM Symposium on Document Engineering**. New York, NY, USA: ACM, 2010. (DocEng '10), p. 27–36. ISBN 978-1-4503-0231-9. Available: <http://doi.acm.org/10.1145/1860559.1860567>. Citation on page 38.

_____. "let me comment on your video": Supporting personalized end-user comments within third-party online videos. In: **Proceedings of the 18th Brazilian Symposium on Multimedia and the Web**. New York, NY, USA: ACM, 2012. (WebMedia '12), p. 253–260. ISBN 978-1-4503-1706-1. Available: <http://doi.acm.org/10.1145/2382636.2382690>. Citation on page 38.

GUIMARÃES, R. L.; CESAR, P.; BULTERMAN, D. C.; ZSOMBORI, V.; KEGEL, I. Creating personalized memories from social events: Community-based support for multi-camera recordings of school concerts. In: **Proc. ACM International Conference on Multimedia**. [S.l.]: ACM, 2011. (MM '11), p. 303–312. Citations on pages 24, 46, 47, 90, and 107.

HAITSMA, J.; KALKER, T. A highly robust audio fingerprinting system with an efficient search strategy. **Journal of New Music Research**, v. 32, n. 2, p. 211–221, 2003. Available: <http://www.tandfonline.com/doi/abs/10.1076/jnmr.32.2.211.16746>. Citation on page 87.

HARDMAN, L.; OBRENOVIĆ, v.; NACK, F.; KERHERVÉ, B.; PIERSOL, K. Canonical processes of semantically annotated media production. **Multimedia Systems**, v. 14, n. 6, p. 327–340, 2008. Citation on page 35.

HAYES, G. R.; PATEL, S. N.; TRUONG, K. N.; IACHELLO, G.; KIENTZ, J. A.; FARMER, R.; ABOWD, G. D. The personal audio loop: Designing a ubiquitous audio-based memory aid. In: SPRINGER. **International Conference on Mobile Human-Computer Interaction**. 2004. p. 168–179. ISBN 978-3-540-28637-0. Available: http://dx.doi.org/10.1007/978-3-540-28637-0_15. Citation on page 25.

JUHLIN, O.; ZORIC, G.; ENGSTRÖM, A.; REPONEN, E. Video interaction: A research agenda. **Personal Ubiquitous Comput.**, Springer-Verlag, v. 18, n. 3, p. 685–692, Mar. 2014. Citations on pages 24 and 26.

KAHEEL, A.; EL-SABAN, M.; REFAAT, M.; EZZ, M. Mobicast: A system for collaborative event casting using mobile phones. In: **Proceedings of the 8th International Conference on Mobile and Ubiquitous Multimedia**. New York, NY, USA: ACM, 2009. (MUM '09), p. 7:1–7:8. ISBN 978-1-60558-846-9. Available: <http://doi.acm.org/10.1145/1658550.1658557>. Citations on pages 45, 47, and 87.

KAY, M.; CHOE, E. K.; SHEPHERD, J.; GREENSTEIN, B.; WATSON, N.; CONSOLVO, S.; KIENTZ, J. A. Lullaby: A capture & access system for understanding the sleep environment. In: **Proceedings of the 2012 ACM Conference on Ubiquitous Computing**. New York, NY, USA: ACM, 2012. (UbiComp '12), p. 226–234. ISBN 978-1-4503-1224-0. Available: <http://doi.acm.org/10.1145/2370216.2370253>. Citation on page 24.

KENNEDY, L.; NAAMAN, M. Less talk, more rock: Automated organization of community-contributed collections of concert videos. In: **Proceedings of the 18th International Conference on World Wide Web**. New York, NY, USA: ACM, 2009. (WWW '09), p. 311–320. ISBN 978-1-60558-487-4. Available: <http://doi.acm.org/10.1145/1526709.1526752>. Citations on pages 46, 47, and 87.

KINDEM, R. B. M. G. Introduction to media production: The path to digital media production. In: . [S.l.]: Focal Press; 4 edition (January 21, 2009), 2009. chap. 3, p. 73–94. Citation on page 35.

KRUMM, J.; DAVIES, N.; NARAYANASWAMI, C. User-generated content. **IEEE Pervasive Computing**, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 7, n. 4, p. 10–11, Oct. 2008. ISSN 1536-1268. Available: <<http://dx.doi.org/10.1109/MPRV.2008.85>>. Citations on pages 23 and 35.

KUKSENOK, K.; BROOKS, M.; MANKOFF, J. Accessible online content creation by end users. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2013. (CHI '13), p. 59–68. ISBN 978-1-4503-1899-0. Available: <<http://doi.acm.org/10.1145/2470654.2470664>>. Citation on page 35.

LEITE, R. W.; CANESSA, E.; ZENNARO, M.; FONDA, C. Tecnologia eya : uma ferramenta para produção e difusão automatizada de aulas digitais na web. **RENOTE : revista novas tecnologias na educação [recurso eletrônico]**. Porto Alegre, RS. Vol. 6, n. 2 (Dez. 2008), 10 p., 2008. Citation on page 87.

MARTINS, D.; PIMENTEL, M. da G. C. End-user ubiquitous multimedia production: Process and case studies. In: **Ubi-Media Computing (U-Media), 2011 4th International Conference on**. [S.l.: s.n.], 2011. p. 197–202. Citations on pages 11, 36, and 38.

MARTINS, D. S. **Models and operators for extension of active multimedia documents via annotations**. Phd Thesis (PhD Thesis) — INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO - USP, 2014. Citations on pages 11, 19, 27, 28, 29, 38, 47, 48, 49, 50, 55, 56, 57, 58, 59, 61, 69, 70, 94, 101, 104, 107, and 129.

MARTINS, D. S.; PIMENTEL, M. d. G. C. Activetimesheets: Extending web-based multimedia documents with dynamic modification and reuse features. In: **Proc. ACM Symposium on Document Engineering**. [S.l.]: ACM, 2014. (DocEng '14), p. 3–12. Citations on pages 11, 49, and 50.

MARTINS, D. S.; VEGA-OLIVEROS, D. A.; PIMENTEL, M. da G. C. Automatic authoring of interactive multimedia documents via media-oriented operators. **SIGAPP Appl. Comput. Rev.**, ACM, v. 11, n. 4, p. 26–37, Dec. 2011. Citations on pages 40, 48, and 129.

MINNEMAN, S.; HARRISON, S.; JANSSEN, B.; KURTENBACH, G.; MORAN, T.; SMITH, I.; MELLE, B. van. A confederation of tools for capturing and accessing collaborative activity. In: **Proceedings of the Third ACM International Conference on Multimedia**. New York, NY, USA: ACM, 1995. (MULTIMEDIA '95), p. 523–534. ISBN 0-89791-751-0. Available: <<http://doi.acm.org/10.1145/217279.215316>>. Citation on page 35.

MUKHOPADHYAY, S.; SMITH, B. Passive capture and structuring of lectures. In: **Proceedings of the Seventh ACM International Conference on Multimedia (Part 1)**. New York, NY, USA: ACM, 1999. (MULTIMEDIA '99), p. 477–487. ISBN 1-58113-151-8. Available: <<http://doi.acm.org/10.1145/319463.319690>>. Citation on page 44.

MÜLLER, R.; OTTMANN, T. The “authoring on the fly” system for automated recording and replay of (tele) presentations. **Multimedia Systems**, Springer, v. 8, n. 3, p. 158–176, 2000. ISSN 1432-1882. Available: <<http://dx.doi.org/10.1007/s005300000042>>. Citations on pages 24 and 44.

NACK, F. Capture and transfer of metadata during video production. In: **Proceedings of the ACM Workshop on Multimedia for Human Communication: From Capture to Convey**. New York, NY, USA: ACM, 2005. (MHC '05), p. 17–20. ISBN 1-59593-247-X. Available: <http://doi.acm.org/10.1145/1099376.1099382>. Citation on page 35.

NETO, O. J. M.; PEREIRA, A. P.; ELUI, V. M. C.; PIMENTEL, M. d. G. C. Posture monitoring via mobile devices: Smartvest case study. In: **Proceedings of the 22Nd Brazilian Symposium on Multimedia and the Web**. New York, NY, USA: ACM, 2016. (Webmedia '16), p. 55–61. ISBN 978-1-4503-4512-5. Available: <http://doi.acm.org/10.1145/2976796.2976870>. Citation on page 92.

OJALA, J.; MATE, S.; CURCIO, I. D. D.; LEHTINIEMI, A.; Väänänen-Vainio-Mattila, K. Automated creation of mobile video remixes: User trial in three event contexts. In: **Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia**. New York, NY, USA: ACM, 2014. (MUM '14), p. 170–179. ISBN 978-1-4503-3304-7. Available: <http://doi.acm.org/10.1145/2677972.2677975>. Citations on pages 24, 25, 46, 47, and 90.

PEARSON, J.; ROBINSON, S.; JONES, M. Paperchains: Dynamic sketch+voice annotations. In: **Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing**. New York, NY, USA: ACM, 2015. (CSCW '15), p. 383–392. ISBN 978-1-4503-2922-4. Available: <http://doi.acm.org/10.1145/2675133.2675138>. Citation on page 24.

PEDERSEN, E. R.; MCCALL, K.; MORAN, T. P.; HALASZ, F. G. Tivoli: An electronic whiteboard for informal workgroup meetings. In: **Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems**. Amsterdam, The Netherlands, The Netherlands: IOS Press, 1993. (INTERCHI '93), p. 391–398. ISBN 90-5199-133-9. Available: <http://dl.acm.org/citation.cfm?id=164632.164957>. Citations on pages 37 and 44.

PIMENTEL, M. G.; ABOWD, G. D.; ISHIGURO, Y. Linking by interacting: A paradigm for authoring hypertext. In: **Proceedings of the Eleventh ACM on Hypertext and Hypermedia**. New York, NY, USA: ACM, 2000. (HYPERTEXT '00), p. 39–48. ISBN 1-58113-227-1. Available: <http://doi.acm.org/10.1145/336296.336315>. Citation on page 37.

PIMENTEL, M. G.; BALDOCHI JR., L. A.; CATTELAN, R. G. Prototyping applications to document human experiences. **IEEE Pervasive Computing**, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 6, n. 2, p. 93–100, Apr. 2007. ISSN 1536-1268. Available: <http://dx.doi.org/10.1109/MPRV.2007.40>. Citation on page 44.

PIMENTEL, M. G.; ISHIGURO, Y.; KERIMBAEV, B.; ABOWD, G. D.; GUZDIAL, M. Supporting educational activities through dynamic web interfaces. **Interacting with computers**, Oxford University Press, v. 13, n. 3, p. 353–374, 2001. Citation on page 24.

PIMENTEL, M. G.; PRAZERES, C.; RIBAS, H.; LOBATO, D.; TEIXEIRA, C. Documenting the pen-based interaction. In: **Proceedings of the 11th Brazilian Symposium on Multimedia and the Web**. New York, NY, USA: ACM, 2005. (WebMedia '05), p. 1–8. Available: <http://doi.acm.org/10.1145/1114223.1114232>. Citation on page 47.

REN, Y.; LI, Y.; LANK, E. Inkanchor: Enhancing informal ink-based note taking on touchscreen mobile phones. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2014. (CHI '14), p. 1123–1132. ISBN 978-1-4503-2473-1. Available: <<http://doi.acm.org/10.1145/2556288.2557302>>. Citation on page 24.

ROWE, L. A. Looking forward 10 years to multimedia successes. **ACM Trans. Multimedia Comput. Commun. Appl.**, ACM, v. 9, n. 1s, p. 37:1–37:7, Oct. 2013. Citations on pages 23, 26, and 35.

Sá, M.; SHAMMA, D. A.; CHURCHILL, E. F. Live mobile collaboration for video production: Design, guidelines, and requirements. **Personal Ubiquitous Comput.**, Springer-Verlag, v. 18, n. 3, p. 693–707, Mar. 2014. Citations on pages 25 and 26.

SCHMIDT, L. B. K. Taking cscw seriously - supporting articulation work. **Computer Supported Cooperative Work**, v. 1, n. 1-2, p. 7–40, 1992. ISSN 09259724. Citation on page 33.

SHENOY, P. Multimedia systems research: The first twenty years and lessons for the next twenty. **ACM Trans. Multimedia Comput. Commun. Appl.**, ACM, New York, NY, USA, v. 9, n. 1s, p. 38:1–38:4, Oct. 2013. ISSN 1551-6857. Available: <<http://doi.acm.org/10.1145/2490859>>. Citations on pages 23, 24, and 107.

SOMMERVILLE, I. **Software Engineering: (Update) (8th Edition) (International Computer Science)**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2006. ISBN 0321313798. Citations on pages 61 and 64.

STANKOVIC, J. A. Research directions for cyber physical systems in wireless and mobile healthcare. **ACM Trans. Cyber-Phys. Syst.**, ACM, New York, NY, USA, v. 1, n. 1, p. 1:1–1:12, Nov. 2016. ISSN 2378-962X. Available: <<http://doi.acm.org/10.1145/2899006>>. Citation on page 23.

STREITZ, N. A.; GEISSLER, J.; HAAKE, J. M.; HOL, J. Dolphin: integrated meeting support across local and remote desktop environments and liveboards. In: **ACM. Proceedings of the 1994 ACM Conference on Computer supported cooperative work**. [S.l.], 1994. p. 345–358. Citations on pages 34 and 37.

Sá, M. de; SHAMMA, D.; CHURCHILL, E. Live mobile collaboration for video production: design, guidelines, and requirements. **Personal and Ubiquitous Computing**, Springer London, v. 18, n. 3, p. 693–707, 2014. Citations on pages 45 and 47.

TEIXEIRA, C. A.; MELO, E. L.; FREITAS, G. B.; SANTOS, C. A.; PIMENTEL, M. D. Discrimination of media moments and media intervals: Sticker-based watch-and-comment annotation. **Multimedia Tools Appl.**, Kluwer Academic Publishers, v. 61, n. 3, p. 675–696, Dec. 2012. Citation on page 38.

TRUONG, K. N.; ABOWD, G. D. Inca: A software infrastructure to facilitate the construction and evolution of ubiquitous capture & access applications. In: **SPRINGER. International Conference on Pervasive Computing**. [S.l.], 2004. p. 140–157. Citation on page 44.

TRUONG, K. N.; ABOWD, G. D.; BROTHERTON, J. A. Personalizing the capture of public experiences. In: **Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology**. New York, NY, USA: ACM, 1999. (UIST '99), p. 121–130. ISBN 1-58113-075-9. Available: <http://doi.acm.org.ez67.periodicos.capes.gov.br/10.1145/320719.322593>. Citations on pages 24, 34, and 107.

TRUONG, K. N.; HAYES, G. R. *et al.* Ubiquitous computing for capture and access. **Foundations and Trends® in Human–Computer Interaction**, Now Publishers, Inc., v. 2, n. 2, p. 95–171, 2009. Citations on pages 24 and 44.

VEGA-OLIVEROS, D. A.; MARTINS, D. S.; PIMENTEL, M. d. G. C. Media-oriented operators for authoring interactive multimedia documents generated from capture sessions. In: **Proc. ACM Symposium on Applied Computing**. [S.l.]: ACM, 2011. (SAC '11), p. 1267–1272. Citations on pages 47 and 129.

VIEL, C. C.; MELO, E. L.; PIMENTEL, M. d. G. C.; TEIXEIRA, C. A. C. Go beyond boundaries of itv applications. In: **Proc. ACM Symposium on Document Engineering**. [S.l.]: ACM, 2013. (DocEng '13), p. 263–272. Citation on page 24.

VIEL, C. C.; MELO, E. L.; PIMENTEL, M. G.; TEIXEIRA, C. A. Multimedia multi-device educational presentations preserved as interactive multi-video objects. In: **Proc. Brazilian Symposium on Multimedia and the Web**. [S.l.]: ACM, 2013. (WebMedia '13), p. 51–58. Citations on pages 44, 47, and 107.

VIEL, C. C.; RODRIGUES, K. R. da H.; MELO, E. L.; BUENO, R.; PIMENTEL, M. da G. C.; TEIXEIRA, C. A. C. Interaction with a problem solving multi video lecture: Observing students from distance and traditional learning courses. **iJET**, v. 9, n. 1, p. 39–46, 2014. Available: <http://www.online-journals.org/index.php/i-jet/article/view/3358>. Citation on page 87.

WEISER, M. The computer for the twenty-first century. **Scientific American**, v. 265, n. 3, p. 94–104, Sep. 1991. Citations on pages 31, 44, and 107.

YEH, R.; LIAO, C.; KLEMMER, S.; GUIMBRETIERE, F.; LEE, B.; KAKARADOV, B.; STAMBERGER, J.; PAEPCKE, A. Butterflynet: A mobile capture and access system for field biology research. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2006. (CHI '06), p. 571–580. ISBN 1-59593-372-7. Available: <http://doi.acm.org/10.1145/1124772.1124859>. Citation on page 24.

YU, Z.; NAKAMURA, Y. Smart meeting systems: A survey of state-of-the-art and open issues. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 42, n. 2, p. 8:1–8:20, Mar. 2010. ISSN 0360-0300. Available: <http://doi.acm.org/10.1145/1667062.1667065>. Citation on page 37.

ZHANG, Y.; ZHANG, L.; ZIMMERMANN, R. Aesthetics-guided summarization from multiple user generated videos. **ACM Trans. Multimedia Comput. Commun. Appl.**, ACM, New York, NY, USA, v. 11, n. 2, p. 24:1–24:23, Jan. 2015. ISSN 1551-6857. Available: <http://doi.acm.org/10.1145/2659520>. Citations on pages 23 and 107.

GLOSSARY

Amateur users: refers as common people that use their own mobile devices to perform a non professional capture and edition of the media captured..

Capture technology: refers to the devices and software use to perform a capture, it could be in a professional or non professional way. They could be cameras, microphones, sensors, cellphones and so on..

Client Application: refers to an application that let communicate the users with a the back-end application with the interchange of information. For example the Facebook app..

Collaborative authoring: refers to perform a multimedia authoring in a collaborative way like for example many people perform comments over a YouTube video..

Collaborative capture scenarios: refers to scenarios where the people that perform the recording, work in a collaborative way..

Collaborative systems: refers to the systems that allow a short team to perform an specific task in a collaborative way like for example when 3 people create a report using *documents* of *Google docs* so they edit the same document at the same time using their own personal computes..

Context-aware information: refers to extra information about a multimedia element that complement the multimedia element like for example the geographic location that indicates where the video was recorded..

Educational domain: refers to recording of educational events like for example classes at the university or thesis defense..

Explicit physical synchronization method: refers to an synchronization method that is performed by an user with a device. The device detect that action using their sensors. For example an user perform a shake movement with a cellphone (shake event)..

Instrumented environments: refers to the multimedia recording devices connected and synchronized in a prepared room that is prepared to record..

Interaction events: refers to meta-data produced in the different stages of the multimedia production process by the interaction between users, devices, media or the live session..

Interactors: refers to operators based on the interaction of a user with some multimedia element..

Low fidelity prototype: (*mockup*) refers to an application developed in a short period of time that is used to have an general idea how it will work..

Media-based annotations: refers to annotations performed in a multimedia element..

Mobile health-care applications: refers to mobile applications that support health like for example a mobile application that help patients to remember the time to take medicine..

Mobile multimedia production: refers to the multimedia production performed only with mobile technology like cellphones and tables..

Mobile streaming: refers to the publish a multimedia element(generally video) in real time using only mobile devices like for example the service of live streaming of Facebook..

Mobile technology: refers to transportable devices that offer instantaneous access to information taken from internet..

Multimedia access: refers to the systems used to visualization of multimedia documents like for example someone watching a video in YouTube that contains subtitles and annotations..

Multimedia authoring: refers to the use of multimedia elements with the objective to enrich and specific multimedia element like for example someone create subtitles in Spanish for a video recorded of a geography class..

Multimedia content: refers to the set of multimedia elements captured in an specific event..

Multimedia document: refers to a digital document composed by a set of multimedia elements..

Multimedia elements: refers to digital elements like for example video, audio, text, geographic locations..

Multimedia production process: refers to the process used in professional video productions, like television and cinema productions.

Native applications: refers to the applications that exist by default in a mobile device like for example the camera application that already existed in the new cellphone..

Portability: refers to the facility to use the same software in different platforms..

Professional capture technology: refers to capture technology to have an professional recording of an event like for example High-definition cameras and professional software to record a class in a university. Generally, tt is expensive..

Social media: are the set of systems (web or mobile) that allow social interaction between users over the internet like for example facebook, instagram or twitter..

Ubiquitous computing: has the objective to turn the human computer interaction the more transparent possible for the users..

Usability: refers to the facility to learn and use a mobile or web application..

User-generated content: refers to the multimedia elements generated by amateur users and published in the social networks..

User-interactions: refers to the interaction that perform a user in a system with the objective to make a task..

CMOVIA API DESCRIPTION

1. **Consult events**, look up all available events(collaborative events) that are in capture process

- Operation: <https://radiant-oasis-3197.herokuapp.com/event>
- Method: GET
- Headers: Accept: application/json
- Output:

Source code 5 – List of available events in JSON

```
1: [  
2:   {  
3:     "created_at": "2015-06-22T16:12:12-03:00"  
4:     "end_date": null  
5:     "id": 41  
6:     "name": "Collaborative Systems class"  
7:     "participant_count": "2"  
8:     "start_date": "2015-06-22T16:12:12-03:00"  
9:     "updated_at": "2015-06-22T16:12:12-03:00"  
10:  },  
11:  {  
12:    "created_at": "2015-05-22T16:10:12-03:00"  
13:    "end_date": null  
14:    "id": 42  
15:    "name": "HCI class"  
16:    "participant_count": "5"  
17:    "start_date": "2015-06-22T16:10:12-03:00"  
18:    "updated_at": "2015-06-22T16:10:12-03:00"  
19:  }  
20: ]
```

2. **Specific event details**, show the detail of an specific event by the *id* attribute.

- Operation: <https://radiant-oasis-3197.herokuapp.com/event/{id}>
- Method: GET

- Headers: Accept: application/json
- Output:

Source code 6 – JSON Event detail for existing {id}

```

1: {
2:   "created_at": "2015-06-22T16:12:12-03:00"
3:   "end_date": null
4:   "id": 41
5:   "name": "Collaborative Systems class"
6:   "participant_count": "2"
7:   "start_date": "2015-06-22T16:12:12-03:00"
8:   "updated_at": "2015-06-22T16:12:12-03:00"
9: }
```

3. Participants of an event, list the context information about all participants of an specific event.

- Operation:
`https://radiant-oasis-3197.herokuapp.com/event/{id}/participants`

- Method: GET
- Headers: Accept: application/json
- Output:

Object composed by the user name and the detail of all media that captured. In the example we can see two participants for the event with existing {id}, *diana.interm* captured photos plus audio and *peter.interm* captured just video.

Source code 7 – Detailed list of context information about the event with existing{id} in JSON

```

1: {
2:   "diana.interm": [
3:     {
4:       "created_at": "2015-05-27T11:12:12-03:00",
5:       "end_date": "2016-01-20T18:23:53-02:00",
6:       "id": 2,
7:       "quality": "high",
8:       "start_date": "2016-01-20T18:21:05-02:00",
9:       "type_name": "images",
10:      "updated_at": "2015-05-27T11:12:12-03:00"
11:    },
12:    {
13:      "created_at": "2015-05-27T11:12:12-03:00",
14:      "end_date": "2016-01-20T18:23:53-02:00",
15:      "id": 3,
16:      "quality": "normal",
17:      "start_date": "2016-01-20T18:21:05-02:00",
18:      "type_name": "audio",
19:      "updated_at": "2015-05-27T11:12:12-03:00"
20:    }

```

```

21:   ],
22:   "peter.interm": [
23:     {
24:       "created_at": "2015-05-27T11:12:12-03:00",
25:       "end_date": "2015-12-17T16:10:25-02:00",
26:       "id": 1,
27:       "quality": "low",
28:       "start_date": "2015-12-17T16:10:04-02:00",
29:       "type_name": "video",
30:       "updated_at": "2015-05-27T11:12:12-03:00"
31:     }
32:   ]
33: }

```

4. **New event**, create a new event and the capture of the first user.

- Operation: <https://radiant-oasis-3197.herokuapp.com/event>
- Method: POST
- Headers: Accept: application/json, Content-Type: application/json
- Input:

Json object with the follow parameters: name(name of the new event), username(user name of Google+) and media(list of media identificator that are recording). In this example *omar.mozo* captured photos plus audio using the Bluetooth headset.

Source code 8 – Details of New event in JSON

```

1: {
2:   "capture": {
3:     "media": [
4:       2,
5:       4
6:     ],
7:     "username": "omar.mozo",
8:     "name": "Ubicomp class"
9:   }
10: }

```

- Output: The detail of the event and the capture created.

Source code 9 – JSON object with capture and event detail for event existing {id}

```

1: {
2:   "event": {
3:     "created_at": "2016-01-21T20:26:31-02:00",
4:     "end_date": null,
5:     "id": 84,
6:     "name": "Ubicomp class",
7:     "start_date": "2016-01-21T20:26:31-02:00",
8:     "updated_at": "2016-01-21T20:26:31-02:00"

```

```

9:   },
10:   "capture":{
11:     "created_at":"2016-01-21T20:26:31-02:00",
12:     "data":null,
13:     "end_date":null,
14:     "event_id":84,
15:     "id":357,
16:     "mime_type":null,
17:     "person_id":12,
18:     "start_date":"2016-01-21T20:26:31-02:00",
19:     "updated_at":"2016-01-21T20:26:31-02:00"
20:   }
21: }

```

5. **Join existing event**, a new participant take part of an existing event.

- Operation:
`https://radiant-oasis-3197.herokuapp.com/event/{id}/join`
- Method: PUT
- Headers: Accept: application/json, Content-Type: application/json
- Input:
Json object with the follow parameters: username(user name of Google+) and media(list of media identificator that are recording). In this example *miguel* join a session capturing photos plus audio.

Source code 10 – Capture object in join operation for a new participant in JSON

```

1: {
2:   "capture":{
3:     "username":"miguel",
4:     "media":[
5:       2,
6:       3
7:     ]
8:   }
9: }

```

- Output:
The detail of the event that *miguel* join into the event with existing {id}. The output is the same of Code 9

6. **Left an event**, when a participant want to stop recording, notify it to the server.

- Operation:
`https://radiant-oasis-3197.herokuapp.com/event/{id}/left`
- Method: PUT
- Headers: Accept: application/json, Content-Type: application/json

- Input:
JSON object with the follow parameters: username(user name of Google+) . In this example *miguel* left the session.

Source code 11 – Capture object in join operation for a new participant in JSON

```

1: {
2:   "capture":{
3:     "username":"miguel"
4:   }
5: }
```

- Output:
The detail of the event that *miguel* left the event with existing {id}. The output is the same of Code 9

7. Upload photo preview, when a participant ,that is recording a collaborative capture, wants to send a photo preview about their capture.

- Operation:
`https://radiant-oasis-3197.herokuapp.com/event/{id}/thumb`
- Method: PUT
- Headers: **Accept:** `multipart/form-data`
- Parameters: Form object with the follow parameters: username(user name of Google+), id(event id), data(image in binary type)

Source code 12 – Form object that include a photo in binary to upload

```

1: username = omar.mozo &
2: id = 41 &
3: data = (binary data)
```

- Output:
The detail of the event where *omar.mozo* send the photo. The output is the same of Code 9

8. Recover photo preview, when a new participant wants to recover a photo preview of any participant in the collaborative capture.

- Operation:
`https://radiant-oasis-3197.herokuapp.com/event/{id}/thumb`
- Method: PUT
- Headers: **Accept:** `image/jpeg`
- Parameters:
The following parameters: id(event id) and username(user name of Google+)

- Output:
Photo preview about the event and the active user

EXAMPLES OF INTERACTORS

Table 4 – Examples of Interactors

Category	Features	Name	Description
All Interactors	attribute	filterByAttribute	list of media element attributes is defined
		filterByAttributeValue	list of mappings between media element attributes and corresponding values is satisfied
		idleMoments	no interaction of a specific type has taken place
AudioInteractor	time	silenceMoments	list of time moments when there were no voices for a period of time in a time interval
		spokenMoments	list of time moments before which someone has spoken for a period of time in a time interval
	attribute	voiceIncrease	list of time moments when there was a consistent increase in speech volume in the given time interval
		conversation	potential number of participants who spoke during this interval
		outstandingMoments	list of time moments when there were outstanding moments (several people are talking at the same time) in the media element

Table 4 – Examples of Interactors

Category	Features	Name	Description
	action	enterAudioMute	list of time moments, within the given time interval, when the mute function was activated
		exitAudioMute	list of time moments, within the given time interval, when the mute function was deactivated
BoardInteractors	time	ChangeBoard	list of time moments when there was a change of slide within the specified time interval
		IdleBoard	list of time moments when there was no change of slide for a period of time within the specified time interval
	attribute	ChangeOnAttribute	list of time moments when there was a change according to selected attribute within the time interval
		FilterByAttributeValue	list of time moments when slides were changed according the value of a selected attribute, within the time interval
TextInteractor	attribute	ChangeOnAttribute	list of time moments when the text message were changed according to a selected attribute
		FilterByAttributeValue	list of time moments when text messages were changed according the value of a selected attribute
	time	silenceMoments	list of time moments when there were no messages exchanged for a period of time
		textMoments	list of time moments before which someone has typed messages for a period of time
VideoInteractors	time	blankMoments	moments when no image was presented for a period of time
		imageMoments	moments before which a user was presented with an image a period of time

Table 4 – Examples of Interactors

Category	Features	Name	Description
		imageIntervals	start moment of time intervals in which a user was presented with an image for a period of time
	attribute	imageType	moments when an image of the given type was presented(e.g. PNG or JPG)
		imageSize	moments when an image of the given size was presented
		cutMoments	moments when there was a cut in the target video due to change of cameras
Inkteractors	time	TimeSlice	list of the ink strokes generated in the specified time interval
	attribute	ChangeOnAttribute	list of time moments when ink strokes were changed according to a selected attribute
		FilterByAttributeValue	list of time moments when ink strokes were changed according the value of a selected attribute
	action	ChangeOnAuthor	list of time moments when there was a change in the author of the strokes
		FilterByAuthor	list of time moments when there was a change in the author of the strokes so that the author is identified by ID, within the given time interval
	position	ChangeOnArea	list of time moments when there was a change in the specified area
		FilterByArea	list of ink strokes drawn in the given area during the specified time interval

Interactors: adapted from the work by [Martins, Vega-Oliveros and Pimentel \(2011\)](#), [Vega-Oliveros, Martins and Pimentel \(2011\)](#), [Martins \(2014\)](#).