Estêvão Bissoli Saleme

A Conceptual Architecture and a Framework for Dealing with Variability in Mulsemedia Systems

Vitória-ES, Brazil

December 2019

Estêvão Bissoli Saleme

A Conceptual Architecture and a Framework for Dealing with Variability in Mulsemedia Systems

Doctoral thesis submitted to the Postgraduate Program in Informatics of the Federal University of Espírito Santo as a partial requirement to obtain the degree of Doctor of Computer Science.

Federal University of Espírito Santo

Technological Center

Postgraduate Program in Informatics

Supervisor: Prof. Dr. Celso Alberto Saibel Santos

Co-supervisor: Prof. Dr. Gheorghita Ghinea

Vitória-ES, Brazil

December 2019

Ficha catalográfica disponibilizada pelo Sistema Integrado de Bibliotecas - SIBI/UFES e elaborada pelo autor

---

S163c

Saleme, Estêvão Bissoli, 1983-

A Conceptual Architecture and a Framework for Dealing with Variability in Mulsemedia Systems / Estêvão Bissoli Saleme. - 2019.

146 f. : il.

Orientador: Celso Alberto Saibel Santos.

Coorientador: Gheorghita Ghinea.

Tese (Doutorado em Informática) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Mulsemedia systems. 2. Multimedia applications. 3. Variability. 4. Conceptual architecture. 5. Software Integration. 6. Frameworks. I. Santos, Celso Alberto Saibel. II. Ghinea, Gheorghita. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 004

A CONCEPTUAL ARCHITECTURE AND A FRAMEWORK FOR DEALING WITH VARIABILITY IN MULSEMEDIA SYSTEMS

Estêvão Bissoli Saleme

Tese submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Aprovada em 03 de dezembro de 2019:

Prof. Dr. Celso Alberto Saibel Santos
Orientador

Prof. Dr. Gheorghita Ghinea
Coorientador

Prof. Dr. José Gonçalves Pereira Filho
Membro Interno

Prof. Dr. Magnos Martinello
Membro Interno

Prof[a]. Dr[a]. Débora Christina Muchaluat Saade
Membro Externo, participação remota

Prof. Dr. Niall Murray
Membro Externo, participação remota

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
Vitória-ES, 03 de dezembro de 2019.

This doctoral thesis is dedicated to my family.

Acknowledgements

First and foremost, I would like to thank God, my fortress and protector. He provided me with the necessary wisdom to undertake this research. Without him, this accomplishment would not have been possible.

My endless gratitude goes to my supervisor, Prof. Celso Alberto Saibel Santos, who has trusted me since I started researching in my master's (2013) under his supervision. He has encouraged me to participate in many different academic activities, which has been key to evolving my abilities. I could not reach this stage without his sage advice.

I am utterly thankful to Prof. Gheorghita Ghinea, my co-supervisor, who graciously received me at Brunel University in London, UK, for six unforgettable months of hard work, ceaseless learning, and huge cultural enrichment.

I would like to thank my journey colleagues at UFES, Jordano, Renato, Marcello, Fábio, and Alessandro. They have not only coauthored papers with me but have also shared their joy and distress we face throughout the course. I am also grateful to my colleagues at Brunel University London (Gebremarian, Ioan, and Nadia), University of Kent (Alexandra), Middlesex University (Ramona), and University of West London (Elahe), and University of Wales Trinity Saint David (Kemi), for giving me the opportunity to visit and collaborate with them. Moreover, I am glad to have collaborated with Prof. Falbo (UFES) and Prof. Andres (National Institute of Informatics, Japan) in subjects related to mulsemedia.

Special thanks go to the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 88881.187844/2018-01 (six months), the Federal Institute of Espírito Santo (IFES), the Fundação de Amparo à Pesquisa e Inovação do Espírito Santo (FAPES), and the Association for Computing Machinery's Special Interest Group on Multimedia (ACM SIGMM), who have conceded grant awards and financial aid to support my research activities.

A further expression of gratitude goes to the professors and staff members of the Department of Computer Science at UFES, who provided me with knowledge and the necessary infrastructure to carry out my research. I would also like to thank all members of the examination committee of this doctoral thesis for their collaboration and constructive feedback.

Last, but not least, I could not forget the unshakable support and forbearance from my family, especially my wife, Talita, my parents, my sisters, and my in-laws. They have not only shared their love with me but have also taught me to be more resilient.

Abstract

The increasing interest in digital immersive experiences has drawn the attention of researchers into understanding human perception whilst adding sensory effects to multimedia systems such as VR (Virtual Reality) and AR (Augmented Reality) applications, multimedia players, and games. These so-called mulsemedia—multiple sensorial media—systems are capable of delivering wind, smell, vibration, among others, along with audiovisual content with the aim of enhancing users' Quality of Experience (QoE) in areas such as entertainment, healthcare, education, culture, and marketing. To support the researchers' investigation, there have been developed many standalone software solutions and incipient architectural proposals to bind these applications to sensory effects devices, such as wind fans, scent emitters, vibration chairs, etc. These devices, in turn, are constantly evolving, making it difficult to update applications to be compatible with them. There is little or no interoperability between software and hardware in this realm, hindering reuse in other contexts. Every time a mulsemedia application is needed, new software is built mostly from scratch. This model has proven to be demanding, time-consuming, and costly mainly because it requires researchers and developers alike to gain knowledge about new devices, connectivity, communication protocols, and other particulars. The fact is that building such systems imposes a number of challenges and requirements (which are discussed in this thesis) due mainly to their ever-evolving and heterogeneous traits. As a result, few mulsemedia systems have remained reusable to be applied to different research purposes as opposed to the use of open mulsemedia datasets. Therefore, the main contribution of this thesis is a decoupled conceptual architecture to deal with variability of scenarios in mulsemedia delivery systems, which includes recommendations to cope with the variation of end-user applications and sensory effect devices through the support and reuse of even unforeseen communication and connectivity protocols, and sensory effects metadata (SEM). To evaluate it, an open-source and robust mulsemedia framework was developed. Then, a performance assessment was carried out on communication protocols for the integration between event-based applications, whereby temporal restrictions play a role, and the framework. Results indicated statistically significant differences in response time providing directions for optimized integrations. Finally, a user QoE subjective evaluation comparing a monolithic mulsemedia system with this framework was undertaken with results suggesting no evinced statistically significant differences in user-perceived QoE between the systems under different aspects. Therefore, it is hoped that this work fosters the area of mulsemedia and HCI (Human-Computer Interaction) in the sense that researchers can leverage either the conceptual architecture to design mulsemedia delivery systems or the framework to carry out their experiments.

Keywords: Mulsemedia systems. Multimedia applications. Variability. Conceptual architecture. Software Integration. Frameworks.

Resumo

O crescente interesse em experiências imersivas digitais têm atraído a atenção dos pesquisadores para a compreensão da percepção humana quando efeitos sensoriais são adicionados a sistemas multimídia, tais como aplicações de realidade virtual e aumentada, reprodutores multimídia e jogos. Os sistemas mulsemídia são capazes de prover vento, cheiro, vibração, entre outros efeitos sensoriais, junto com conteúdo audiovisual com o objetivo de melhorar a Qualidade de Experiência (QoE) dos usuários em áreas tais como entretenimento, saúde, educação, cultura e marketing. Para apoiar a investigação dos pesquisadores, várias soluções *standalone* de software e propostas arquitetônicas incipientes têm sido desenvolvidas para vincular essas aplicações a dispositivos de efeitos sensoriais, tais como ventiladores, emissores de odores, cadeiras vibratórias, etc. Esses dispositivos, por sua vez, estão em constante evolução, dificultando a atualização de aplicativos para se tornarem compatíveis com eles. Há pouca ou nenhuma interoperabilidade entre software e hardware neste domínio, impedindo a reutilização em outros contextos. Toda vez que uma aplicação mulsemídia é necessária, um novo software é construído a partir do zero. Esse modelo têm se mostrado trabalhoso, demorado e oneroso principalmente porque exige que pesquisadores e desenvolvedores adquiram conhecimento sobre novos dispositivos, protocolos de conectividade e de comunicação além de outras características técnicas. O fato é que a construção de tais sistemas impõe uma série de desafios e requisitos devido principalmente a seus traços evolutivos e heterogêneos. Consequentemente, poucos sistemas mulsemídia têm permanecidos reusáveis, na direção oposta a datasets abertos mulsemídia. Portanto, a principal contribuição desta tese é uma arquitetura conceitual desacoplada para lidar com a variabilidade de cenários em sistemas de entrega mulsemídia, que inclui recomendações para suportar a mudança de aplicações de apresentação e dispositivos de efeitos sensoriais através do suporte e reutilização de até mesmo protocolos não previstos de comunicação e conectividade, e metadados de efeitos sensoriais (SEM). Para avaliá-lo, um *framework* mulsemídia de código aberto e robusto foi desenvolvido. Em seguida, foi realizada uma avaliação de desempenho de protocolos de comunicação para a integração entre aplicações baseadas em eventos, em que as restrições temporais desempenham um papel importante, e o *framework*. Os resultados indicaram diferenças estatisticamente significativas no tempo de resposta, fornecendo orientações para integrações otimizadas. Por fim, uma avaliação subjetiva de QoE do usuário comparando um sistema monolítico mulsemídia com o *framework* foi realizada, com resultados sugerindo que não houve diferenças estatisticamente significativas na QoE percebida pelos usuários entre os sistemas sob diferentes aspectos. Portanto, espera-se que este trabalho fomente a área da mulsemídia e interação humano-computador, no sentido de que os pesquisadores possam aproveitar a arquitetura conceitual para projetar sistemas de entrega mulsemídia ou o *framework* para realizar seus experimentos.

Palavras-chave: Sistemas mulsemídia. Aplicações multimídia. Variabilidade. Arquitetura conceitual. Integração de software. *Frameworks*.

List of Figures

List of Tables

List of abbreviations and acronyms

API	Application Programming Interface
AR	Augmented Reality
ARAIG	As Real As It Gets
CoAP	Constrained Application Protocol
DIY	Do It Yourself
EU	European Union
HCI	Human-Computer Interaction
HMD	Head-Mounted Display
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
IP	Internet Protocol
KNX	Konnex
MCU	Microcontroller Unit
MOS	Mean Opinion Score
MQTT	Message Queuing Telemetry Transport
Mulsemedia	Multiple Sensorial Media
MUVII	Multi User Virtual Interactive Interface
NCL	Nested Context Language
OO	Object-Oriented
PC	Personal Computer
QoE	Quality of Experience
QoS	Quality of Service
SDK	Software Development Kit

SE	Sensory Effects
SEM	Sensory Effects Metadata
SER	Sensory Effects Renderer
TCP	Transmission Control Protocol
TS	Transport Stream
UPnP	Universal Plug and Play
URL	Uniform Resource Locator
USB	Universal Serial Bus
VR	Virtual Reality
XML	Extensible Markup Language

Contents

1 Introduction

This chapter introduces an overview of this thesis highlighting its context, motivation, research goals, and structure.

1.1 Context and Motivation

Over the past years, mulsemedia has enticed end-users into more realistic and immersive experiences by providing them the opportunity to enjoy multimedia content enriched with sensations of smell, touch, and taste (GHINEA et al., 2014). Many researchers have carried out studies to understand how humans perceive these extra stimuli, and thus, to apply multisensory experiences to vast areas across the industry. They have conducted user QoE (Quality of Experience) experiments to investigate levels of satisfaction and annoyance whilst consuming mulsemedia content under varied settings (WALTL; TIMMERER; HELLWAGNER, 2010a; RAINER et al., 2012; YUAN et al., 2014; MURRAY et al., 2017; MONKS et al., 2017; JALAL et al., 2018b). In entertainment, video players, VR (Virtual Reality) and AR (Augmented Reality) applications, games, and movie theaters have employed mulsemedia to enhance enjoyment (WALTL et al., 2013; YECIES, 2016; VI; ARTHUR; OBRIST, 2018; RANASINGHE et al., 2018). In healthcare, multisensory applications have been used for simulation, training, and treatments (SPENCER, 2006; SERRANO; BAÑOS; BOTELLA, 2016; MEULEN et al., 2016; STONE et al., 2017). In education, it is believed that multisensory learning can be more effective for students (ZOU et al., 2017; COVACI et al., 2018). In culture, museums and exhibitions have engaged users in multisensory artwork (CHU et al., 2016; CLAISSE et al., 2018). In marketing, Petit, Velasco, and Spence (2019) have pointed out a plethora of technologies to deliver in-store multisensory experiences to offer to customers. The possibilities and opportunities for multisensory experiences are countless (GHINEA et al., 2014; OBRIST et al., 2016; SULEMA, 2016).

In order to create mulsemedia systems and to support the researchers' investigation, heterogeneous software (where end-users primarily interact with) and hardware (which will generate sensory effects) must be combined to deliver a variety of effects of lighting, wind, vibration, smell, among others, under varied conditions and restrictions (GHINEA et al., 2014; SULEMA, 2016; COVACI et al., 2018; SALEME et al., 2019a). Towards this end, SDKs (Software Development Kit) and APIs (Application Programming Interface) have been reported as standalone solutions to access sensory effect devices (CONTI, 2003; ITKOWITZ; HANDLEY; ZHU, 2005; KAKLANIS; VOTIS; TZOVARAS, 2015; SANFILIPPO; WEUSTINK; PETTERSEN, 2015; KOLSANOV et al., 2016; GALLACHER et al., 2016; BALZAROTTI; BAUD-BOVY, 2018; MURRAY et al., 2017b; HOWELL et al., 2016; MCGOOKIN; ESCOBAR, 2016; DOBBELSTEIN; RUKZIO; HERRDUM, 2017; CANNA et al., 2019). In this context, these solutions have provided means to develop applications that support some type of sensory effect from some sort of computer application. In contrast to these solutions, there have been incipient proposals of conceptual architectures and frameworks (SUK; HYUN; YONG, 2009; CHOI; LEE; YOON, 2011; YOON, 2013) taking into account some degree of standardization, including the MPEG-V standard (YOON et al., 2015), to achieve interoperability and deal with variability of scenarios of usage to conceive mulsemedia systems. Another trendy approach to creating mulsemedia systems is the IoT (Internet of Things), which has also emerged as a potential solution to integrate different devices in mulsemedia environments (JALAL et al., 2018a; LIN; YANG; LIN, 2018).

Besides having a heterogeneous trait, sensory effect devices are constantly evolving, which imposes hurdles for applications to be compatible with them. Whenever a mulsemedia application is needed, new software is built mostly from scratch because either most of them do not support standardized SEM (Sensory Effects Metadata) (YOON et al., 2015) nor have proper mechanisms to allow hardware replacement without coding. A situation to illustrate this outlook is when users wish to use different rendering devices and multimedia applications for the same purpose because they are not satisfied with some features in them or just because others present new characteristics craved by them. This change is not straightforward in mulsemedia systems mostly because the solutions presented in the literature (CHO, 2010; WALTL et al., 2013; KIM; JOO, 2014; LUQUE et al., 2014; BARTOCCI et al., 2015) were devised for too specific purposes except for multiple compatibilities albeit some of them make an initial effort (SALEME; SANTOS, 2015; JALAL et al., 2018a; LIN; YANG; LIN, 2018) to cope with it. Applications and devices designed for the same purpose are seldom interchangeable. This model has proven to be demanding, time-consuming, and costly mainly because it requires researchers and developers alike to gain knowledge about new devices, connectivity, communication protocols, and other particulars over and over again. Thus, few mulsemedia systems have remained reusable to be applied to different research purposes as opposed to the use of open mulsemedia content datasets (WALTL et al., 2012; MURRAY et al., 2017a), which have been reportedly reused in (ADEMOYE; GHINEA, 2009; GHINEA; ADEMOYE, 2010; WALTL; TIMMERER; HELLWAGNER, 2010a; WALTL; TIMMERER; HELLWAGNER, 2010b; ADEMOYE; GHINEA, 2013; MURRAY et al., 2013b; MURRAY et al., 2014; WALTL et al., 2014; MURRAY et al., 2016; ADEMOYE et al., 2016; MURRAY et al., 2017; AMORIM et al., 2019).

Although the concern for allowing software applications to reach the sensory effect devices is self-evident from the creation of some standardization, SDKs and APIs, incipient conceptual architectures and frameworks, and IoT solutions, many issues have remained unsolved to ultimately provide mechanisms for researchers to reuse digital multisensory systems in variable scenarios of usage. The only current standard, MPEG-V, has not been widely adopted due to uncertain reasons (steep learning curve owing to its complex set of 2 languages and 7 vocabularies, personal preferences, or avoidance of paying royalties to the MPEG group). SDKs and APIs have dealt with one sense at a time requiring, thereby, the systematic combination of them in mulsemedia systems. The issue is that they have distinct features and implementations. As a result, their integration might not be supported on the same operating system. As for the incipient conceptual architectures and frameworks found in the literature, they still need to be evolved and materialized somehow to real environments. With regard to the IoT solutions, if not properly assembled, the system can become cumbersome and as a result, include undesired delays. Furthermore, these solutions would have to adapt themselves to support interaction with heterogeneous timeline- and event-based applications and take into consideration standardization and techniques to cope with SEM, which are not the business of IoT platforms. The case of interactive event-based applications is singular because every millisecond is precious, that is, as soon as an event occurs in an interactive application, a mulsemedia renderer has to deliver some types of sensory effects, such as haptics, as swiftly as possible without spoiling users QoE (ADELSTEIN; LEE; ELLIS, 2003; RANK; SHI; HIRCHE, 2010; KIM; OSGOUEI; CHOI, 2017). Tolerable times are quite difficult to pinpoint since they might vary from one context to another (ANTONAKOGLOU et al., 2018); nonetheless, these temporal restrictions are a concern which mulsemedia systems shall take into account.

By following traditional software engineering techniques, such as conditional compilation of source code, *system variability* can be enabled (MISTRIK; GALSTER; MAXIM, 2019). This concept is related to a system being customized for specific needs through adaptations in its architecture to support different scenarios of usage (GROHER; WEINREICH, 2013; GALSTER; AVGERIOU, 2014; MISTRIK; GALSTER; MAXIM, 2019). However, unprecedented types of software, such as VR and AR applications, and hardware (wind fans, smell machines, collar heaters, vibrating vests, etc.) to deliver sensory effects have increased the complexity of developing reusable mulsemedia systems and required for the accommodation of even unforeseen technologies. Therefore, mulsemedia systems have suffered an evolutionary pressure for more mature and adjustable features that can encompass a broader range of contexts to meet mutable and unusual requirements. Indeed, incipient examples whereby researchers have reused some mulsemedia components with different devices are found in (JALAL et al., 2018b; JALAL et al., 2018a), where the authors used a third party mulsemedia renderer (SALEME; SANTOS, 2015) to compose their mulsemedia system and to undertake QoE evaluations, and (COVACI et al., 2019b), who used the framework described in (SALEME; SANTOS; GHINEA, 2019b). The need for reusable components for multisensory computational applications has also been raised by Obrist et al. (2017), who noticed the lack of frameworks to let HCI (Human-Computer Interaction) researchers and designers exploit multisensory interactions, whereas Akhtar et al. (2019) have manifested the need for resources (databases, open-source software, and experimental setups) that allow reproducible research in multimedia QoE assessment.

Building and integrating such complex systems in an adaptable fashion impose many challenges that involve multifunctionality (to provide operations for different applications to support multisensory effects) and reusability (to accommodate these changes constantly), reactivity and timeliness to ensure reliable mechanisms for quick response time under temporal constraints, and manageability and configurability to deal with software and hardware heterogeneity with minimal or no coding (BROY, 2006; SALEME; SANTOS; GHINEA, 2019a).

Therefore, this work proposes a conceptual architecture for mulsemedia delivery systems (implemented by means of a framework) that takes reuse at a great extension, supports diversified protocols required by applications and different types of multimedia applications themselves (timeline- and event-based), and accommodates upcoming standards and technologies. Furthermore, it takes into account delay eventually introduced by software integration and hardware mechanical processes, provides means of customization by configuration without changing internal components, and considers future growth not relying only on existent technologies, protocols, and standards. In light of this, end-users should be oblivious whether the mulsemedia system is a complex set of intertwined components or whether it is a monolithic application—all that matters is their experience, which should not be affected by the system's architecture. Thus, it is hoped that researchers can leverage either the conceptual architecture to design mulsemedia delivery systems or the framework to carry out their experiments.

1.2 Research Goals, Objectives, and Contributions

The general objective of this thesis is to design a decoupled conceptual architecture to deal with variability of scenarios in mulsemedia delivery systems implementing it by means of a framework, and then, evaluate it from the point of view of performance of communication protocols for the integration between event-based applications and a user QoE subjective evaluation comparing an existent monolithic mulsemedia system with this framework.

From the motivation, the following general research questions and specific objectives to answer them are defined as follows.

- How can researchers and developers design/leverage reusable mulsemedia systems for different contexts considering varied end-user applications and heterogeneous devices?
  - Introduction of a survey that makes evident the heterogeneous trait of mulsemedia software and hardware highlighting gaps and shortcomings in this field;
  - Identification of the challenges for mulsemedia delivery systems and requirements to be met so as to overcome them;
  - Proposal of a flexible conceptual architecture that aims to be independent of technology to tackle the challenges and meet the emerging requirements;
  - Materialization of the conceptual architecture into a mulsemedia delivery framework capable of properly integrating with diversified applications (both timeline- and event-based ones) and heterogeneous rendering devices;
  - Implementation of different case studies to demonstrate the framework's capability to adapt itself to different scenarios of usage through configuration;

- How can networked event-based mulsemedia systems have their performance improved to avoid undesired delays, which would eventually spoil user QoE?
  - Assessment of the framework's performance with event-based applications, which require prompt responses;
- Do users perceive mulsemedia experiences differently when mulsemedia systems are monolithic or have a decoupled implementation approach?
  - Evaluation of users' QoE when exposed to a mulsemedia system using the framework and an existent monolithic mulsemedia system, making a comparison between them.

The scope of this work is limited to these aforementioned objectives. Therefore, it does not cover how to create a model to enhance QoE, to instruct how to develop mulsemedia devices, to demonstrate how to encode mulsemedia, and to present mechanisms to adapt content and context according to user preferences.

1.3 Thesis Outline

The remainder of this thesis is organized into the following chapters.

Chapter 2 presents a background on mulsemedia and important elements to understand this work, such as different types of mulsemedia applications (timeline- and event-based), temporal constraints, and QoE (including mulsemedia experiences and evaluation methods). Solutions of software to operate and support sensory devices, as well as incipient conceptual architectures and frameworks, and programming languages are presented and discussed. Finally, a summary brings gaps and shortcomings from related work.

In Chapter 3, challenges and requirements from the obstacles encountered in related work are presented as well as the proposal of a decoupled conceptual architecture for mulsemedia systems that meets these requirements. The proposal incorporates abstract techniques to address issues in this domain and, therefore, it presents a design that takes into account recurrent problems mainly related to variability of scenarios of usage in mulsemedia systems without compromising reusability.

Chapter 4 describes a practical mulsemedia framework that relies on the proposed conceptual architecture. It implements the concepts claimed by the architectural model to fulfill the requirements. In particular, aspects such as communication and connectivity protocols, standards, interaction behavior, and other particulars are depicted.

Chapter 5 brings case studies and prospects of experimental results from the perspectives of QoS (Quality of Service) and QoE. The case studies present heterogeneous real-world scenarios of usage in which the framework has been materialized. Thereafter, a performance assessment on communication protocols for the integration between event-based applications (whereby temporal restrictions play a role) and the framework is reported. Finally, a subjective QoE experiment whereby users rate different aspects of their experience using an existent monolithic mulsemedia system and this framework is presented and then compared to find out whether the latter's architecture has an impact on user-perceived QoE.

Chapter 6 summarizes the contributions to the field of mulsemedia, answering the research questions. Furthermore, it outlines the current limitations and provides a basis for future work.

1.4 List of Publications

Papers that directly contribute to this thesis have been published/accepted in journals and conferences as follows.

- Journal: SALEME, E. B.; COVACI, A.; MESFIN, G.; SANTOS, C. A. S.; GHINEA, G. Mulsemedia DIY: A Survey of Devices and a Tutorial for Building Your Own Mulsemedia Environment. *ACM Computing Surveys*, v. 52, n. 3, p. 58:1–58:29, jun. 2019. ISSN 0360-0300.

- Journal: SALEME, E. B.; SANTOS, C. A. S.; GHINEA, G. Coping with the challenges of delivering multiple sensorial media. *IEEE MultiMedia*, v. 26, n. 2, p. 66–75, April 2019. ISSN 1070-986X.

- Journal: SALEME, E. B.; SANTOS, C. A. S.; GHINEA, G. A mulsemedia framework for delivering sensory effects to heterogeneous systems. *Multimedia Systems*, v. 25, n. 4, p. 421–447, Aug 2019. ISSN 1432-1882.

- Journal: COMSA, I.-S.; SALEME, E. B.; COVACI, A.; MESFIN, G.; TRESTIAN, R.; SANTOS, C. A. S.; GHINEA, G. Do I Smell Coffee? The Tale of a 360° Mulsemedia Experience. *IEEE MultiMedia*, 2019. ISSN 1070-986X.

- Conference: COVACI, A.; TRESTIAN, R.; SALEME, E. B.; COMSA, I.-S.; MESFIN, G.; SANTOS, C. A. S.; GHINEA, G. 360° Mulsemedia: A Way to Improve Subjective QoE in 360° Videos. In: *Proceedings of the 27th ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2019. (MM '19), p. 2378–2386. ISBN 978-1-4503-6889-6.

- Conference: SALEME, E. B.; SANTOS, C. A. S.; GHINEA, G. Improving response time interval in networked event-based mulsemedia systems. In: *Proceedings of the 9th ACM Multimedia Systems Conference*. New York, NY, USA: ACM, 2018. (MMSys '18), p. 216–224. ISBN 978-1-4503-5192-8.

- Conference: SALEME, E. B.; CELESTRINI, J. R.; SANTOS, C. A. S. Time Evaluation for the Integration of a Gestural Interactive Application with a Distributed Mulsemedia Platform. In: *Proceedings of the 8th ACM on Multimedia Systems Conference - MMSys'17*. New York, New York, USA: ACM Press, 2017. p. 308–314. ISBN 9781450350020.

Other relevant contributions to the field of mulsemedia have also been made along this doctoral journey:

- Journal: COVACI, A.; SALEME, E. B.; MESFIN, G. A.; HUSSAIN, N.; KANI-ZABIHI, E.; GHINEA, G. How do we experience crossmodal correspondent mulsemedia content? *IEEE Transactions on Multimedia*, 2019.

- Journal: SALEME, E. B.; SANTOS, C. A. S.; FALBO, R. A.; GHINEA, G.; ANDRES, F. MulseOnto: a Reference Ontology to Support the Design of Mulsemedia Systems. *Journal of Universal Computer Science*, v. 25, n. 9, 2019.

- Journal: AMORIM, M. N. de; SALEME, E. B.; NETO, F. R. de A.; SANTOS, C. A. S.; GHINEA, G. Crowdsourcing authoring of sensory effects on videos. *Multimedia Tools and Applications*, v. 78, n. 14, p. 19201–19227, Jul 2019.

- Journal: MESFIN, G. A.; HUSSAIN, N.; KANI-ZABIHI, E.; COVACI, A.; SALEME, E. B.; GHINEA, G. QoE of Cross-modally Mapped Mulsemedia: an Assessment Using Eye Gaze and Heart Rate. *Multimedia Tools and Applications*, 2019.

- Conference: SALEME, E. B.; SANTOS, C. A. S.; FALBO, R. A.; GHINEA, G.; ANDRES, F. Towards a reference ontology on mulsemedia systems. In: *Proceedings of the 10th International Conference on Management of Digital EcoSystems*. New York, NY, USA: ACM, 2018. (MEDES '18), p. 23–30. ISBN 978-1-4503-5622-0.

- Conference: SALEME, E. B.; COVACI, A.; MESFIN, G.; SANTOS, C. A. S.; GHINEA, G. Sumarização de dispositivos de efeitos multissensoriais para interações humano-computador. In: *Extended Proceedings of IHC '19: XVIII Brazilian Symposium on Human Factors in Computing Systems*. Porto Alegre, RS, Brazil: SBC, 2019. (IHC '19).

2 Background and Related Work

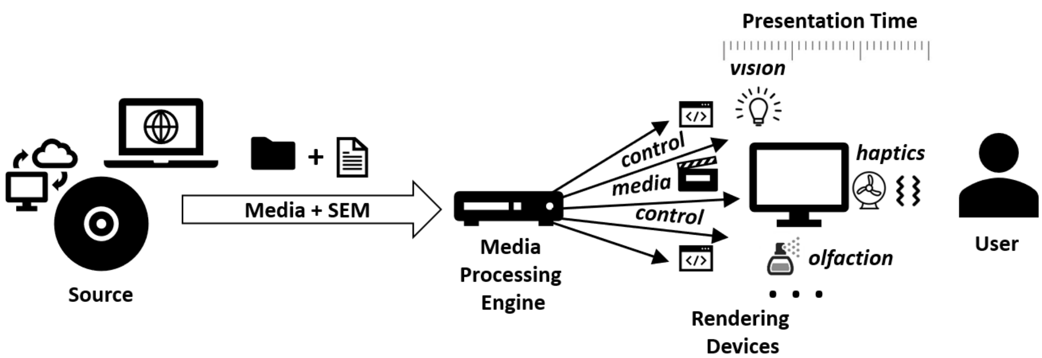
This chapter introduces the concepts about mulsemedia systems that are useful to understand this thesis, including different types of mulsemedia applications (timeline- and event-based), temporal constraints, and QoE in the context of mulsemedia experiences and evaluation methods. Moreover, it encompasses the state-of-the-art of software that deals with haptic, olfactory, and gustatory devices to deliver sensory effects, presenting hardware, conceptual architectures and frameworks, programming languages, and SDKs and APIs. Multipurpose mulsemedia systems are discussed and then compared. Finally, it concludes highlighting major gaps and shortcomings of current mulsemedia systems.

2.1 Background

2.1.1 Timeline and Event-based Mulsemedia Systems

To understand how typical mulsemedia systems work, Waltl, Timmerer, and Hellwagner (2009) presented an ordinary scenario of a *timeline-based* mulsemedia application. Firstly, the main media, such as a movie and its SEM file, are obtained from a physical media or online service. Then, a media processing engine acts to interpret the media resources, adapting the media, as well as the SEM, to the devices which will render the sensory effects. Also, user preferences are considered. Finally, the user environment is extended with devices (or actuators) capable of stimulating sensory effects such as vibration chairs, wind fans, scent emitters, and so on. Figure 1 represents this concept.

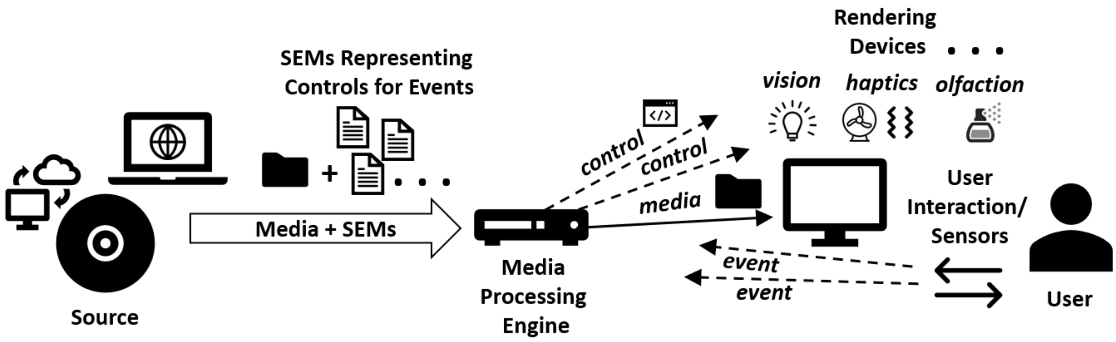
Figure 1 – Concept of timeline-based mulsemedia applications.



Source: Adapted by the author from (WALTL; TIMMERER; HELLWAGNER, 2009).

In addition to the previous scenario, Santos, Neto, and Saleme (2015) envisaged mulsemedia systems working with *event-based* multimedia applications. This type of mulsemedia system bears some similarity with timeline-based ones, but actuators are either activated by events that occur in the virtual world from user interactions or as a response to a stimulus from the real world captured through sensors by the virtual world (Figure 2). Instead of synchronizing continuous media with the actuators just once, event-based multimedia applications have a requirement for quick response time, that is, as soon as an event happens the media processing engine has to deliver it as fast as it can to the actuators so as to create more realistic immersive experiences.

Figure 2 – Concept of event-based mulsemedia applications.



Source: Created by the author.

2.1.2 Temporal Issues in Mulsemedia Systems

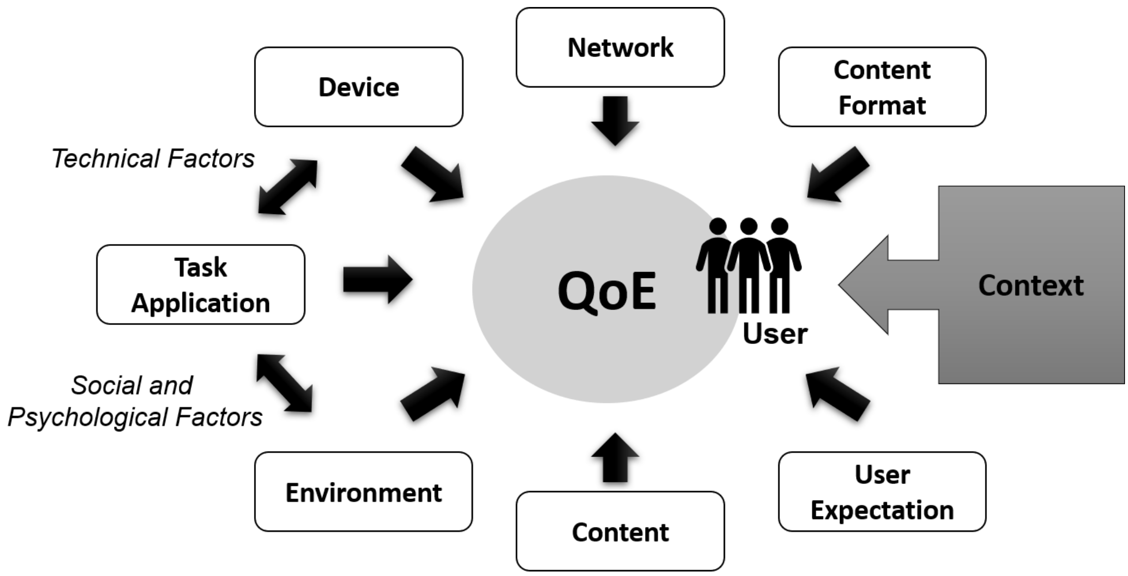
By introducing a network and considering sensory effects metadata processing time in mulsemedia systems, a set of hurdles emerge, such as network delay, jitter, packet loss, sensory effects metadata transmuting, timetable building, among others, which can have an impact on users' QoE (COVACI et al., 2018). As a result, timing issues within the integration of different applications to deliver sensory effects must be considered so that they do not affect performance, which would be detrimental to users QoE.

Tolerable delays for multisensory interactions have been identified in the literature. These times are quite strenuous to pinpoint once they might vary from one context to another. For instance, Nakamura, and Miyashita (2012) found out that electric taste can be presented with visual stimuli with a difference between [13ms, 837ms]. Murray et al. (2017) discovered different thresholds depending on the scent type, that is, foul scent [0s, +15s], spicy, fruity, flowery scents [-10s, 10s], and burning scents [-5s, +10s]. This was found out using the same experimental material and research instruments of earlier work (ADEMOYE; GHINEA, 2009) which identified olfactory/video sync tolerable delays of [-30s, +20s]. For haptic accompanied by videos, Yuan et al. (2015) indicated [0s, 1s] as a tolerable range. On the other hand, when haptic is presented with a head-mounted display, like the work of Adelstein, Lee, and Ellis (2003), system latency must be no higher than 17ms. As pointed out by Kim, Osgouei, and Choi (2017), users negatively notice even a 40ms delay in a touchscreen event-based application with haptic feedback. Indeed, haptic delay perception might even be different in discrete and continuous events in many applications depending on how the operator issues action commands and what information is fed back (RANK; SHI; HIRCHE, 2010) and researchers have studied how to deal with it (ANTONAKOGLOU et al., 2018). Regardless of the particularity of each case, it is clear that there is a concern about temporal aspects in mulsemedia systems that shall be taken into account.

2.1.3 Quality of Experience in Mulsemedia Systems

Ebrahimi (2009a) states that "there is no longer only a question of which features are included in a multimedia product or service, but also how well such features are addressed, and even more importantly, which impact they have on end-users." The quality of the presentation delivered to users may be affected by the technology as well as end-to-end issues such as delays in the interactions, usability, human factors, and context, as shown in Figure 3. Likewise, Brunnström et al. (2013) believe that QoE comes from the achievement of users' expectations with regard to utility, the level of enjoyment considering their personalities, and their current state, whereas QoS is focused on telecommunications and cope with physical systems performance. From this perspective, this section presents how mulsemedia experiences have improved QoE and evaluation methods to carry out similar research.

Figure 3 – QoE impact factors.



Source: Adapted by the author from (EBRAHIMI, 2009b).

2.1.3.1 Mulsemedia Experiences

This section describes the efforts of many researchers towards comprehending users QoE in mulsemedia systems (ADEMOYE; GHINEA, 2009; WALTL; TIMMERER; HELLWAGNER, 2010a; RAINER et al., 2012; YUAN et al., 2014; MURRAY et al., 2017; MONKS et al., 2017; JALAL et al., 2018b). They are highlighted next with a focus on experimental design (participants, materials, and setup) and results in order to understand how the experiments were carried out.

In a pioneering work, Ademoye, and Ghinea (2009) explored the temporal boundaries of olfactory multimedia applications from users' perspectives. Forty-two participants took part in their experiment, 14 females and 28 males, between the ages of 18 and 40. Participants watched six 90-seconds video-clips with a resolution of 240×180 pixels and associated with different smells. Each video was comprised of 3 temporal segments with different synchronization (olfaction ahead of audiovisual content, in-sync, and olfaction behind audiovisual content). A specific program to display the multimedia presentation with olfactory data was designed. The scents were emitted by Vortex Active, a smell dispensing system connected via USB to a computer. After collecting answers from a questionnaire on perception, data from participants indicated the presence of two main synchronization regions from -30s to +20s, with more tolerance to olfaction ahead of audiovisual content.

Waltl, Timmerer, and Hellwagner (2010a) invited 24 students (11 females and 13 males) between the age of 18 and 37 years to take part in a subjective experiment whereby the users watched two video clips (an action movie and a documentary). The environment was comprised of a computer with a monitor running SEMP (WALTL et al., 2013) and an amBX Premium Kit (fans, vibration bar, lights, sound). For each video clip, 4 versions with different bitrates were presented with and without sensory effects adding up to 16 sequences. The authors concluded that for each clip and bitrate version, the MOS (Mean Opinion Score) value is higher if annotated with sensory effects than without sensory effects. Furthermore, on average, sensory effects show an improvement of about 0.5 MOS (5-point scale) on average compared to video resources without sensory effects.

With the aim to provide sensory effects in a web-based application and assess user QoE in this context, Rainer et al. (2012) built a web browser plug-in and a library named AmbientLib. The authors carried out this research at 3 universities: AAU Klagenfurt, Austria; RMIT University, Australia; and UoW, Australia, and recruited 26 (18 females and 8 males), 21 (12 females and 9 males), and 21 (6 females and 15 males) participants respectively. They aged from 20 to 63. Fifteen video clips were presented to the users among action, news, commercial, documentary, and sports. Their setup included a computer with a monitor running AmbientLib 1.5 and Web browser plug-in 1.5 and an amBX Premium Kit (fans, vibration bar, lights, sound). The results show that, in general, sensory effects enhance the QoE, and action and sports genres have the highest improvement of QoE. Moreover, active emotions (i.e. surprise, fun, and worry) are increased in their intensity in the presence of sensory effects.

Yuan et al. (2014) assessed the impact of intensity of haptic and wind on QoE. Eighteen users (11 males and 7 females) participated in their experiment between 20–36 years. Twelve sequences from "Back to the Future" and "Jurassic Park" were selected with a resolution of

1280×720 pixels, frame-rate of 30fps, and an average bit-rate of 2500Kbps. For haptic effects, a vibration vest was set up, whereas a USB fan was used to generate airflow at different intensities. Roughly 70% of the participants reported that both haptic and airflow effects enhance the sense of reality and increase enjoyment. Furthermore, different levels of haptic intensity did not translate into significant enhancement, whereas reduced levels of airflow indicated reduced perception levels.

The study of Murray et al. (2017) analyzed QoE under the perspective of olfaction-based mulsemedia experiments. One study involved 86 participants and aimed at finding out the user perception from a single olfactory stimulus. They used the SBi4-radio v2 scent emitter controlled by the Exhalia Java-based SDK, VLC media player 1.0.1, and a software framework to control the presentation of olfactory data and video. Fourteen video clips and 11 different scent types were employed. Skew levels between the various media components were presented in step sizes of 5s, and they ranged from -30s to +30s in this experiment. They reported that participants who were stimulated by pleasant scent informed higher QoE levels. In addition, the authors found out that the participants preferred olfaction presented after the video as opposed to before with high ratings for enjoyment, relevance, and reality at skew levels of +10s.

A 3D vs. 2D mulsemedia experiment was conducted by Monks et al. (2017). Forty-four participants (37 males and 7 females) aged from 20 to 60 years took part in their experiment in an unrelated (between-participants) design. The environment included a gaming vest for haptic effects, an olfaction dispenser for olfaction effects, and a USB fan for generating airflow effects. Sixteen video clips were extracted from "Back to the Future" and "Jurassic Park" both in 2D and 3D video format. The authors concluded that the visual discomfort usually reported by users when wearing 3D glasses was decreased with sensory effects. In comparison with 2D-mulsemedia-based video clips, user QoE the sense of reality and enjoyment were higher with 3D-mulsemedia-based video clips.

Jalal et al. (2018b) assessed the quality of experience of users in a home entertainment scenario. The authors champion that one of the targets of the next generation of TV broadcast services is to provide realistic media content to the users. The environment assembled by the authors included a TV, a desktop PC, three RGB Smart Philips LED light devices for lighting effect, an air conditioner for airflow, a smartphone for vibration effects, and the PlaySEM platform (SALEME; SANTOS, 2015) to play the videos and to render sensory effects. The assessment involved 40 participants (31 males and 9 females, between 22–50 years old), 2 per session, to give feedback over 10 video clips of 30-40 seconds enriched with mulsemedia. The results pointed out that 85% of the users agreed that sensory effects increase the sense of reality in the experiment, 67.5% were not distracted by the sensory effects, 80% enjoyed the experience with mulsemedia, and 70% judged the timing of the sensory effects appropriated.

With regard to objective evaluations in mulsemedia experiences, the work of Egan et al. (2016) combined heart rate and electrodermal activity monitoring to subjective questions. They correlated the results and found out that high values of these objective metrics were associated with physiological arousal. Keighrey et al. (2017) also showed the potential and benefits of using these objective metrics as indicators of user QoE for immersive experiences in AR applications. Thereby, physiological devices can be useful in effective state monitoring and are a valid way to gather sometimes concealed data about the experience. Complementary to subjective assessments, these objective evaluations have the potential to bring revealing insights.

2.1.3.2 Evaluation Methods

Investigating QoE involves capturing users' level of satisfaction or boredom whilst engaged in an application or service in computers. In fact, this is not all plain sailing because QoE ranges from technical aspects (e.g. devices, content format, and network) to psychosocial factors (e.g. environment, content valence, arousal, expectation, and current emotional state). QoE has been assessed by either performing subjective surveys and objective evaluations, such as those reported in Section 2.1.3.1. Additionally, technical recommendations have been used together, such as ITU-R-BT.500-13[1] (Methodology for the subjective assessment of the quality of television pictures), ITU-T-P.910[2] (Subjective video quality assessment methods for multimedia applications), ITU-T-P.913[3] (Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment), and ISO 8589:2007[4] (Sensory analysis - general guidance for the design of test rooms).

Users' QoE assessment is undoubtedly time and effort demanding. However, there has been some guidance in the literature, notably the works of Rainer, and Timmerer (2014) and Murray et al. (2017b). In a nutshell, Rainer, and Timmerer (2014) provide the following steps in order to carry out subjective evaluations:

1. Introduction - it describes the experiment to the user including how to rate the experience;
2. Pre-questionnaire - it is used to collect demographics;
3. Main evaluation - it includes training users and collects their perceptions;
4. Post-questionnaire - to know whether users have participated in similar subjective evaluations.

A detailed and stepwise tutorial/guide, but focused on olfactory-based mulsemedia experiences, is presented by Murray et al. (2017b). Their work includes a comprehensive study of approaches for QoE evaluation, including aspects such as methods, environment, types of scents, length of the experiment, quantity, and balance of participants. Two important recommendations

[1] Recommendation ITU-R-BT.500-13 available at: <https://www.itu.int/rec/R-REC-BT.500-13-201201-I/en>
[2] Recommendation ITU-T-P.910 available at: <https://www.itu.int/rec/T-REC-P.910-200804-I/en>
[3] Recommendation ITU-T-P.913 available at: <https://www.itu.int/rec/T-REC-P.913-201603-I/en>
[4] ISO 8589:2007 - Sensory analysis - available at: <https://www.iso.org/standard/36385.html>

that they provide in mulsemedia assessment encompass "performing assessment in controlled and known conditions with minimum distraction" and "reducing physical condition and psychological factor effects on human judgment." The authors also include thorough proposals for participants' trial and training, physical environments and experimental design, and methods.

In relation to the type of assessment for mulsemedia systems, it will depend mostly on the research question and hardly on the way the environment is built. In objective assessments, though, the employed equipment should be adapted accordingly. For instance, an eye-tracker for monitoring eye gaze on screens should be different for VR goggles.

Evaluating QoE in mulsemedia is not a straightforward task. A great deal of time must be employed to arrange the environment for the experience, which involves not only setting up the devices but also the creation of mulsemedia content. Taking into account that other researchers might be interested in shortcutting this time, Waltl et al. (2012) made available an extensible mulsemedia dataset[5] to be used in different setups. They gathered 76 video clips with different lengths from varied genres, including action, documentary, sport, news, and commercials, and annotated them with MPEG-V to provide wind and vibration effects. Another noticeable mulsemedia dataset is reported by Murray et al. (2017a). With the aim of making research reproducible and allowing researchers to follow unpaved ways on the same data, the authors collected and made available a mulsemedia dataset[6]. A total of 6 video clips of 90-second length were annotated with olfactory effects. The genres included cookery shows, news, and documentary associated with the following categories of smell: burnt, flowery, foul, resinous, spicy, and fruity. The data was written in text format separated by commas. Information about the test environment, as well as employed research methods, are also described in the work.

2.1.4 Multisensory Devices

Multisensory environments can be deployed by using devices that stimulate various senses at the same time. To this end, a variety of technological elements can be used to construct a multisensory environment. These components are mostly used in academic settings, although, recently, the industry started to be interested in building multisensory environments. As multisensory devices are of paramount importance for mulsemedia systems, this section provides a background of them by summarizing the most recent commercial and prototype devices from the work of Saleme et al. (2019a), especially to display the heterogeneity of hardware that delivers sensory effects.

2.1.4.1 Hardware for Haptic Effects

Haptic technology refers to everything a user touches or is touched by to control or interact with an entity controlled by a computer. Some of these interfaces are energetically passive

[5] Sensory Experience Lab's dataset available at <http://selab.itec.aau.at/software-and-services/dataset>
[6] Murray's dataset available at <http://www.niallmurray.info/Research/appendix>

(a button, a keyboard), whilst some are energetically active (force feedback devices, vibrotactile vests). The techniques and the key challenges characteristic to this medium are discussed in detail in (DANIEAU et al., 2013)—a comprehensive survey that presents technologies and examples for enhancing audiovisual content with haptics. The focus in this section though is on active devices that deliver sensory effects to users.

Force feedback gears (that consist typically of vibrotactile actuators embedded into clothes) and suits already have an established business within the area of wearables haptic devices. In the 90s, Auralizer created a system whereby audio waves were converted into vibrations. Likewise, haptic gears such as those presented by Shah, Basteris, and Amirabdollahian (2014) and Prasad et al. (2014) have been applied in HCI to provide feedback on impact and serve as an aid for motorcyclists. This kind of gear was also used as a guide so that robots can steer humans in cooperative work (SCHEGGI; AGGRAVI; PRATTICHIZZO, 2017). A vibrotactile vest produced by KOR-FX[7] fits in this category and uses a simplistic approach to transform audio signals into haptic feedback. The audio signal coming from games or media is processed and converted with special transducers into pinpointed high-definition vibrotactile feedback that allows users to feel the on-screen action. Subpac 101[8] is another haptic vest conceptually akin to KOR-FX as mechanism and price. An extra version whereby the equipment can "wear" an existent seat is also ready for use. ARAIG (As Real As It Gets)[9] produces feedback on numerous degrees by incorporating speakers in a collar to create a surrounding effect around the user. Moreover, the user's experience is intensified with vibration and audio feedback, and electrical stimulation by flexing particular muscles and reproducing sensations of touch. The Tesla suit[10] is a full-body neoprene suit with "conductive threads that tricks the senses using neuromuscular electrical stimulation." The Tesla suit promises to create "a range of tactile sensations" including vibrations and thermal ones. To do this end, it has several actuators spread through the body to provide comprehensive haptic feedback. Dexmo[11] is an exoskeleton glove for VR developed by (GU et al., 2016). Apart from capturing motion, this product also offers force feedback.

Vibrotactile mice and joysticks are often used as portable devices through which users experience haptic feedback. One of the first haptic mice to be developed and explored in virtual environments was that of the EU MUVII (Multi-User Virtual Interactive Interface) project[12]. The gaming industry is constantly using vibrotactile technology to enhance immersion in video games with examples like the Rival 600 from Steel Series[13] and the Joy-Con from Nintendo[14], which contains an advanced haptic feedback mechanism called "HD Rumble." The controller is composed of actuators that provide users with feelings of touching objects. Another

7 KOR-FX available at <http://www.korfx.com>
8 Subpac 101 available at <https://subpac.com/subpac-101/>
9 ARAIG available at <https://araig.com>
10 Tesla suit available at <https://teslasuit.io>
11 Dexmo available at <https://www.dextarobotics.com/en-us>
12 MUVII project available at <https://cordis.europa.eu/project/rcn/57839/factsheet/en>
13 Rival 600 available at <https://steelseries.com/gaming-mice/rival-600>
14 Joy-Con available at <https://www.nintendo.com/switch/features/>

handheld device, Windy Sight Surfers (RAMALHO; CHAMBEL, 2013), is "an interactive mobile application for the capture, visualization, and navigation of 360° immersive videos." It has a wind accessory composed of two fans attached to a tablet, which presents 360° content. Despite being a prototype, the authors showed that this system can elevate immersion and presence. Another endeavor is Haplet, which is "an open-source, portable and affordable haptic device with collocated visual, force and tactile feedback" (GALLACHER et al., 2016). This device is based on Hapkit and Haptic Paddle, which present a system for creating haptic effects from 1 degree of freedom device (MORIMOTO; BLIKSTEIN; OKAMURA, 2014). It allows users to combine their devices with haptic feedback effects. The authors state that "this design can replicate the natural way in which we use our hands to interact with the physical world." In (WHITMIRE et al., 2018), the authors propose a handheld virtual reality controller that renders fingertip haptics. This consists of an interchangeable wheel that moves in relation to its position in the virtual environment. In (BENKO et al., 2016), the authors present NormalTouch and TextureTouch - two controllers that use different actuation methods to render haptic 3D shape. However, these present limitations in rendering angles, forces, and heights. Tactile effects were also obtained via finger-mounted haptic feedback devices. They convey cutaneous force information by deforming the skin on the fingertips (SCHORR; OKAMURA, 2017).

When it comes to desktop setups, devices like Novint Falcon, Phantom Omni or Ultrahaptics are the most popular and easiest to integrate into diverse systems. Novint Falcon was often used in research with different applications: to enhance educational videos (KIM et al., 2010) or to touch images in the video (CHA; EID; SADDIK, 2009), whilst Phantom Omni was employed to enable users to feel the acceleration associated with videos (DANIEAU et al., 2012). Ultrahaptics is another commercial haptic device that employs "focused ultrasound to project discrete points of haptic feedback on user's hands" (CARTER et al., 2013). This has been successfully integrated with HoloLens in designing mixed reality human-computer experiences, as described by Kervegant et al. (2017). Ultrahaptics showed promising results in respect of mid-air interactions in cars, decreasing the eyes of the road time, whilst not compromising the driving performance (SHAKERI; WILLIAMSON; BREWSTER, 2018).

Wind devices are a particular case of haptic hardware in which the sensory effect is obtained by generating airflow, which brushes against human skin. The work of Moon, and Kim (2004) brought an early attempt to create surrounding wind in the user's environment. Following this approach, VirWind[15] tries to create a 3D effect in the environment blowing air from four vertical poles composed of four fans each one.

Another class of vibrotactile devices is that of haptic chairs. Feel Three[16] consists of a 3DOF motion simulator. It was first created by Kumagai (2010) and then evolved to its current state. A half-sphere platform composed of a set of motors and omnidirectional wheels

[15] VirWind available at <https://www.vrfocus.com/tag/virwind/>
[16] Feel Three available at <http://www.feelthree.com>

is responsible for producing motion effects, including pitch, roll, and yaw. Roto VR[17] is a platform-based interface that promises to transform the traditional seated VR set-up into a totally immersive endlessly revolving experience - complete with motorized turns, no tangling cables, and a double rumble effect. To some degree, it takes after the concept of Haptic ChairIO (FENG et al., 2015). The Roto VR is designed to make VR experiences even more immersive whilst reducing the effects of simulator sickness. Turning your head will activate the motors in the base, while controls located at the players' feet enable movement.

2.1.4.2 Hardware for Olfactory Effects

Until now, digitally-controlled scent devices have been used in a variety of applications: for enhancing the QoE in multimedia and mulsemedia applications (ADEMOYE; GHINEA, 2009; MURRAY et al., 2013b; YUAN et al., 2014; YUAN; GHINEA; MUNTEAN, 2015), for augmenting the immersion in entertainment/training virtual reality applications (ISCHER et al., 2014; HOWELL et al., 2016), for studying its potential in e-learning (ADEMOYE; GHINEA, 2013), for studying the connection between smells and autobiographical memories (CHU; DOWNES, 2000) or for analyzing what moods or emotions are triggered by smells (RÉTIVEAU; IV; MILLIKEN, 2005; SARID; ZACCAI, 2016), or indeed whether olfactory congruence matters in mulsemedia (GHINEA; ADEMOYE, 2012).

Olfactory stimulation is achieved mostly by using "analog" methods from fragranced shampoos (PORCHEROT et al., 2010), cylindric felt-tip pens (PICHON et al., 2015), ambient odors (TOET; SCHAIK; THEUNISSEN, 2013), odorant stimuli provided by Firmenich (DELPLANQUE et al., 2012), as well as smelling jars (FUCCIO et al., 2016; COVACI et al., 2018). Odor materials are generally stocked in liquid or solid structure - in the case of the latter, mostly wet with liquid. To deliver the scents, these stored materials need to be conveyed to the user's nose through the air. According to Yanagida (2012), computer-controlled olfactory devices achieve this in several ways: "natural vaporization, vaporization accelerated by airflow, heating or atomization."

In (YAMADA et al., 2006), the authors presented a wearable olfactory device for olfactory stimuli according to the position of the person. Based on spatial localization sensors, this device was used to create an odor field in a virtual reality space. Another interesting system was proposed by Yanagida et al. (2003). This olfactory device consists of "a nose tracker and a scent projector composed of an air cannon, a scent generator, and a 2 degrees-of-freedom platform that is controlled so that the air cannon aims just under the user's nose." In (NAKAMOTO et al., 2008), the authors addressed the limitation of the gas-based scents in olfactory devices by developing an apparatus that deals with liquid odor. They built a system capable of real-time scent blending, and, based on it, they developed a cooking game to evaluate any change in presence experienced by the participants. In (MATSUKURA; YONEDA; ISHIDA, 2013), a

[17] Roto VR available at <https://www.rotovr.com>

new type of olfactory system was introduced. In this case, the scent was distributed to the user through four ventilators that were fixed on the corners of the screen. This showed potential for further development of novel interactive multimedia systems, but it has as the main drawback the fact that it cannot generate multiple scents simultaneously. Although the authors provide significant proof of work for all the above devices, the development steps are not described in detail to allow for replication by other researchers.

SBi4[18] from Exhalia uses airflow to vaporize and delivers (by default) one of four fragrances at the time. In (MURRAY et al., 2014), the authors stated that SBi4 is "more reliable and more robust than the other devices on the market" and the scents are more realistic. SBi4 was used in numerous studies that investigated the QoE in desktop systems enhanced with olfactory content (MURRAY et al., 2013a; MURRAY et al., 2017; MURRAY et al., 2014; ZOU et al., 2017). However, there are some considerations that researchers need to keep in mind when working with this olfactory device detailed in (MURRAY et al., 2014). Another option from Exhalia is uScent Collection[19] that delivers odors in rooms of different size (depending on the model). These devices work with one cartridge, and they are programmed remotely using the platform[20] provided by the developers. An ultrasonic USB essential oil diffuser called "The Keylia"[21] is offered by Aroflora. As its name suggests, this device diffuses essential oil, operates at intervals of 10, 30, or 60 seconds, and starts emitting the aroma as soon as it is connected to the USB port of any kind of machine supporting USB. Olorama[22] is another technology that could offer researchers new ways of integrating the sense of smell into their projects. This solution combines hardware, software and essential oils in the synchronization of audiovisual scenes with scents. The wireless olfactory device fits both a small room or a big cinema and uses airflow to vaporize only one odorant cartridge at a time. Developers promise a simple and quick integration and provide Unity and Unreal code as an example.

It is remarkable that a number of papers have been written to propose reproducible olfactory systems, thus benefiting a larger part of the research community. Addressing the limitations of olfactory research in immersive virtual environments, Herrera and colleagues (HERRERA; MCMAHAN, 2014; HOWELL et al., 2016) presented an effective and affordable desktop olfactory device that relies on vapor to deliver smell effects. The authors used affordable components (the device is estimated to cost US$ 55) and provided detailed information about the design process and the software used to control the olfactory device, that could easily be replicated by other researchers. Hajukone is another open-source low-cost olfactory design, this time in a wearable format (MCGOOKIN; ESCOBAR, 2016). It was built as an alternative to research devices that are not presented in full detail to allow reproduction. Thus, it makes use of electronic elements that are fairly easy to find in the market. As opposed to the device

18 Exhalia SBi4 <http://www.exhalia.com/fr/>
19 uScent available at <http://www.exhalia.com/us/produits/espaces-olfactifs/uScent/>
20 i-Scent available at <http://i-scent.fr/login>
21 The Keylia available at <https://bit.ly/2SmjG1o>
22 Olorama available at <http://www.olorama.com/en/>

described in (HERRERA; MCMAHAN, 2014; HOWELL et al., 2016), Hajukone supports multiple scents that are emitted through ultrasonic transducers. InScent (DOBBELSTEIN; HERRDUM; RUKZIO, 2017) is a "miniaturized open-source wearable olfactory device that allows users to receive personal scent notifications." Similar to Hajukone, it allows replicability through 3D printing. At only 102g, inScent has 8 cartridges, each of them containing scents to deliver over 70 "scentifications." Amores, and Maes (2017a) describe the development of a prototype that users can wear called "Essence". The aim was to create an attractive and light olfactory device for applications that can deliver different strengths of smell related to the user's biodata. This work is further expanded to "Bioessence", a device that can be attached to the user's clothes in a form of clip or necklace (AMORES; MAES, 2017b). It can release the limit of three scents and passively captures vibrations representing the beating of the heart and the respiration through clothes. Salminen et al. (2018) present an "olfactory display prototype for emitting and sensing odors." They used an intersurgical mask attached to a VR headset that covers part of the user's face. It was then connected to a vent hole that comes from an aromatized container or to a device to receive scents. Hasegawa, Qiu, and Shinoda (2018) depict a system to control the spatial distribution of aromas through an ultrasound-driven approach, guiding a vaporized scent to the user's nostrils. This technique could be useful not only in this particular case but also for removing remaining odors while presenting multiple olfactory experiences sequentially.

Despite media excitement, most of the olfactory devices launched thus far are proof-of-principle prototypes. Although it seems hard to convince users that digital olfaction is desirable, a potential explanation behind the restricted prosperity of this technology is the lack of correlation between hardware and software developers and interaction experts. The work put in developing these devices is often not detailed; thus, it cannot be reproduced by third parties. Whilst the dialogue between these stakeholders will undoubtedly intensify when a mulsemedia killer app is found (GHINEA et al., 2014), this does not preclude undertakings in these areas, as shown in the literature.

2.1.4.3 Hardware for Gustatory Effects

Authentic tasting experiences can be created once we activate the sense of taste, retronasal olfaction, and trigeminal nerve (SPENCE et al., 2017). However, this is very challenging because it implies stimulating all the senses in the right way, with an intensity that feels natural. Tastes and flavors are complex because most of them cannot yet be generated by stimulating the human palate directly on the tongue, which is able to detect at least the controversial five basic tastes (sweet, sour, bitter, salty, and umami). Other things that surround the tasting experience (e.g. the roasted, the fruity) are related to smell. Sensations of heat (e.g. hot pepper), cold (cool associated with mint), and several food properties such as crunchiness and creaminess, are detected by the trigeminal sense (SPENCE; PIQUERAS-FISZMAN, 2016).

In terms of devices and systems that stimulate the taste and could be included in multisensory systems, some deal with direct stimulation of this sense, and some modify people's experience of taste by stimulating other senses like vision and olfaction. Although the latter could be considered a part of a gustatory-based mulsemedia system, the former is the focus in this section, which aims at making evident gustatory hardware heterogeneity.

As described in (BUJAS et al., 1974; PLATTIG; INNITZER, 1976), basic tastes have been delivered by actuating on the tongue in order to stimulate people's palate. Recently, progress in this area has been achieved with studies like (WEI et al., 2011; MURER; ASLAN; TSCHELIGI, 2013; RANASINGHE; DO, 2017).

Wei et al. (2011) created CoDine, a dining table system that "augment and transport the experience of communal family dining to create a sense of coexistence among remote family members." It does not produce taste from electrochemical as others do, but it has a submodule called Food Teleportation, which is a prototype, that writes personalized edible messages on the food and delivers it to the users using a robotic carriage.

Lollio (MURER; ASLAN; TSCHELIGI, 2013) has been proposed as a novel interaction method within a game and was built to interact with the user's tongue by pumping specific tastes from a portable and small box to the tip of a lollipop. Its development is described in detail, allowing for replication. One of its limitations is that it delivers taste sensations only on a sweet-sour interval.

Digital Lollipop is another experimental instrument that digitally simulates tastes by electrical stimulation of the taste-buds, described in detail by its authors (RANASINGHE; DO, 2017). More complex than Lollio, it reports taste sensations additional to sweetness and sourness, such as saltiness and bitterness, and also proposes a way to control the intensity of sourness. The authors tested their solution in experimental tests, whereby they made significant observations: the interface was uncomfortable over certain values of the current intensity, it was challenging to align the device on the user's tongue, and the subjective opinions provided by participants highlighted that some users were not able to recognize certain taste sensations. Participants' feedback indicated portability and its enhancement with smell emissions were directions in which the device could be improved.

A gustatory device created by Karunanayaka et al. (2018) called "The Thermal Taste Machine" produces the effect of tastes by varying the temperature, in bouts, on the user's tongue. The authors reveal that creating and altering the feeling of tastes for "sweet, chemical, minty, fatty, pleasantness, heating, and cooling" had favorable outcomes. Although the design and development processes are presented in detail, building these types of interfaces requires high expertise in the field.

In related work, Vi et al. (2017) devised TastyFloats, a machine where small pieces of food are levitated acoustically and delivered on the user's tongue. As the authors recognize, this system has many issues to be solved before it appears as a steady product, mainly related to speed and quantity. Moreover, the user's environment conditions, temperature, and characteristics of the food also need to be taken into account.

As most of the gustatory devices are DIY (Do It Yourself) hardware encountered in the literature, little information related to the availability of their software is provided.

2.2 Related Work

2.2.1 Conceptual Architectures and Frameworks, and Programming Languages

The works of Suk, Hyun, and Yong (2009), Choi, Lee, and Yoon (2011), and Yoon (2013) are all endeavors to promote architectures and frameworks for delivering sensory effects. A device inter-locked media service framework and its technology for media generation and a media controller are proposed by Suk, Hyun, and Yong (2009). Their framework is a conceptual model where they outline the process of creating, packing, and transmitting SEMs synchronizing devices with media. This proposal is rather similar to the description of Waltl, Timmerer, and Hellwagner (2009) on how mulsemedia systems work. However, it does not address event-based multimedia applications and lacks practical implementations of the whole proposal.

Choi, Lee, and Yoon (2011) present concepts and guidelines for a broadcast-based framework for streaming service with sensory effects to bring 4-D entertainment for homes, relying mainly on the MPEG-V standard. However, the authors do not show the implemented system. Moreover, whereas MPEG-V introduces some standardization on their conceptual framework, it does neither consider accommodation of future standards nor event-based multimedia applications. Likewise, Yoon (2013) suggests technologies to be used for delivering sensory effects in a home environment, including MPEG-2 TS (Transport Stream), MPEG-V, and UPnP. The author seems to extend what is described in (CHOI; LEE; YOON, 2011) and first presents concerns with regard to performance, suggesting compression for SEM. Nevertheless, the limitations remain the same as in (CHOI; LEE; YOON, 2011).

Sulema (2018) proposes a programming language for effective processing of multimodal data in order to allow the development of mulsemedia applications for several areas, including education, health, among others. It resembles a declarative language to present mulsemedia content. The constraint though is that there is no real system to evaluate the development of mulsemedia systems from this programming language. Another curb is related to support for event-based multimedia applications.

A new concept application referred to as $360°$ Mulsemedia envisaging a conceptual Mulsemedia Delivery System for 5G networks is introduced by Comsa, Trestian, and Ghinea (2018). They propose to encode the videos embedding sensory effects into each frame, which in turn, has a matrix of intensities of scent intensities according to the user's viewport. Although they introduce a groundbreaking proposal, no standardization is used to annotate the videos and does not account for event-based multimedia applications.

2.2.2 SDKs and APIs for Haptic, Olfactory, and Gustatory Effects

Unlike mulsemedia software systems, which directly deliver multisensory experiences to the users, SDKs and APIs in the context of mulsemedia provide means to develop applications that support some type of sensory effect from some sort of computer application. In a nutshell, APIs offer interfaces for software to cooperate with other software whereas SDKs include a set of tools such as libraries, documentation, samples, and so forth, to enable the development of software.

2.2.2.1 Haptic Software

Whilst many commercial haptic devices such as Feel Three, Roto VR, Dexmo and Tesla Suit (SALEME et al., 2019a), to name a few, have their own SDK to control their devices, there has been an upsurge of open-source solutions to deal with haptics over the past years. This type of interface can be used as either an input device or an output device. Even though a myriad of software solutions to control haptic hardware has been created, the scope of study in this section is concentrated on software that handles devices that generate artificial stimuli to users, that is, SDKs and APIs for tactile and kinesthetic devices.

Novint Falcon, a haptic device that offered an SDK limited to Windows (MARTIN; HILLIER, 2009), has contributed to foster the development of many open cross-platform plugins, libraries, SDKs, and API to control it. A specific library named libNiFalcon[23] was developed for this purpose. Many other solutions for haptics have included supported to Novint Falcon, such as JTouchTool[24], Haptik Library[25], CHAI 3D[26] (CONTI, 2003), HAPI[27] from H3DAPI, OpenHaptics[28] (ITKOWITZ; HANDLEY; ZHU, 2005), etc. Most of them have supported connection to other popular haptic devices, including SensAble PHANToM devices, providing a layer of abstraction as well as the commercial SDK Immersion's TouchSense[29].

JavaScript solutions have also been developed to provide haptic interaction for web-based applications. P4A Haptic Toolkit[30] (KAKLANIS; VOTIS; TZOVARAS, 2015), JHaptic library[31] and Haptics.js[32] are efforts to deliver vibrotactile effects providing a cross-browser compatibility

[23] libNiFalcon available at <https://github.com/libnifalcon>
[24] JTouchTool available at <https://github.com/IanJohnArcher/JTouchToolkit>
[25] Haptik Library available at <http://sirslab.dii.unisi.it/haptiklibrary/>
[26] CHAI 3D available at <https://github.com/chai3d/chai3d>
[27] HAPI (H3DAPI) available at <http://www.h3dapi.org/>
[28] OpenHaptics available at <https://www.3dsystems.com/haptics-devices/openhaptics>
[29] Immersion's TouchSense SDK available at <https://www.immersion.com/technology/#touchsense-technology>
[30] P4A Haptic Toolkit available at <https://github.com/NickKaklanis/WebHapticModule>
[31] JHaptic library available at <https://github.com/guari/jhaptic>
[32] Haptics.js available at <http://www.hapticsjs.org/>

layer. They do not deal with hardware level, therefore, it might be required an SDK for the former and compatibility between browser and hardware for the latter.

Other efforts to provide access to haptic devices have been found in the literature. SimHaptics[33] (SANFILIPPO; WEUSTINK; PETTERSEN, 2015) is an open-source library to deliver haptic feedback that is compatible with devices produced by Force Dimension. To be applied in the field of training in medicine, Virtual Surgery SDK was created by Kolsanov et al. (2016). According to the authors, it provides realistic force feedback and allows the possibility to reuse its components to create other solutions. Haplet (GALLACHER et al., 2016) provides APIs in C++ and Python to interact with computers and tablets. Both hardware and software solutions are open-source[34]. HPGE (Haptic Plugin for Game Engines) (BALZAROTTI; BAUD-BOVY, 2018) was developed to provide haptics for game engines. It is based on CHAI 3D (CONTI, 2003) and focuses extensively on Unity3D.

2.2.2.2 Olfactory Software

As reported by Murray et al. (2017b) and Howell et al. (2016), there have been developed SDKs and APIs for commercial olfactory devices, such as Cyrano (for iOS devices), Scentee (for Android and iOS), Exhalia SBi4 and Dale Air Vortex Activ (Windows), and Olorama (unknown). The caveat is that these software solutions cannot be used with other devices. In other words, their business model relies on selling their hardware accompanied by software to make them work, not allowing reuse when an olfactory device is replaced. Conversely, McGookin, and Escobar (2016), Howell et al. (2016), and Dobbelstein, Rukzio, and Herrdum (2017) have created open solutions that allow some degree of reuse.

In (MCGOOKIN; ESCOBAR, 2016), Hajukone[35] is presented as an open device for olfactory experiences. It was designed so that it could be reproducible by human-computer interaction researchers in this domain. Although it is focused on the device itself, its Arduino-based code is open as well.

The Arduino-based olfactory solution described by Howell et al. (2016) is another effort to make an affordable olfactory device that can be reproduced by researchers. Akin to Hajukone (MCGOOKIN; ESCOBAR, 2016), the work focuses mostly on the olfactory device. Likewise, there is a module written for Arduino that controls a small computer fan that delivers airflow for vaporization and scent delivery.

Dobbelstein, Rukzio, and Herrdum (2017) devised inScent[36]—a wearable olfactory device that notifies users when receiving messages on their smartphones. The software that controls the device is also open-source, according to the authors. This device uses an Arduino-

[33] SimHaptics available at <https://github.com/filipposanfilippo/SimHaptics>
[34] Haptlet available at <http://crgallacher.com/haply-project-open-source-haptics/>
[35] Hajukone available at <https://github.com/davidmcgookin/Haju>
[36] inScent available at <https://www.uni-ulm.de/?inscent>

based microcontroller called BLE Nano, that allows wireless communication between the smartphone and the olfactory device. A framework is provided for smartphones to reach the device. An interface must be implemented to send messages to the Arduino module.

2.2.2.3 Gustatory Software

Still in its infancy due mainly to its complexity (SALEME et al., 2019a), there has been little work on taste for mulsemedia systems. The works of Ranasinghe and colleagues (RANASINGHE; CHEOK; NAKATSU, 2012; RANASINGHE et al., 2017, 2017), Karunanayaka et al. (2018), and Vi et al. (2017) introduce some initiatives to deliver gustatory experiences. However, they do not provide either SDKs or APIs. An exception is an open-architecture to deliver taste synchronized with a functional magnetic resonance imaging system created by Canna et al. (2019). They developed an open software based on Arduino to start and stop the delivery of gustatory from a device they have devised. Although it has a specific purpose, their Arduino controller module named "Gustometer"[37] and device can be reused with other systems.

2.2.3 Multipurpose Mulsemedia Systems

Created by Cho (2010), Sensorama gives some flexibility to use SEM in timeline combined with a list of events that can be triggered on demand. On the one hand, it allows the presenter of the mulsemedia content to deliver dynamic effects in the user's environment. On the other hand, its static list of predefined sensory effects hinders its expansion. Apart from the other solutions, the Sensorama system is aimed at 4D movie theaters integrated to actuators, which are part of a CAVE (Cave Automatic Virtual Environment). It supports light, wind, fog, flash-light, and vibration effects through robust devices.

SEMP[38] (WALTL et al., 2013) is a player capable of reproducing multimedia content with sensory effects annotated in MPEG-V. It supports the devices amBX Gaming PC peripherals to deliver light, wind, and vibration and Vortex Activ to disseminate aromas in the user's environment. Furthermore, it offers a poll interface so that users can give feedback about their experience. According to the results presented by its authors, users pointed out an increase in QoE when exposed to the system coupled with the aforementioned devices. Particularly, intensifying the feelings of joy, fun, and worry and decreasing others such as boredom. Code to deal with sensory effects is embedded in SEMP, and it does not intend to deal with event-based multimedia applications. Moreover, support for additional hardware implies changing its source-code.

Based on the idea of reusing the infrastructure of sensors and actuators of a modern car, Kim, and Joo (2014) developed the Sensible Media Simulator. Despite being a simulator, it does not avoid coping with physical devices to simulate the sensory effects and includes a LED lamp, a fan, a fan with a heater to simulate warmth, and a vibrator. The simulator has a web interface

[37] Gustometer available at <https://github.com/antocanna88/gustometer>
[38] SEMP available at <http://sourceforge.net/projects/semediaplayer/>

based on the proprietary technology Flex from Adobe, which runs within web browsers. This idea comes from the purpose of allowing portability, that is, the system is able to run on different browsers. It also has independent modules for controlling different parts of the application, such as presentation and processing module. However, its architecture does not allow the use of other applications as presentations. The reason for this is not to deal with communicating with devices not supported.

Luque et al. (2014) devised the Media Processing Engine (also called Receiver Gateway). They designed and implemented a solution that integrates sensory effects to a hybrid (internet-broadcast) television system. The SEM is coded according to the format defined by the KNX (an abbreviation of Konnex) protocol, which is part of the KNX system, a bus system for building control able to exchange data via a common bus network. Data is transmitted via Wi-Fi to the receiver. The system is programmed to control a fog machine, a scent vaporizer, and an ambient light LED strip. It takes into account standardization and extensibility to some extent, but it is not designed to be reused with event-based multimedia applications. As its source-code is not available, analysis on support for new hardware is restricted.

A Multimedia-Multisensorial Platform is also presented by Bartocci et al. (2015). The idea is to create a platform to present mulsemedia by using two different strategies to transmit the media (MPEG-2 TS and IP–Internet Protocol—over the network) and a hardware controller called MCU (Microcontroller Unit) to perform the conversion of sensory effects described in MPEG-V into commands for the physical devices like the Microcontroller module. However, it has some limitations, i.e. it does not allow the reuse of the MCU with other multimedia applications. The devices are capable of producing olfactory and thermal effects.

PlaySEM is mulsemedia platform compatible with MPEG-V presented in 2015 by Saleme, and Santos (2015). The platform is composed of three main decoupled components (i) the SE Video Player[39], (ii) the SER (Sensory Effects Renderer) 1.0.0[40], which processes MPEG-V metadata and prepares commands to control the devices, and (iii) an Arduino-based microcontroller module, responsible for receiving the commands and driving different actuators. This very early version of PlaySEM SER was then evolved to its 1.1.0 version[41] to be used not only by the SE Video Player but also by several different applications, from multimedia players to any event-based application such as games, VR/AR software, and interactive applications (SALEME; CELESTRINI; SANTOS, 2017; SALEME; SANTOS; GHINEA, 2018). Despite expanding its versatility, it had some limitations further explained in Chapter 4 (Section 4.1).

Jalal et al. (2018a) proposed an IoT-based architecture for mulsemedia delivery to TV users in a home entertainment scenario in which they used not only the aforementioned PlaySEM SER (SALEME; SANTOS, 2015) but also the PlaySEM SE Video Player. The solutions

[39] PlaySEM SE Video Player available at <https://github.com/estevaosaleme/PlaySEM_SEVideoPlayer>
[40] PlaySEM SER 1.0.0 available at <https://github.com/estevaosaleme/PlaySEM_SERenderer/releases/tag/1.0.0>
[41] PlaySEM SER 1.1.0 available at <https://github.com/estevaosaleme/PlaySEM_SERenderer/releases/tag/1.1.0>

were placed respectively in the Aggregation Layer and Application Layer of their IoT-based architecture, which is composed of two more layers: Virtualization, where they used another set of hardware in conjunction with PlaySEM SER and its Arduino-based microcontroller module, and Physical, where the devices are placed. This approach started making evident the reuse of PlaySEM SER, albeit it had the same restrictions to deal with variability in a broader sense whereby applications and devices can be seamlessly swapped.

Another solution to deliver sensory effects for home entertainment is depicted by Lin, Yang, and Lin (2018). Akin to the proposal of Jalal et al. (2018a), the authors advocate for an IoT-based architecture where they combine IoTtalk Server, to control the rendering devices, with Video Service Platforms, through TheaterTalk. Applications for devices shall be created to handle them and configured on IoTtalk Server. Changeability and extendability are taken into consideration through this strategy. However, their work does not account for SEM standardization, requiring designers to annotate videos using their own authoring tool that produces content compatible with their system. This could eventually hinder compatibility between content and applications. Moreover, although it works with different video services, the use of their system with event-based multimedia applications is not addressed.

Table 1 presents a comparison between multipurpose mulsemedia systems. *Timeline-based applications functionalities* and *Event-based applications functionalities* indicate whether the system provides support for these types of interaction. *Reusable SER* reveals if the system allows the reuse of its engine to process sensory effects decoupled from the end-user application. *SEM standardization support* points out if the system supports standardized SEM, whereas *SEM standardization expansion* indicates whether the system can support more than one standardized SEM or another form of SEM. *Hardware heterogeneity support* is the possibility to add new devices without interfering in the SER's structure. *Multi-communication protocols for applications support* has to do with supporting integration with end-user applications using standardized communication protocols. *SDK/API accommodation to support new integration* means that the system is able to third-party tools without interfering in the SER's structure. *Multi-operating system support* indicates if the system can run on different operating systems. Finally, *Open-source* means if the system is open-source.

2.3 Gaps and Shortcomings

The works found in the literature certainly bring contributions to the mulsemedia arena. However, most of the systems presented in this chapter were built for too specific purposes, without allowing for interoperability in software and hardware level, which may hinder the integration of other applications in home environments, museums, 4D movie theaters, VR applications enriched with multisensory effects, etc., with hardware for delivering sensory effects. Conversely, some authors present proposals of conceptual architectures and frameworks

Table 1 – Main features of the multipurpose mulsemedia systems. (1) Sensorama, (2) SEMP, (3) Sensible Media Simulator, (4) Media Processing Engine, (5) Multimedia-Multisensorial Platform, (6) PlaySEM SER 1.0.0, (7) IoT-based architecture, (8) TheaterTalk/IoTtalk Server, and (9) PlaySEM SER 2.0.0 (framework presented in this thesis).

Mulsemedia Systems Features	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Timeline-based applications functionalities	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Event-based applications functionalities	No	No	No	No	No	Yes*	Yes*	No	Yes
Reusable SER	No	No	No	No	No	Yes	Yes	Yes*	Yes
SEM standardization support	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes
SEM standardization expansion	No	No	No	No	No	No	No	No	Yes
Hardware heterogeneity support	No	No	No	No	No	No	Yes*	Yes*	Yes
Multi-communication protocols for applications support	No	No	No	No	No	No	No	No	Yes
Multi-connectivity protocols for hardware support	No	No	No	No	No	No	No	Yes*	Yes
SDK/API accommodation to support new integration	No	No	No	No	No	No	No	No	Yes
Multi-operating system support	No	No	No	No	Unk	Yes	Yes	Unk	Yes
Open-source	No	Yes	No	No	No	Yes	Yes*	No	Yes

No: the feature is absent.
Yes: the feature is fulfilled by the system.
*Yes**: the feature is partially met (provided by a third-party already integrated to the system).
Unk: there is not enough or no information available.

Source: Created by the author.

taking into account some degree of standardization, but these ideas still need to be evolved and materialized somehow to real environments. As proposed by some authors, IoT might emerge as a solution to integrate different devices in mulsemedia environments. However, if not properly assembled, the system can become cumbersome and as a result, include undesired delays. Furthermore, IoT solutions would have to adapt themselves to support interaction with heterogeneous timeline- and event-based applications and take into consideration standardization and techniques to cope with SEM. As for SDKs and APIs, they have been developed to deal with each sense separately. They usually aim at working as interfaces for specific devices. Despite this limitation, they can be combined in a framework to perform varied tasks. Haptic has appeared as the most supplied of software solutions to cope with heterogeneity, whereas olfactory and gustatory have been little explored due mainly to the shortage of off-the-shelf devices.

When it comes to QoE, identifying stringent numbers for tolerable delays in digital multi-sensory experiments is quite strenuous to do and rather dependent on the context of the interaction. Many studies have identified limits in varied setups with distinct types of sensory effects, as described in Section 2.1.2. Nonetheless, temporal aspects shall not be forgotten when employing different technologies to deliver sensory effects so as to avoid negative effects in QoS that could eventually be detrimental to QoE. By knowing timetables beforehand, one could design a system taking them into consideration to avoid unbidden delays.

Mulsemedia systems give rise to a complex scenario stemming from their unconventional needs while producing, transmitting, integrating, and presenting distinct sensory effects integrated with audiovisual content under varied constraints and conditions. Whilst the literature has made evident the increases in QoE in mulsemedia systems, as shown in Section 2.1.3.1, the mass adoption of these systems will cross the line of how the experience is delivered. With constant advances in software and hardware, approaches to dealing with variability of scenarios of usage will be of crucial to provide proper tools so that researchers and designers alike can create mulsemedia delivery systems and exploit multisensory interactions (OBRIST et al., 2017). From the perspective of related work, the following major gaps and shortcomings were identified:

- Mulsemedia renderers have been coupled in multimedia applications making harder reuse with other applications;
- When mulsemedia renderers are decoupled from multimedia applications, they lack support for varied protocols to interact with them;
- When mulsemedia solutions are able to work with timeline-based multimedia applications, they seldom give support for event-based ones;
- Whilst MPEG-V has been supported by some mulsemedia systems, they are not prepared for changes in the case that they need to support SEM in another standard;
- Delay is rarely taken into account to design mulsemedia systems—it lacks mechanisms to eventually compensate delays introduced for some hardware or connectivity and communication protocols;
- Hardware heterogeneity is mostly tackled isolated by some SDKs or APIs (in haptic) which means that mulsemedia systems hardly ever provide configurable means for device replacement;
- Mulsemedia systems lack some way to adapt themselves for different profiles of usage without changing their code;
- Noticeably, most mulsemedia systems do not consider future growth (extensibility) and rely only on existent technologies, protocols, and standards.

In the next chapter, the challenges and requirements that take totally new valences for mulsemedia systems taking into account these major gaps and shortcomings are presented. In addition, a conceptual architecture that copes with them is described.

3 Mulsemedia Systems Challenges, Requirements, and a Conceptual Architecture

This chapter describes the major challenges for delivering sensory effects from the outlook of gaps and shortcomings found in the past chapter. Through hypothetical scenarios of usage, requirements are identified to overcome these challenges and underpin the proposal of a conceptual architecture that addresses variability of scenarios, which include recommendations to cope with the variation of end-user applications and sensory effect devices through the support and reuse of communication and connectivity protocols, and sensory effects metadata standardization. The conceptual architecture's goals, guidelines, and components are then described.

3.1 Challenges and Requirements for Mulsemedia Delivery Systems

From the literature review in Chapter 2, it is not hard to realize the multitude of concerns that mulsemedia systems have to cope with. Software- and hardware-based systems are increasingly become bigger and more complex due to different scenarios, conditions, and constraints they have to operate in. Mulsemedia systems fit this context. Major obstacles and issues in deploying mulsemedia systems have been identified in (GHINEA et al., 2014; MURRAY et al., 2017b; OBRIST et al., 2017; COVACI et al., 2018). They include mulsemedia integration, synchronization, intensities, wearable, and other heterogeneous devices for delivering sensory effects, and remote delivery of mulsemedia components. Thereby, focused exclusively on the design and implementation of mulsemedia systems, it is reinforced the fundamental challenges for delivering multisensory effects to heterogeneous applications first described in (SALEME; SANTOS; GHINEA, 2019a) under the influence of Broy (2006).

Sections 3.1.1 to 3.1.3 characterize the identified challenges and establish requirements to be addressed in order to face each of them. The requirements are identified by an ID (C_iR_j, which stands for Challenge *i*, Requirement *j*) to be subsequently referred to and work as inputs for the conceptual architecture. Although the applicability of these requirements can be verified through the implementation of case studies further described in this work (Chapter 5, Section 5.1), their formal measurement is not proposed hereby. Furthermore, whilst it is acknowledged that measures can promote efficiency, this work is limited to applying them, determining whether they were met or not, and mapping them into the conceptual architecture.

3.1.1 Challenge I - Multifunctionality and Reusability

As described in Section 2.1.4 of the previous chapter, there is constant advance in mulsemedia devices, with a plethora of ways to connect them to systems. In parallel, new approaches to interacting with multimedia applications have emerged, such as multi-touch interfaces, voice processing, and brain-computer interfaces, giving rise to new kinds of complex interactive systems. Due to the heterogeneity that these devices and user interfaces have been developed, integrating them into other systems requires adaptation to different circumstances of operating systems, presentation interfaces, standards, connectivity, and communication. The scenario below helps illustrate the challenge of multifunctionality and reusability.

Scenario challenge I - Multi-interactive environments.
Prof. John is keen on carrying out studies on user QoE in games and movies. He has a powerful desktop computer (at Lab A.) on which he places users to play educational games and a laptop (at Lab B.) where he puts users to watch movies. He recently found out a compelling and engaging way that he thinks it would enrich users' experiences even more by adding extra features to stimulate other senses. In no time, he bought devices to produce light, wind, vibration, and scent effects in his research environments. He was so engrossed with the possibilities that he forgot to check whether the applications could be seamlessly integrated with his new devices. Yet, he realized that the desktop computer runs a different operating system from the laptop.
Main requirements: (a) multifunctional, (b) compatible, (c) portable, and (d) compliant/interoperable.

In cases where multimedia applications, such as video players, games, VR-360° applications, have to be integrated with new sensory effect devices taking into account different operating systems, requirements for mulsemedia systems to be multifunctional (C_1R_1), compatible (C_1R_2), portable (C_1R_3), and compliant/interoperable (C_1R_4) should be met. *Multifunctional* means that a mulsemedia system shall provide operations for multimedia applications to add multisensory effects into it, being able to work with timeline- and event-based multimedia applications. It has to be *compatible* with its counterpart, i.e. it should provide reusable implementations of services to communicate with the multimedia applications. Moreover, it should be able to run in the expected environments that the user needs (*portable*). It leads to an explicit requirement for compliance, that is, mulsemedia systems should meet rules or standards to allow interoperability in several levels, such as communication with applications and connectivity with devices.

Architectural patterns should be applied to mulsemedia systems for allowing reuse. By decoupling layers, multimedia applications could use common services from an interoperable mulsemedia system that decouples presentation (end-user application) from mulsemedia processing and rendering. This separation could keep apart concerns such as adaptive multimedia streaming, buffering, video encoding and decoding, among others not directly related to mulsemedia processing and rendering.

3.1.2 Challenge II - Reactivity and Timeliness

Event-based multimedia applications (Chapter 2, Section 2.1.1) are ones for which every second is precious. Mulsemedia systems, in this case, shall work in real-time reacting as quickly as possible to events generated by event-based multimedia applications controlling swiftly all the sensory effect devices required to perform an action. Not meeting real-time deadlines would be detrimental to user QoE as depicted in Chapter 2 (Section 2.1.2). The following scenario exemplifies this issue.

Scenario challenge II - Responsiveness in games.
Prof. John was finally able to integrate his devices into his systems. However, he realized that the devices were not responding properly when the users were playing a game, that is, after an explosion scene, the wind and vibration devices started much later than the scene on the computer screen. Sometimes, the sensory effects were not even delivered to the users.
Main requirements: (a) reliable, and (b) responsive.

These two challenges demand mulsemedia systems be reliable (C_2R_1), and responsive (C_2R_2). Responsiveness in real-time event-based mulsemedia applications is crucial. An input will entail one or more reactions in mulsemedia systems instantaneously. It shall be *reliable* in order to perform actions consistently, producing correct outputs under a time constraint. To be *responsive* signifies that short response time is required for some situations such as in the scenario portrayed. Thus, delay is an issue that needs to be taken into account in mulsemedia systems.

3.1.3 Challenge III - Manageability and Configurability

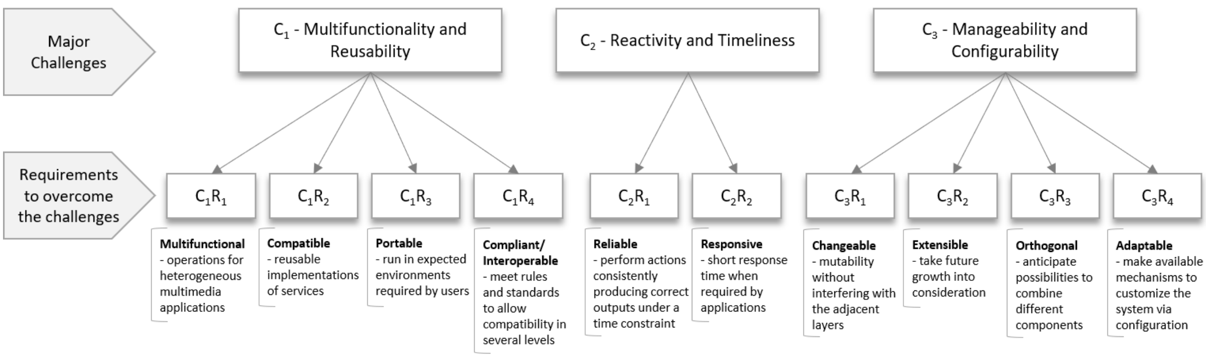
Complex architectures composed of heterogeneous devices present a significant challenge for mulsemedia systems. A device can be connected to a mulsemedia system through many connectivity protocols such as Wi-fi, Bluetooth, Zigbee, and so on, allowing wireless communication. In addition to that, a multimedia application could impose different communication protocols on the mulsemedia system depending on the programming language it was developed, operation system it was designed for, and temporal restriction it has to meet. The next scenario depicts story whereby Prof. John wants to replace his scent device seeking new features and applications, but the connectivity protocol is not the same as the one his previous device used. Moreover, he also faces issues related to the code, which has low cohesion and is highly coupled.

> Scenario challenge III - Replacing devices.
> *Lately, Prof. John decided to replace his scent device because the prior had a limitation to emit just a few sets of scents, and it was restricted to one scent at a time. He found a wearable emitter that met his needs. On top of that, he wanted to use it in a scented VR application so that he could investigate user QoE under the perspective of content and smell hedonic valence. In contrast to his old device, the latter is connected through Bluetooth. Even worse, he noticed that his system was hard-coded to work with the former USB/Serial device.*
> Main requirements: (a) changeable, (b) extensible, (c) orthogonal, and (d) adaptable.

Therefore, mulsemedia systems should be able to work with heterogeneous scenarios. This is the case in which architectural and design patterns boost mulsemedia systems to be built so that they are *changeable* (C_3R_1), *extensible* (C_3R_2), *orthogonal* (C_3R_3), and *adaptable* (C_3R_4). The ability to be *changeable* features mutability in its structure without interfering with the adjacent layers of the mulsemedia system. Extensibility plays an important role in changeability in the sense that the design of the system takes future growth into consideration. Orthogonality in architectural level will also aid the designer to anticipate possibilities to combine different components. Adaptability makes available mechanisms to customize the system to their needs without needing to change the system, that is, mulsemedia systems should be designed so that users can configure and extend the application according to their specific needs.

Figure 4 portrays a summary of the major challenges and requirements for mulsemedia systems delivery described in this section.

Figure 4 – Summary of the major challenges and requirements for mulsemedia systems delivery.



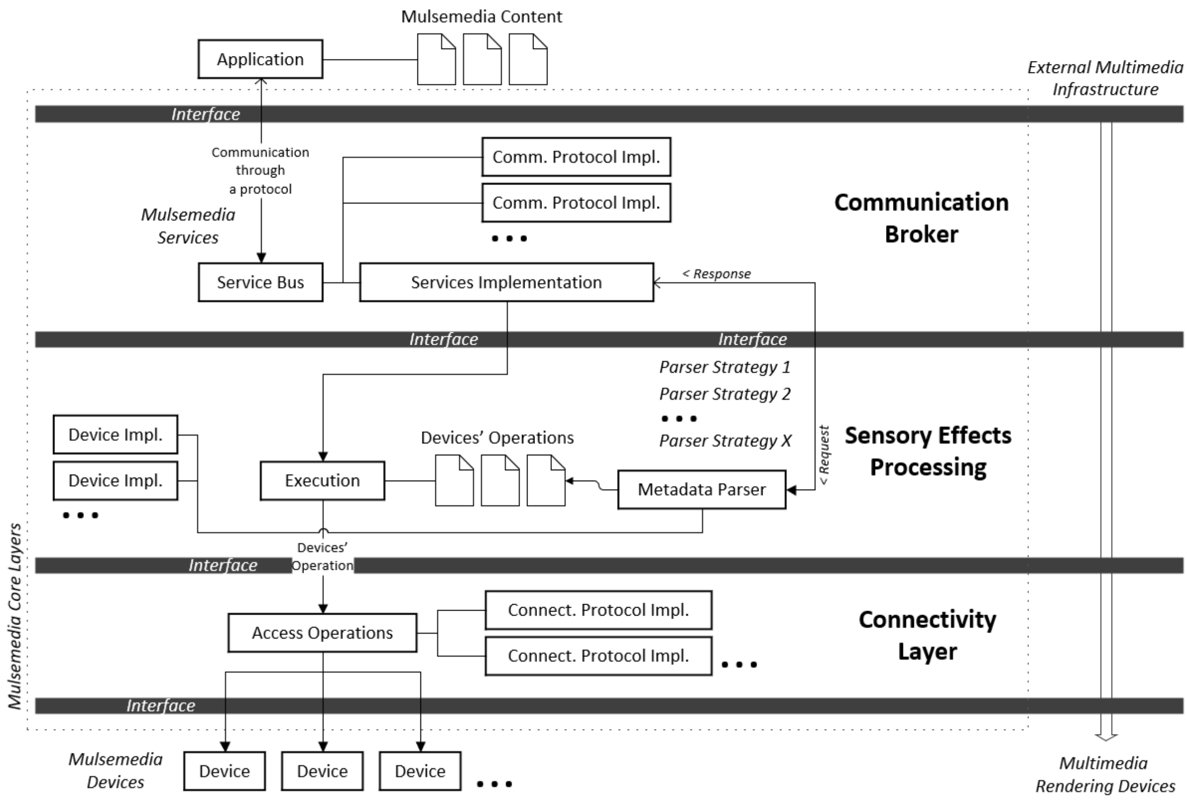
Source: (SALEME; SANTOS; GHINEA, 2019b).

3.2 Conceptual Architecture

What distinguishes this conceptual architecture for mulsemedia delivery system from others is its emphasis on separated layers that promote reuse of communication protocols for applications, SEM standards, specific device implementations, and connectivity protocols to reach sensory effect devices. This conceptual architecture does not intend to provide technical details, but rather it focuses on key constructs and abstractions to help the design of reusable mulsemedia systems taking into account the requirements discussed in the last section. On the other hand, this conceptual architecture is a starting point to elaborate lower level architectures, which is done in Chapter 4. Its main idea is to show the decomposition of the system, which includes its layers and goals, guidelines for its design, and main components, including definition (purpose), responsibilities (duties), constraints (restrictions on the degree of freedom), interactions (collaborations with other components—uses and used by), resources (items managed, affected, or needed by it), processing (activities to perform its responsibilities), and interfaces (services provided by it).

The conceptual architecture (Figure 5) is divided into several layers: (i) *Mulsemedia Applications*, (ii) *Communication Broker*, (iii) *Sensory Effects Processing*, (iv) *Connectivity Layer*, and (v) *Sensory Effects Rendering*, which are subsequently described. Two real instantiations of this architecture from the work of Covaci et al. (2019b) (360° Mulsemedia System) and Jalal et al. (2018b) (TV Broadcast Mulsemedia System) are presented in APPENDIX A.

Figure 5 – Conceptual architecture for Mulsemedia Delivery Systems.



Source: Created by the author.

3.2.1 Mulsemedia Applications

3.2.1.1 Goals and Guidelines

This layer represents mulsemedia applications (timeline- or event-based), its associated content (such as metadata files), and external multimedia infrastructure, such as AR/RV devices, content, sensors, etc. Whilst it is recognized that these latter components are part of mulsemedia delivery systems, this infrastructure is isolated from the main application to separate concerns and focus on mulsemedia issues. Thereby, other multimedia concerns such as adaptive streaming, media buffering, video encoding and decoding are kept apart and not dealt with in this conceptual architecture.

Mulsemedia Applications should be able to interface with mulsemedia services provided by SERs from this point. These services, provided by the layer *Communication Broker* would include operations to convey and process SEM, start, pause, position, and stop sensory effects rendering. In timeline-based applications, these operations should be synchronized with the user interface whilst in event-based ones, points of triggering events shall be coded to invoke sensory effects rendering.

3.2.1.2 Components

3.2.1.2.1 Application

This component constitutes of end-user applications where mulsemedia is consumed.

Responsibilities: It conveys multisensory information from the virtual world to end-users or provides mechanisms to interact with computers.
Constraints: It shall synchronize user interface operations with mulsemedia services when needed.
Interactions: *Mulsemedia Content* and *Service Bus* (*Communication Broker*).
Resources: *Mulsemedia Content*.
Processing: It invokes operations from the *Service Bus*.
Interfaces: Output -> *Communication Broker*.

3.2.1.2.2 Mulsemedia Content

This component represents one or more sensory effects metadata files.

Responsibilities: It provides times, intensity, and other related attributes to sensory effects to start and stop for the interactions provided by the applications.
Constraints: Mulsemedia content shall be provided in some standardized fashion such as MPEG-V but not restricted to it.
Interactions: *Application*.
Resources: N/A.
Processing: N/A.
Interfaces: N/A.

3.2.2 Communication Broker

3.2.2.1 Goals and Guidelines

The communication broker has the purpose of making easier the reuse of services provided by SERs in a uniformed fashion via a single point of access for mulsemedia applications (timeline- or event-based) under the operation of required protocols. Not only does this layer abstract the complexity of sensory effects processing and hardware interfacing, but it also allows the inclusion of communication protocols without interfering in them. The latter feature benefits legacy or applications restrained by their technical environment, in which a reduced set of choices of communication protocols to integrate them with SERs can be made.

Operations requested by end-user applications to the communication broker shall be forwarded to the *Sensory Effects Processing* layer. A mechanism to create a facade of services to receive requests of operations (such as to convey and to process SEM, to start, to pause, to position, and to stop sensory effects rendering) from mulsemedia applications must be available through different communication protocols. The services do not need to be rewritten for each protocol—implementation of these common operations has to be considered so that they can be inherited.

The requirements that the communication broker should meet are: *multifunctionality* (C_1R_1), to provide services for different types of applications; *compatibility* (C_1R_2), to provide reusable implementations of services to communicate with the multimedia applications; *extensibility* (C_3R_2), to take future growth into consideration by allowing the addition of new communication protocols; *compliance/interoperability* (C_1R_4), the communication protocols that this layer uses shall be standardized; *orthogonality* (C_3R_3), to ensure that designers can anticipate possibilities to combine different protocols with the services offered by this layer; *adaptability* (C_3R_4), to configure and extend the system according to specific needs; *changeability* (C_3R_1), to allow changes without meddling in concerns not related to communication with mulsemedia applications; *portability* (C_1R_3), to run in the expected environments; *reliability* (C_2R_1), to guarantee that messages are correctly delivered from mulsemedia applications to this layer; and *responsiveness* (C_2R_2), to meet strict requirements imposed by event-based applications.

3.2.2.2 Components

3.2.2.2.1 Service Bus

This is a communication mechanism where end-user applications reach mulsemedia services through communication protocols.

Responsibilities: It provides a facade of services to receive requests of operations (such as to convey and to process SEM, to start, to pause, to position, and to stop sensory effects rendering) from mulsemedia applications.
Constraints: It shall consider performance requirements (for event-based applications) and support standardized protocols for communication with mulsemedia applications.
Interactions: *Application* (*Mulsemedia Applications*), *Communication Protocol Implementations*, and *Services Implementation*.
Resources: N/A.
Processing: It transfers requests for operations from mulsemedia applications to the *Services Implementation* and provides responses to them.
Interfaces: Input -> *Mulsemedia Applications*.

3.2.2.2.2 Communication Protocol Implementations

This component supports the *Service Bus* with specific implementations for communication protocols.

Responsibilities: It provides specific implementations for communication protocols.
Constraints: N/A.
Interactions: *Service Bus*.
Resources: N/A.
Processing: It receives requests and provides responses in a standardized network communication protocol.
Interfaces: N/A.

3.2.2.2.3 Services Implementation

This is a component to provide the implementation generic of operations for the *Service Bus* independently of communication protocols.

Responsibilities: It has to feed the *Service Bus* with the implementation of common operations for mulsemedia applications such as to convey and to process SEM, to start, to pause, to position, and to stop sensory effects rendering.
Constraints: N/A.
Interactions: *Service Bus*, and *Execution* and *Metadata Parser* (*Sensory Effects Processing*).
Resources: N/A.
Processing: It performs operations for the services exposed by the *Service Bus*.
Interfaces: Output -> *Sensory Effects Processing*.

3.2.3 Sensory Effects Processing

3.2.3.1 Goals and Guidelines

The sensory effects processing layer deals with metadata processing converting them into specific commands for controlling the sensory effect devices. Moreover, it transfers requests of operations such as to convey and to process SEM, to start, to pause, to position, and to stop sensory effects rendering from the *Communication Broker* to the *Connectivity Layer* to control the devices.

This layer shall allow the inclusion and configuration of different strategies to convert sensory effect metadata into specific commands for different devices. Doing this process without affecting adjacent layers is key. Towards this end, these parser strategies should not generate commands for devices directly. Instead, they shall employ a generic message structure when coded, and then, when the system runs, this structure has to be converted into specific commands to control the sensory effect devices with the help of classes that implement specific commands for the devices. At this stage, these device implementations shall not know the connectivity protocol to reach the device. This abstraction should be created to allow changes of connectivity protocols (dealt by *Connectivity Layer*), for instance when the same device has to operate with wireless or wired protocols depending on the need of the developers.

The requirements that this layer should take into account are: *compliance/interoperability* (C_1R_4), sensory effect metadata standards should be implemented towards this need; *extensibility* (C_3R_2), to allow further expansion to support different sensory effect metadata standards and sensory effect devices; *orthogonality* (C_3R_3), to make sure designers can envisage the combination of other layers with different sensory effect metadata standards and devices; *changeability* (C_3R_1), to ensure modification without interfering in other layers; and *adaptability* (C_3R_4), to set up and expand the system according to specific needs.

3.2.3.2 Components

3.2.3.2.1 Metadata Parser

This component converts SEM into commands to control sensory effect devices.

Responsibilities: It transforms SEM conveyed by the *Services Implementation* (*Communication Broker*) into specific operations for the sensory effect devices to be used by the *Execution* component.
Constraints: Parser strategy(ies) for SEM standard(s) shall be implemented.
Interactions: *Services Implementation* (*Communication Broker*), *Devices' Operations*, and *Device Implementations*.
Resources: *Devices' Operations*
Processing: It processes SEM under requests of *Services Implementation* (*Communication Broker*), stores the specific operations for devices in a structure to be subsequently retrieved, and provides a response as soon as it finishes.
Interfaces: Input -> *Communication Broker*.

3.2.3.2.2 Devices' Operations

This is a structure to store temporarily one or more specific operations for the sensory effect devices.

Responsibilities: It provides times, intensity, and other related attributes to sensory effects to start and stop specific sensory effect devices.
Constraints: N/A.
Interactions: *Metadata Parser* and *Execution*.
Resources: N/A.
Processing: N/A.
Interfaces: N/A.

3.2.3.2.3 Execution

This component executes operations to control sensory effect devices.

Responsibilities: It is responsible for executing operations demanded by the *Services Implementation* (*Communication Broker*) interfacing with *Connectivity Layer* to communicate with sensory effect devices.
Constraints: Its activities have to operate in real-time.
Interactions: *Devices' Operations*, *Services Implementation* (*Communication Broker*), and *Access Operations* (*Connectivity Layer*).
Resources: *Devices' Operations*.
Processing: It reads specific operations from the *Devices' Operations* and accesses sensory effect devices through the *Access Operations* (*Connectivity Layer*).
Interfaces: Input -> *Communication Broker*. Output -> *Connectivity Layer*.

3.2.3.2.4 Device Implementations

This is a component that provides implementations for specific hardware.

Responsibilities: To provide specific commands to control each hardware.
Constraints: N/A.
Interactions: *Metadata Parser*.
Resources: N/A.
Processing: It provides specific commands for the *Metadata Parser* to build a timetable with the *Devices' Operations* when processing SEM.
Interfaces: N/A.

3.2.4 Connectivity Layer

3.2.4.1 Goals and Guidelines

This layer makes available a set of connectivity protocols so that the sensory effect devices can be connected through different protocols. In environments where wires are bulky, a designer might want to replace wired connectivity by a wireless one using the same device. If the sensory effect device supports more than one way of integration, this should be done without changing the devices' implementation. There are other cases where different devices use the same connectivity, and the implementation of this connectivity can be leveraged by them. Thus, this layer is kept apart from the *Sensory Effects Processing* layer so that connectivity protocols can be reused.

Once the command for the sensory effect devices has already been processed in another layer, the connectivity layer shall provide to the *Sensory Effects Processing* layer only operations to access the devices, such as opening, closing, and sending messages through different protocols that should be implemented. Although communication protocols (like those described in the communication broker) can be used over connectivity protocols, this conceptual architecture does require it because most sensory effect devices do not implement standardized communication mechanism apart from connectivity.

The requirements that the connectivity layer should consider are: *portability* (C_1R_3), to run in the expected environments; *compliance/interoperability* (C_1R_4), the connectivity protocols that this layer shall be standardized; *reliability* (C_2R_1), to make sure that messages are conveyed from the *Sensory Effects Processing* reach the devices; *responsiveness* (C_2R_2), to provide connectivity protocols that meet strict requirements imposed by event-based applications; *changeability* (C_3R_1), to allow alterations without interfering in concerns not related to connectivity with sensory effect devices; *extensibility* (C_3R_2), to provide flexibility to add new connectivity protocols; *orthogonality* (C_3R_3), to allow designers to anticipate possibilities to combine different protocols for the devices; *adaptability* (C_3R_4), to configure and extend the system according to specific needs.

3.2.4.2 Components

3.2.4.2.1 Access Operations

This component provides common operations to access sensory effect devices, such as open/close connections, and send messages.

Responsibilities: To provide connectivity operations to each sensory effect device.
Constraints: It has to offer the choice to keep the connection with devices open, if supported by them, to optimize performance.
Interactions: *Execution* (*Sensory Effects Processing*), and *Device* and *Connectivity Protocol Implementations*.
Resources: N/A.
Processing: It receives commands from the *Execution* (*Sensory Effects Processing*) component and transfers them to sensory effect devices with the assistance of the *Connectivity Protocol Implementations*.
Interfaces: Input -> *Sensory Effects Processing*. Output -> *Sensory Effects Rendering Devices*.

3.2.4.2.2 Connectivity Protocol Implementations

This component supports the *Access Operations* with specific implementations for connectivity protocols.

Responsibilities: It provides specific implementations for connectivity protocols.
Constraints: N/A.
Interactions: *Access Operations*.
Resources: N/A.
Processing: It receives requests and provides responses in a standardized network connectivity protocol.
Interfaces: N/A.

3.2.5 Sensory Effects Rendering Devices

3.2.5.1 Goals and Guidelines

This layer represents devices that deliver sensory effects to end-users included in the mulsemedia system, which should be reached only by the *Connectivity Layer* to avoid violating this conceptual architecture.

3.2.5.2 Components

3.2.5.2.1 Device

This component constitutes a set of hardware to deliver sensory effects to end-users.

Responsibilities: It renders sensory effects such as wind, vibration, smell, etc.
Constraints: N/A.
Interactions: *Access Operations* (*Connectivity Layer*).
Resources: N/A.
Processing: It performs electro-chemo-mechanical operations to deliver sensory effects.
Interfaces: Input -> *Connectivity Layer*.

3.3 Concluding Summary

This chapter started by describing the challenges of mulsemedia delivery systems to deal with variability of scenarios of usage. Representative requirements of desired characteristics to overcome those challenges were discussed through hypothetical scenarios. They underpin the proposal of a conceptual architecture that deals with the variation of end-user applications and sensory effect devices through the support and reuse of communication and connectivity protocols, and SEM standards. This conceptual architecture is grounded on the separation of concerns by dividing a mulsemedia system into five main layers: (i) *Mulsemedia Applications*, which isolates multimedia issues; (ii) *Communication Broker*, which provides the reuse of mulsemedia services provided by SERs through different communication protocols; (iii) *Sensory Effects Processing*, which copes with metadata processing regardless of the SEM standard; (iv) *Connectivity Layer*, which allows the reuse of different connectivity protocols for sensory effect devices independently of the implementation of commands for devices; and (v) *Sensory Effects Rendering*, to represent devices that deliver sensory effects to end-users. The layers' components are described, and their interfaces are mapped to support the design of lower-level architectures.

As stated in this chapter, this conceptual architecture is restricted to mulsemedia components. Furthermore, it does not intend to provide technical details nor naming conventions. Nevertheless, it provides the basis to design mulsemedia delivery systems that could be built once and extended continually to support different scenarios of usage in a tailored fashion taking into consideration its flexibility to expand and reuse its components. Although this conceptual architecture is empirically validated through case studies where it has been applied (further described), this work does not perform formal validations. Whilst this validation process would increase the architecture's trust level, the trade-off between formality and architecture's assimilation could be compromised by adding too many particulars to non-critical systems.

Lowering the architecture level is not straightforward. Indeed, the odds are high that more technical challenges will emerge when interweaving the system's components. Software design issues such as creating an object by specifying a class explicitly, dependence on hardware and software platform, dependence on object representations or implementations, tight coupling, and extending functionality by subclassing are examples of them. Thus, a lower level of abstraction of this conceptual architecture is precisely what the next chapter portrays whilst materializing it into a mulsemedia framework.

4 Framework for Delivering Sensory Effects to Heterogeneous Systems

This chapter presents the materialization of the conceptual architecture into a framework, named PlaySEM SER 2[1]. It strictly follows the principles established in the previous chapter. PlaySEM SER 2's description is based on the work of Saleme, Santos, and Ghinea (2019b). Its evolution and design patterns applied are introduced followed by a depiction of the framework's architectural design, which involves the implementation of the communication broker, the sensory effects processing, and the connectivity with devices. Furthermore, details on the interaction between timeline- and event-based applications with PlaySEM SER 2 are provided. Finally, configurations strategies that encompass initial setup, communication and connectivity protocols, temporal aspects, and debug and simulation are portrayed. The chapter ends with a concluding summary highlighting the framework's strengths and acknowledging current technical limitations.

4.1 PlaySEM SER's Evolution

This framework unfolds from the PlaySEM platform described in Section 2.2.3. PlaySEM SER has evolved both vertically and horizontally to meet the aforementioned requirements, to accommodate changes, and thus, to become a framework. Vertically leads to the separation of levels to cope with abstraction, whereas horizontally means to have logical components to deal separately with different concerns. Back then, in (SALEME; SANTOS, 2015), PlaySEM SER did not have an adaptable and scalable architecture whereby new and forthcoming protocols and technologies could be included without changing its components despite being decoupled from its multimedia counterpart. A device replacement meant it should be compatible with its same microcontroller and the code of the latter updated. Moreover, it was inextricably linked to MPEG-V, which hindered the use of PlaySEM SER with either other standards or someone's own metadata. It did not provide support for smell and dealt inadequately with event-based applications. Third-party components such as libraries that implement communication and connectivity protocols were not managed, running the risk of being out-of-date. A change meant an alteration in the code, not only in a configuration file. As reuse is a pivotal aspect of PlaySEM SER 2, Java *Generic Types* with *Reflection* were combined to support a set of architectural and design patterns that allow it to be applied to different contexts. Other reasons for the choice of Java stem chiefly from the following facts: (i) Java is rather popular—it is the second most tagged programming language on GitHub (a code hosting platform with more than 2 million

[1] PlaySEM SER 2.0.0 available at <https://github.com/estevaosaleme/PlaySEM_SERenderer/releases/tag/2.0.0>

active repositories) up to the date of this thesis (GITHUT, 2019), just behind JavaScript; (ii) Java is platform-independent, which means that software developed in this language is able to run on different types of machines. This would be especially useful, for instance, to port PlaySEM SER to trendy wearable mulsemedia systems; (iii) it is object-oriented (OO), which means that a plethora of OO widespread design patterns can be applied to PlaySEM SER; (iv) it is less complex than C++; and (v) there are several third-party resources that are developed as libraries that can easily be integrated into PlaySEM SER to support new protocols for either connectivity or communication, for the sake of argument. Although the framework is written in Java, the conceptual architecture described in Chapter 3 can be implemented in other programming languages.

4.2 Design Patterns Applied

A layered architecture subdivides solutions so that each tier can evolve separately. Criteria for separating concerns may vary from system to system and include abstraction, granularity, hardware distance, and rate of change (BUSCHMANN; HENNEY; SCHMIDT, 2007). Concurrently, design patterns aid developers in making design choices so that they do not compromise reusability (GAMMA et al., 1995). Patterns allow reusing the experience of others who experienced similar problems by describing a recurrent problem in an environment and a core solution to them without ever doing it the same way twice.

PlaySEM SER 2 has applied the patterns `abstract factory`, `singleton`, `facade`, `observer`, `strategy`, and `template method` to make the solution more flexible and ultimately reusable. These patterns help to avoid many problems such as creating an object by specifying a class explicitly, dependence on hardware and software platform, dependence on object representations or implementations, tight coupling, and extending functionality by subclassing.

The pattern `abstract factory` was used to provide interfaces for devices, services, connectivity, and messages without specifying their concrete classes. An instance of a concrete class, which contains specific implementations, is created at run-time in this case. The main idea is that the system becomes independent of how its products and services are created, composed, and represented, therefore, promoting consistency among the classes and isolating specific solutions.

An application needs only one instance of a factory per group of matter, which leads to the use of the pattern `singleton`. It provides a central point of access for services, devices, and connectivities within PlaySEM SER 2. Moreover, the system relies on this pattern for accessing a single instance of a timeline per event so that the sensory effects are arranged over it.

`Abstract factory` classes are often implemented with `factory methods`, which in turn, let subclasses redefine certain steps of an algorithm without changing the algorithm's structure. It is useful to implement the invariant parts of an algorithm leaving it up to subclasses to implement the specific behavior. In the following sections of this chapter, abstract services, devices, and connectivities classes implementing the invariant steps of the algorithm will be presented. As distinct to `factory methods`, the pattern `strategy` was applied to vary entire algorithms eliminating conditional statements. It is a choice of implementation. For instance, this is the case where a new device for a specific kind of effect should be selected instead of another, or a new SEM standard emerges. It is used mainly for the communication broker, metadata parser, devices, and connectivity interfaces within PlaySEM SER 2.

To provide a high-level of abstraction for PlaySEM SER 2 and expose elementary mulsemedia services to multimedia applications, the `facade` pattern has been applied. It helps to layer the system, i.e. the communication broker is a `facade` for multimedia applications, whereas connectivity interface is a `facade` for the drivers of the devices providing operations to open and close connections, besides sending a command to a device. Therefore, the communication between the layers is simplified through `facades`. It also promotes weak coupling between systems. Generally, `facades` require only one instance of an object having a straight relationship with `singletons`. In addition, the `facade` pattern is combined with `abstract factories` within PlaySEM SER 2 so that the behavior of a service obeys what is defined through a selected strategy.

Last but not least, the pattern `observer` (also known as publish-subscribe) was applied to PlaySEM SER 2. After processing the SEM, the system should notify the integrated multimedia application (subscriber) the end of this process so that other operations to control the devices can be started. In the case of the classes of services provided by the communication broker, the abstract service class fires a property change so that the concrete classes reach the multimedia application via their own standards of communication.

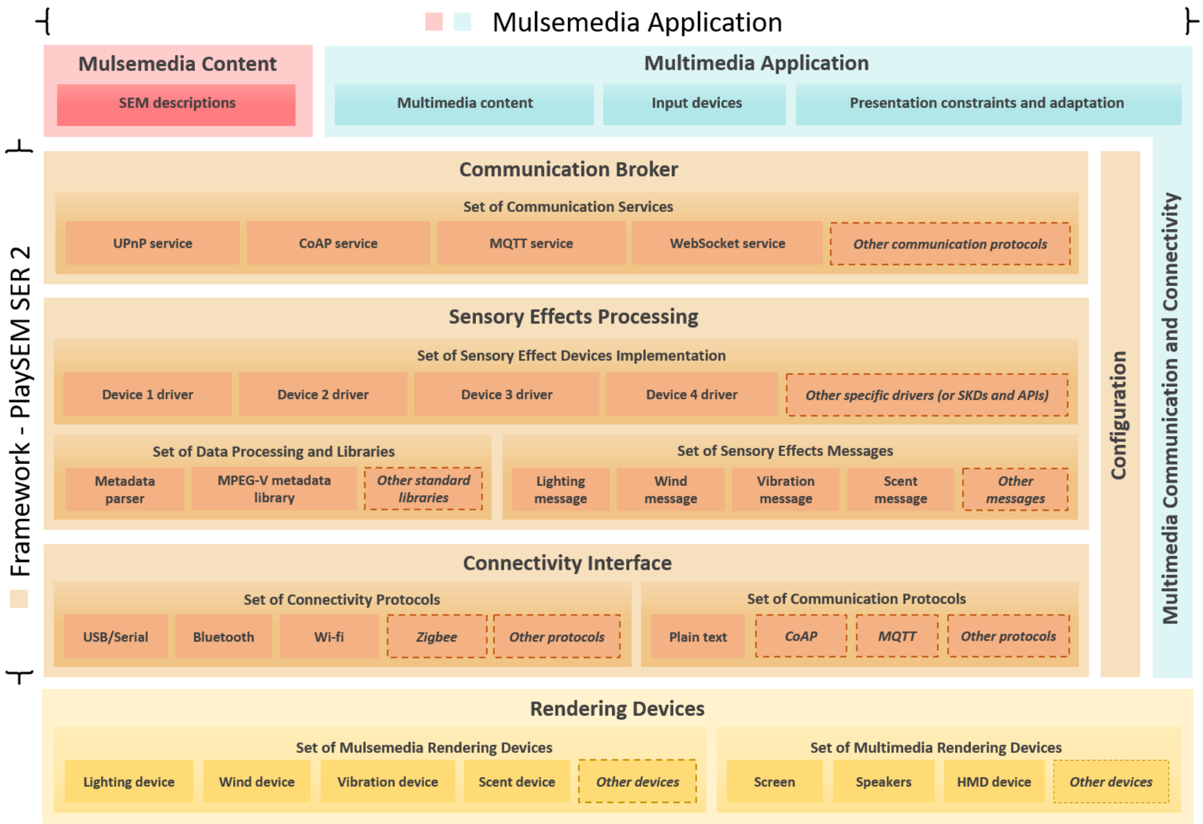
4.3 PlaySEM SER 2's Architectural Design

The architectural design of a system provides agility to disseminate knowledge about its structure (SOMMERVILLE, 2015). In this fashion, Figure 6 depicts the elements of a mulsemedia delivery system with a focus on the framework. `Mulsemedia Application` is the point of departure, followed by blocks corresponding to PlaySEM SER 2 (`Communication Broker`, `Sensory Effects Processing`, and `Connectivity Interface`), `Multimedia Communication and Connectivity` and `Rendering Devices`, which are presented next.

4.3.1 Mulsemedia Application

The `Mulsemedia Application` block encompasses several components related to applications whereby the user interacts actively. `Multimedia Application` can be either

Figure 6 – Architectural design of PlaySEM SER 2 and the relationship between mulsemedia system elements.



Source: Adapted by the author from (SALEME; SANTOS; GHINEA, 2019b).

a timeline-based application or an event-based one. Both subtypes of multimedia application can make use of `Multimedia Content` (video, audio, etc.), `Input Devices` (RGB-D camera, audio input, joystick, etc.), `Presentation Constraints and Adaptation` (spatial and temporal dimensions, transformation rules, content adaptation, etc.), and `Multimedia Communication and Connectivity` (drivers, protocols, and other schemes provided by the operating system in a layer parallelized to PlaySEM SER 2). Finally, `Mulsemedia Content` represents SEM descriptions associated with multimedia content or actions triggered by event-based applications.

4.3.2 PlaySEM SER 2 Core Layers

`Communication Broker`, `Sensory Effects Processing`, and `Connectivity Interface` form PlaySEM SER 2's core. Next, their details are provided.

4.3.2.1 Communication Broker and its Implementation

The `Communication Broker` works as a facade for a `Mulsemedia Application` to convey the SEM descriptions to PlaySEM SER 2 and to provide reusable abstract services for starting, pausing, stopping, and synchronizing multimedia content/interaction with sensory effects devices. This facade provides different ways to interact with mulsemedia applications—it works according to the application's category. The communication between a `Mulsemedia Application` and PlaySEM SER 2 `Communication Broker` can be established through many mulsemedia `services` offered by different protocols such as `UPnP`, `CoAP`, `MQTT`, and `WebSocket`. `Other communication protocols` can be easily provided in this layer. The protocol to be used will depend on the application's requirement.

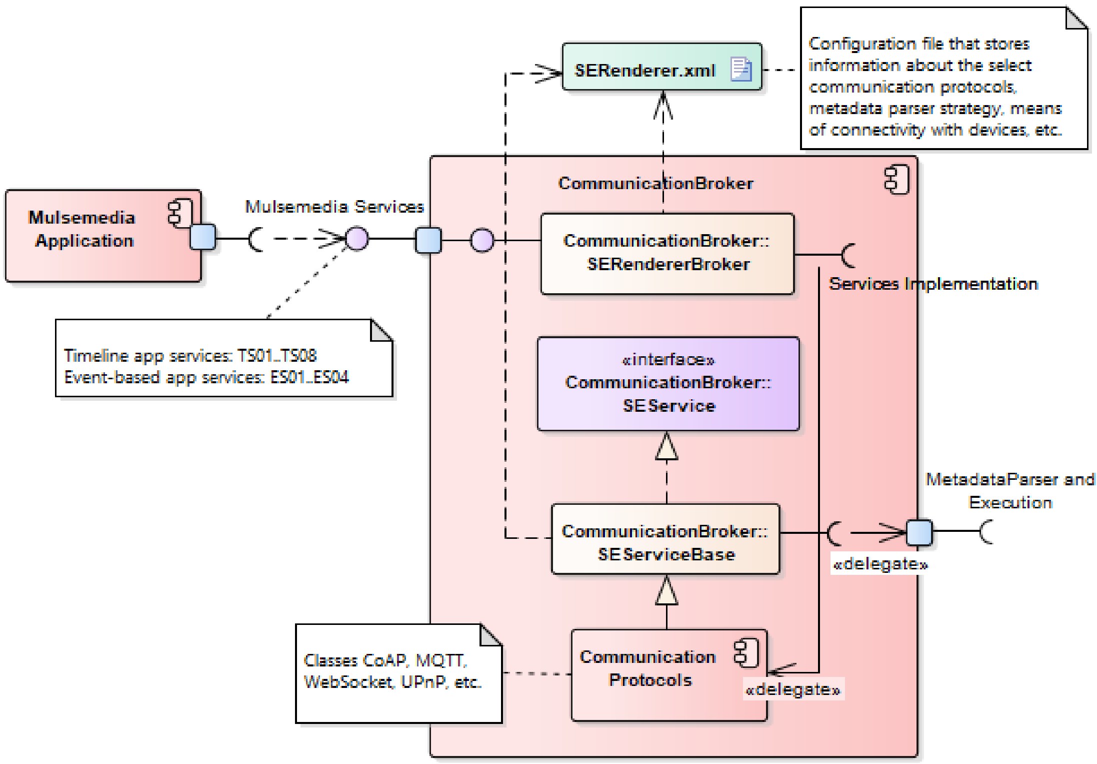
Figure 7 depicts a component diagram that shows how the components of a mulsemedia application and the communication broker are wired as well as their interfaces. `Mulsemedia Application` requires an interface to access mulsemedia services, which is provided by the class `SERendererBroker` from the `Communication Broker`. `SERendererBroker` is dependent on a configuration file `SERenderer.xml` that stores information about the selected protocols for a certain instance. As soon as it receives requests from its interface, it delegates them to a protocol that inherits the implementation of services from the class `SEServiceBase`, which implements an interface `SEService`. `SEServiceBase`, in turn, delegates the request to the sensory effects processing to parse metadata and control sensory effect devices. Details on how to interact with the services provided by the `Communication Broker` is provided in Section 4.4.

Currently, PlaySEM SER 2's communication broker supports the UPnP, CoAP, MQTT, and WebSocket protocols, but are not restricted to them. UPnP defines an architecture for communication in pervasive networks (UPnP Forum, 2008). The proposal is that devices and systems from different brands communicate with each other based on the HTTP protocol and XML language, offering flexibility and ease of use. It provides automatic discovery, description, control, presentation and sending of events remotely.

CoAP is a transfer protocol designed for use with constrained environments, that is, systems with restricted conditions (SHELBY; HARTKE; BORMANN, 2014). CoAP is based upon Representational State Transfer (REST) architecture providing a request/response and publish/subscribe interaction with very low overhead. It runs on UDP, however, it has its own reliability mechanism when a message confirmation is needed. According to Shelby, Hartke, and Bormann (2014), it could also be used over other transport protocols such as SMS, TCP, or SCTP. Kovatsch, Lanter, and Shelby (2014) reported that their framework, entitled as Californium CoAP, showed 33 to 64 times higher throughput than high-performance HTTP Web servers. Thus, CoAP has a promising performance in distributed systems.

Just like CoAP, MQTT is a lightweight protocol thought to work with resource-constrained devices (BANKS; GUPTA, 2014). However, it is limited to a publish/subscribe interaction based on topics and runs on top of TCP/IP. The protocol itself is designed to be simple. A client can publish messages to the broker and subscribe to topics for receiving events. The protocol has three qualities of service for message delivery. These are: *At most once*, with a low guarantee,

Figure 7 – Component diagram to represent the interconnection between a mulsemedia application and the communication broker provided by the framework.



Source: Created by the author.

in which message loss can occur; *At least once*, with guaranteed delivery, but duplications can happen; and *Exactly once*, which has the assurance that a message will arrive just once. The intrinsic characteristic of MQTT makes it a minimized protocol to reduce network overhead.

Unlike CoAP and MQTT, the WebSocket protocol was not designed originally for constrained devices. It aims to provide a mechanism for browser-based applications, such as games, simultaneous editing tools, user interfaces exposing server-side services in real-time, and so on, that need two-way communication with servers that do not rely on opening multiple HTTP connections (FETTE; MELNIKOV, 2011). In contrast to polling, it does not have to repeat HTTP headers in each request, attenuating communication overhead. It helps developers to build scalable real-time web applications through a single socket over the internet via a browser (PIMENTEL; NICKERSON, 2012). The fact that many applications nowadays are implemented in browsers, and most of them provide support for WebSockets, makes it a requirement for PlaySEM SER 2 to incorporate.

The services provided by PlaySEM SER 2 provide multifunctionality for different multimedia applications. Table 2 presents a description of these services and their prerequisites. An identifier containing the prefixes *E* and *T* is given for representing operations related to Event-based and Timeline multimedia applications.

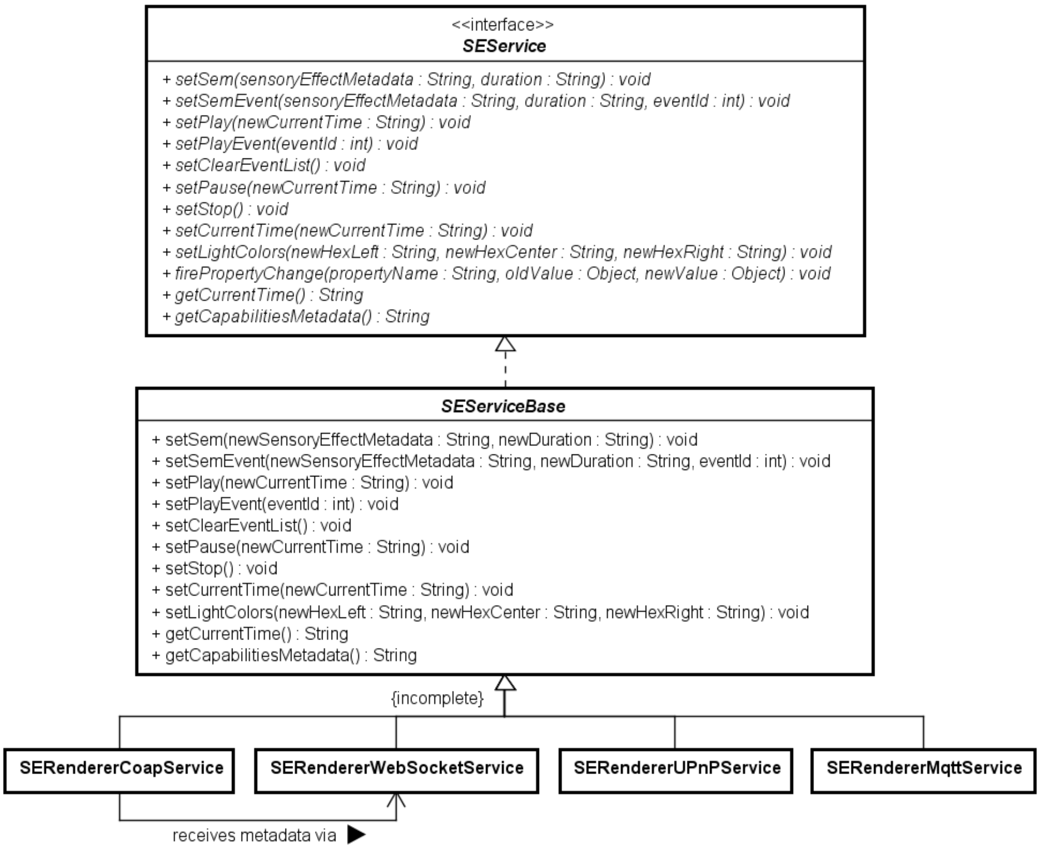
Table 2 – Services provided by PlaySEM SER 2 to promote multifunctionality for heterogeneous multimedia applications.

Id	Service	Description	Prerequisite
ES01	SetSemEvent (metadata, duration, eventId)	It processes sensory effects metadata, converting it into specific commands for handling the devices for a specific duration and a single event.	-
ES02	SetPlayEvent (eventId)	It starts rendering sensory effects for a specific preprocessed event.	ES01
ES03	SetClearEventList ()	It empties a list of previously preprocessed events.	ES01
TS01	SetSem (metadata, duration)	It processes sensory effects metadata, converting it into specific commands for handling the devices for a specific duration.	-
TS02	SetPlay (time)	It starts the rendering of sensory effects at a specific time.	TS01
TS03	SetPause (time)	It freezes the rendering of sensory effects at a specific time.	TS01, TS02
TS04	SetStop ()	It stops the rendering of sensory effects.	TS01, TS02
TS05	SetCurrentTime (time)	It places the cursor of execution at a specific time.	TS01
TS06	SetLightColors (leftColor, centerColor, rightColor)	It adjusts the colors of the lighting device in real-time. This service is used for a specific purpose of transmitting the colors extracted from a frame of a video to the device. It is separated into 3 parts which are left, center, and right of the device.	-
TS07	GetCurrentTime ()	It returns the current time on the SER. It is useful for synchronization purposes.	TS01, TS02
TS08/ ES04	GetCapabilitiesMetadata ()	It returns the capabilities of the devices in a standard.	-

Source: (SALEME; SANTOS; GHINEA, 2019b).

Additional services and other protocols can be added at this layer. Figure 8 shows an excerpt of PlaySEM SER 2's class diagram for services. There is an interface `SEService` that determines which services will be implemented. The abstract class `SEServiceBase` defines the behavior of each service, except for the operation `firePropertyChange`, which must be implemented in each individual specialization of the `SEServiceBase` abstract class. The concrete classes `SERendererCoapService`, `SERendererWebSocketService`, `SERendererUPnPService`, and `SERendererMqttService` allow the exposure of the implemented services respectively through the protocols CoAP, WebSocket, UPnP, MQTT. Owing to a payload limitation in CoAP, there is a directional dependency from `SERenderer-erCoapService` to `SERendererWebSocketService` so that the metadata is received through WebSocket to overcome this issue. The other services operate without restriction when using CoAP.

Figure 8 – Excerpt of the PlaySEM SER 2's class diagram for services. The generalization is *incomplete*—other classes of services can be added.



Source: (SALEME; SANTOS; GHINEA, 2019b).

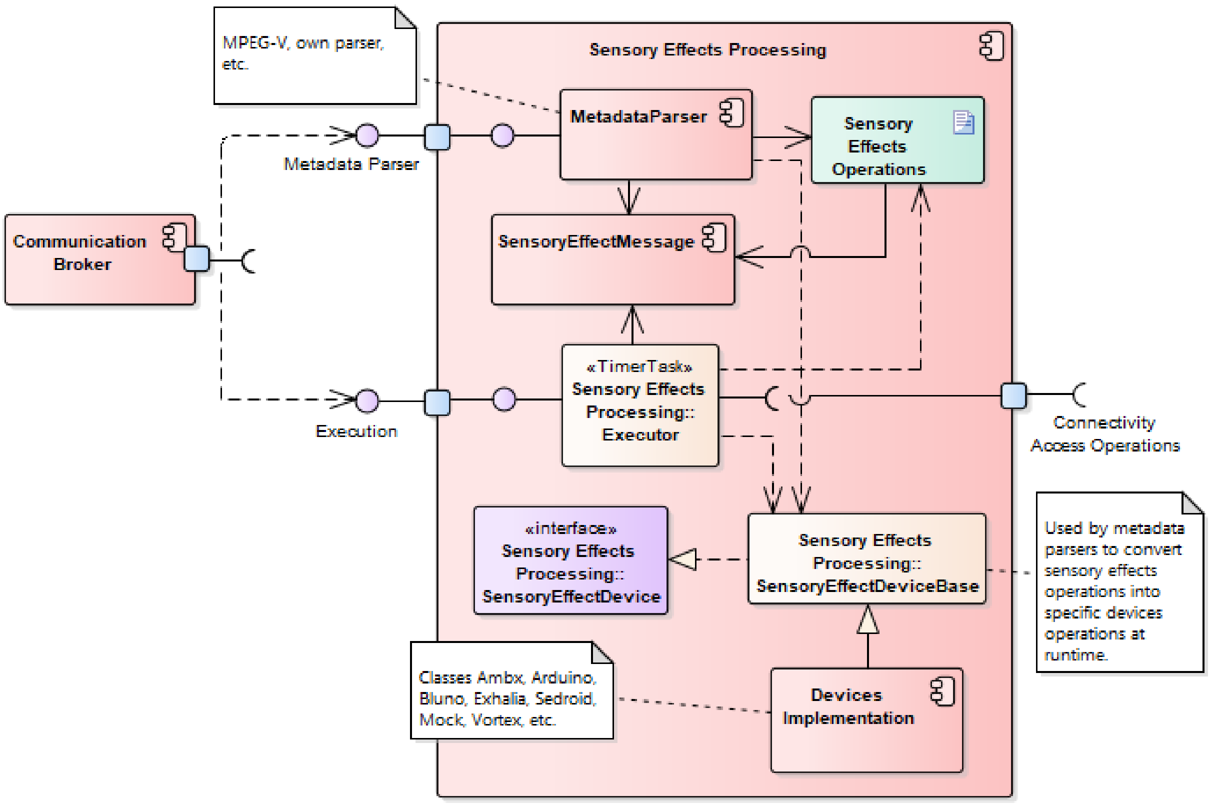
To add support for future protocols, a new class is included extending the `SEService-Base` abstract class and described in the configuration file (Section 4.5). In the case of new common services, they should be implemented in the abstract class so that their behavior is propagated to subclasses, not affecting the structure of PlaySEM SER 2. The incorporation of varied communication protocols allows multimedia applications to choose the best interface for them not only based on performance, but also on their own requirements and constraints. For instance, manipulating WebSockets on web applications would require less effort than UPnP.

4.3.2.2 Processing, Standardization and Devices' Implementation

After establishing communication with PlaySEM SER 2 and conveying the SEM, the `Sensory Effects Processing` layer has to interpret and convert metadata into messages to control the devices. The messages are non-device-dependent, which means that the `Meta-data parser` is able to queue generic messages related to `Lighting message`, `Wind message`, `Vibration message`, and `Scent message` to subsequently deliver them to the specific devices configured in PlaySEM SER 2. `Other messages` types can be added to support other kinds of sensory effects. This isolation level gives flexibility for mulsemedia systems to work with any standard and any device. Natively, PlaySEM SER 2 supports the MPEG-V standard through the aid of the `MPEG-V metadata library`. `Other standard libraries` can be added to deal with future standards. Also, in this layer, specific device drivers are implemented. Though most devices require a specific connectivity protocol, at this point, no connectivity is provided, which means that the same device can be connected by reusing different protocols. The implementation of the drivers is concerned with how to convert the queue teemed with generic messages into specific commands for the devices. Furthermore, they can make a bridge between PlaySEM SER 2 and SDKs and APIs by implementing means of interfacing with them. Thereby, PlaySEM SER 2 is able to embrace as many devices as it can from different brands and with distinct particularities within this layer.

A component diagram presented in Figure 9 shows the interconnection between the communication broker and the sensory effects processing.

Figure 9 – Component diagram to represent the interconnection between the communication broker and the sensory effects processing provided by the framework.

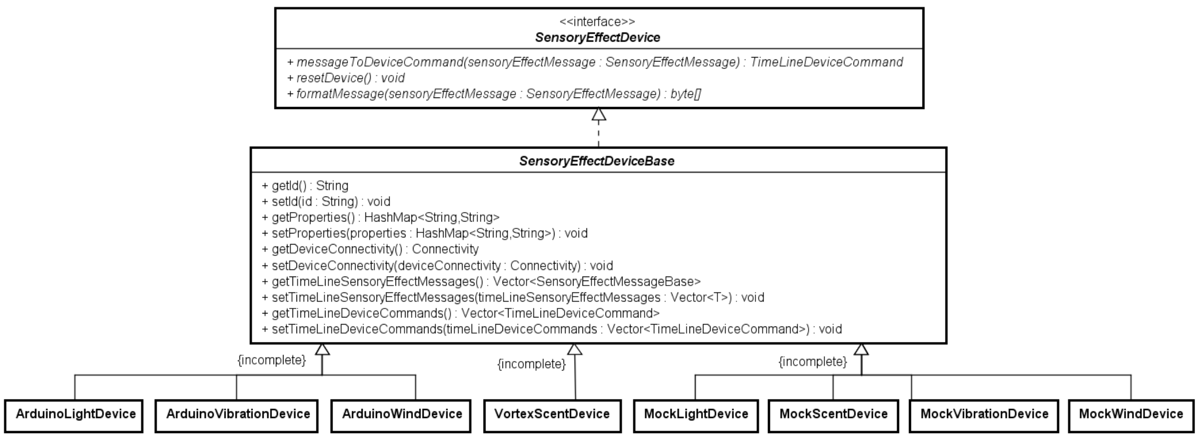


Source: Created by the author.

The `Communication Broker` requires an interface to pass requests to the `Sensory Effects Processing`. The latter provides interfaces to parse SEM metadata and control sensory effect devices. `MetadataParser`, which represents the implementation of an MPEG-V or an own parser, is associated with `SensoryEffectMessages` to convert SEM into generic messages and `Sensory Effects Operations`, where conversions are stored. `MetadataParser` depends on the class `SensoryEffectDeviceBase`, which implements an interface `SensoryEffectDevice`, and is used by the component `MetadataParser` to convert the generic messages into specific ones when the system runs. The interface to control sensory effect devices is provided by the class *Executor*, which reads `Sensory Effects Operations` for inherited classes of `SensoryEffectDeviceBase` and requires an interface from the connectivity layer to access and operate the sensory effect devices.

A key aspect of SEM standardization is related to the strategy for parsing metadata. A specialized algorithm is created for each standard, keeping the same generic operations. For parser strategies, there is no need to implement neither to extend classes. Types of effects not supported by the SER can be included within the metadata parsers. Before running PlaySEM SER 2, the parser must be indicated as well (see Section 4.5). Regarding the implementation of the specific devices, it is akin to the scheme used for services. Figure 10 presents an excerpt from PlaySEM SER 2's class diagram for devices.

Figure 10 – Excerpt of the PlaySEM SER 2's class diagram for devices. Once the generalization is *incomplete*, other classes of devices can take part.



Source: (SALEME; SANTOS; GHINEA, 2019b).

The interface `SensoryEffectDevice` rules which operation will be implemented in all specific devices. The abstract class `SensoryEffectDeviceBase` provides the implementation of a set of common operations for each device. These operations are mainly concerned with aspects such as converting patterns of messages, retrieving specific properties for each device, and linking them to predefined modes of connectivity. The concrete classes `ArduinoLightDevice`, `ArduinoVibrationDevice`, `ArduinoWindDevice`, and `VortexScentDevice` have a particular implementation considering specific outputs. The classes `MockLightDevice`, `MockVibrationDevice`, `MockWindDevice`, and `MockScentDevice` are predefined to use the console to output the outcome of the processing for the purpose of simulation (see Section 4.6).

To expand PlaySEM SER 2 to support other devices, a new specific device class should be created extending the `SensoryEffectDeviceBase` abstract class bearing in mind that the operations specified in the interface `SensoryEffectDevice` must be coded. A description of the added device must be specified in the configuration file (Section 4.5). A particular feature allows developers not to be concerned about connectivity and focus only on the commands to control the new device. Through configuration, it is possible to reuse implemented connectivities, as presented in Section 4.3.2.3. In case that a device cannot avoid the use of either SDKs or APIs, an extension of `SensoryEffectDeviceBase` can be created to interface with them.

4.3.2.3 Connectivity with Devices

The `Connectivity Interface` allows PlaySEM SER 2 to establish connections with multiple and heterogeneous devices. A set of protocols such as `USB/Serial`, `Bluetooth`, and `Wi-Fi` are supported. `Zigbee` is suggested due to the fact that this protocol is used in many portable devices. However, it does not hinder the expansion of `Other protocols` such as Sigfox and 6LowPAN, for instance. To exchange messages between the devices and PlaySEM SER 2's `Connectivity Interface`, communication protocols such as `CoAP` and `MQTT` are suggested when needing a standard, but `Plain text` (text not specially formatted) represented in bytes is commonly used to convey the commands to the devices. `Other protocols` are permitted as well.

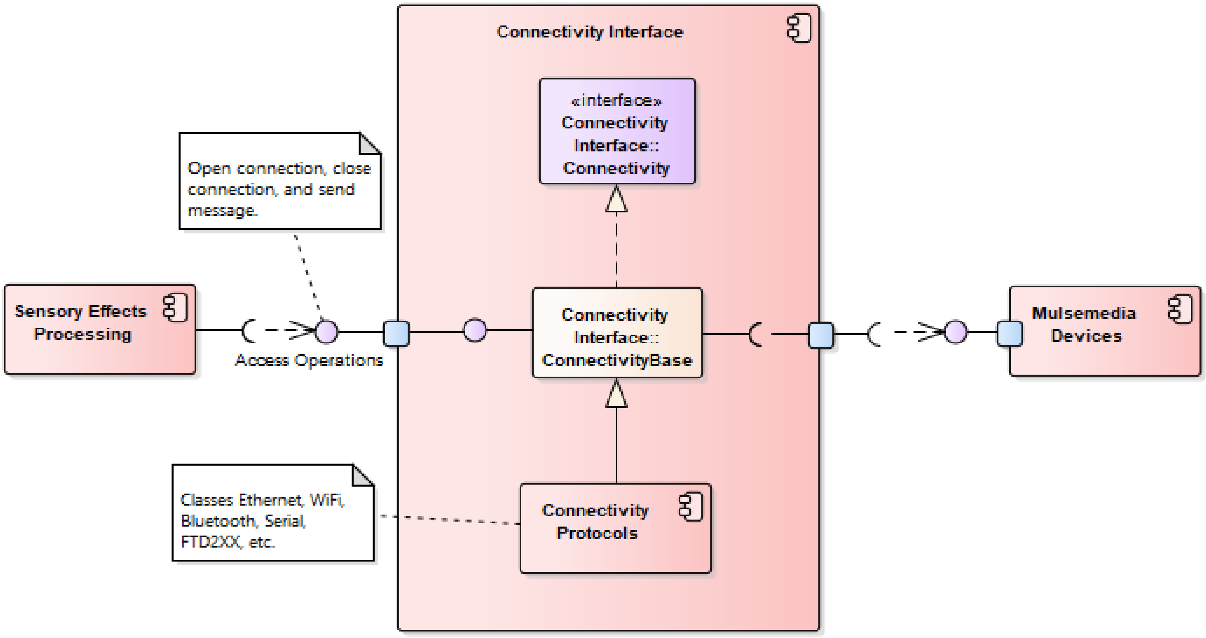
A component diagram that depicts the interconnection between the sensory effects processing, the connectivity interface, and mulsemedia devices, is shown in Figure 11.

`Sensory Effects Processing` requires an interface for connectivity operations, which is provided by the class `ConnectivityBase`. The latter implements an interface `Connectivity` and requires an interface to access and operate `Mulsemedia Devices`.

Figure 12 presents an excerpt from PlaySEM SER 2's class diagram for device connectivities in order to help understanding how to include new connectivity protocols. The interface `Connectivity` plays a role in the shape of a contract that the concrete classes have to sign. Basically, there are three operations to handle a connection, which include opening, closing, and sending a message through the supported protocol. The abstract class `ConnectivityBase` implements some common operations for the connectivity, such as knowing if the connection has been established and getting specific properties for each connectivity, for instance, a virtual port the protocol should use to communicate. The concrete classes `EthernetWifiOutput`, `BluetoothOutput`, `ConsoleOutput`, `FTD2XXOutput`, and `SerialOutput` implement specific directives to establish connections with Ethernet and Wi-Fi, Bluetooth, and USB/Serial devices, proprietaries (such as FTD2XX) or not. The `ConsoleOutput` helps the mock classes described in Section 4.3.2.2 to simulate devices delivering messages on the console on which the system runs.

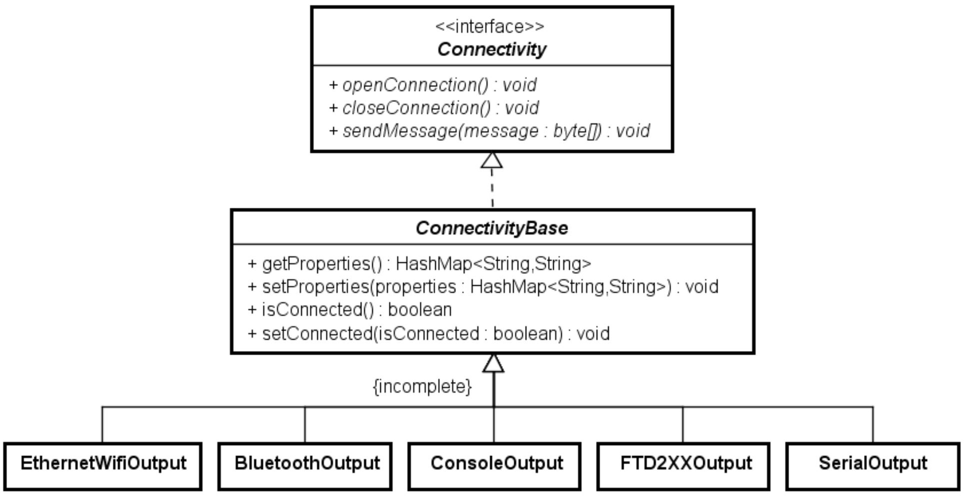
Like the other extensions for services and devices, to include other connectivity protocols

Figure 11 – Component diagram to represent the interconnection between the sensory effects processing and the connectivity interface provided by the framework.



Source: Created by the author.

Figure 12 – Excerpt of the PlaySEM SER 2's class diagram for device connectives. The generalization is *incomplete*, meaning that other connectivity classes can be considered.



Source: (SALEME; SANTOS; GHINEA, 2019b).

it is necessary to extend the abstract class. A new class must be added extending the class `ConnectivityBase`. The operations of the interface `Connectivity` shall be coded. The same pattern should be used for either SDKs or APIs as though they were connectivity means to access the devices. Moreover, the new connectivity has to be described in the configuration file (Section 4.5).

4.3.3 Rendering Devices

Finally, the `Rendering Devices` layer separates the output devices that are responsible for conveying content from the digital to the physical world. `Screen`, `Speakers`, `HMD device`, among others, represent devices to reproduce audiovisual content for multimedia applications, whereas `Lighting device`, `Wind device`, `Vibration device`, `Scent device`, and `Other devices`, represent sensory effects devices connected to PlaySEM SER 2 for integrating multiple sensory effects in mulsemedia applications.

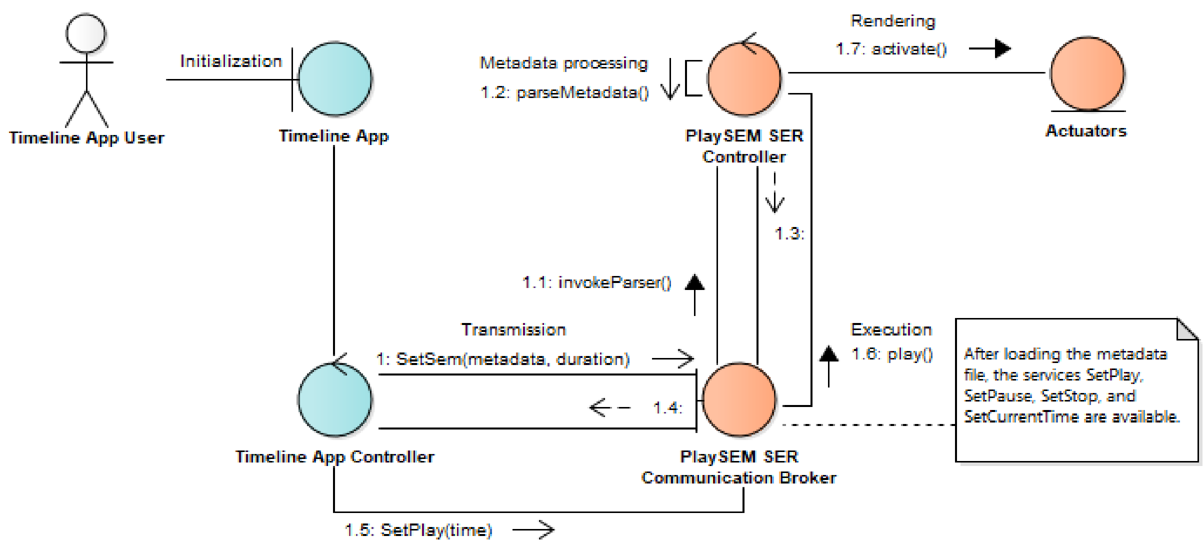
4.4 Interaction with PlaySEM SER 2

PlaySEM SER 2 is designed so that it can be seamlessly integrated with timeline- and event-based multimedia applications. This section describes the collaboration that must be woven between multimedia applications and PlaySEM SER 2 by using the services provided by the latter (see Table 2). The focus is predominantly on the interactions between each other.

4.4.1 Timeline Applications

The devices responsible for rendering sensory effects are synchronized with a continuous medium in the virtual world, i.e. movies, songs, and so on. As soon as a medium is open, the SEM file shall be sent to PlaySEM SER 2. Additionally, at every key moment in which the user is interacting with the interface such as playing, pausing, stopping, forwarding, or rewinding a medium, the same information must be communicated to the mulsemedia system. Figure 13 outlines this process of collaboration.

Figure 13 – Interaction between timeline multimedia applications and PlaySEM SER 2.



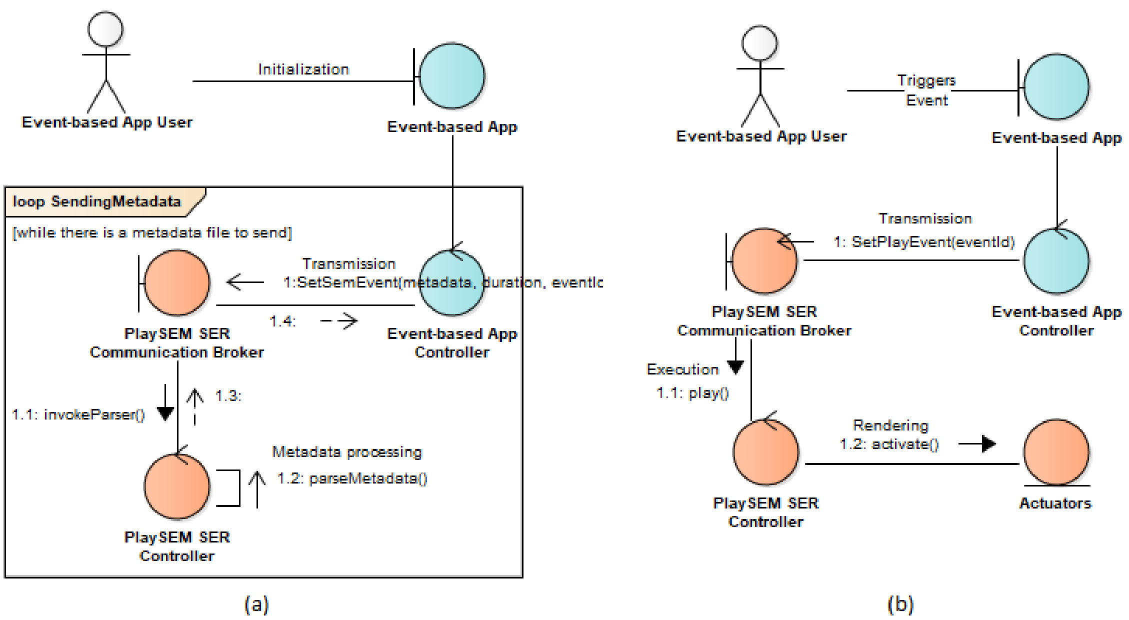
Source: (SALEME; SANTOS; GHINEA, 2019b).

After initializing the `Timeline App`, the `Timeline App Controller` conveys the SEM to the *PlaySEM SER Communication Broker* through the service `SetSem`. After that, the `PlaySEM SER Controller` parsers the metadata converting it into specific commands for controlling the devices and notifies the `PlaySEM SER Communication Broker` to relay this information to the `Timeline App Controller`. Following that, the service `Set-Play` should be transmitted from the `Timeline App Controller` to the `PlaySEM SER Communication Broker` to notify the `PlaySEM SER Controller` that the sensory effects can be rendered. Other services could be called here, always obeying the prerequisites described in Table 2.

4.4.2 Event-based Applications

Contrary to the interaction for timeline applications, the process for event-based applications encompasses two steps. The goal is to reduce repetitive tasks and the number of exchanged messages through the network (SALEME; SANTOS; GHINEA, 2018). Figure 14, item (a), presents the first step.

Figure 14 – Interaction between event-based multimedia applications and PlaySEM SER 2 split in (a) loading process, and (b) interacting process.



Source: (SALEME; SANTOS; GHINEA, 2019b).

As soon as the `Event-based App Controller` and the `PlaySEM SER Communication Broker` establishes a connection, the metadata for each event should be transmitted in a loop through the service `SetSemEvent` before the user starts interacting with the multimedia application. Next, the `PlaySEM SER Controller` parsers the metadata converting it into specific commands for controlling the devices, stores it in a queue identified by the id of the event, and notifies the `PlaySEM SER Communication Broker` to relay this information to the `Event-based App Controller`. In the second step (Figure 14, item b), as soon as the `Event-based App` receives a stimulus, the service `SetPlayEvent` is conveyed to the `PlaySEM SER Communication Broker`, which in turn, seeks the event id and sparks the devices promptly.

4.5 Configuration Strategies

This section describes how to customize PlaySEM SER 2 to operate under different settings by setting it up accordingly. It also allows the system to have different behaviors dynamically without changing its code. All the settings are specified within the file `SERenderer.xml`, which is explained in the following sections and is fully presented in APPENDIX B.

4.5.1 Initialization and Setup

Before starting PlaySEM SER 2, the setup should be informed. The excerpt below shows which parameters should be adjusted.

```
<!-- Main configuration -->
<communicationServiceBroker>upnpService</communicationServiceBroker><!--
   COMMUNICATION INTERFACE TO BE USED -->
<metadataParser>mpegvParser</metadataParser><!-- STANDARD TO BE
   CONSIDERED WHEN PARSING METADATA -->
<lightDevice>mockLight</lightDevice><!-- DEVICE FOR RENDERING LIGHT -->
<windDevice>mockWind</windDevice><!-- DEVICE FOR RENDERING WIND -->
<vibrationDevice>mockVibration</vibrationDevice><!-- DEVICE FOR
   RENDERING VIBRATION -->
<scentDevice>mockScent</scentDevice><!-- DEVICE FOR RENDERING SCENT -->
<debugMode>true</debugMode><!-- PRINT MESSAGES ON CONSOLE --> ...
```

The attribute `communicationServiceBroker` gives a choice between different communication protocols (see Section 4.3.2.1). An id such as `upnpService` should be filled in for this attribute. It also requires a strategy such as `mpegvParser` to parse metadata according to a sensory effects metadata standard. As PlaySEM SER 2 currently supports four types of sensory effects, the attributes `lightDevice`, `windDevice`, `vibrationDevice`, and `scentDevice` must be informed. Finally, the attribute `debugMode` specifies whether debug messages will be printed in the console whereby the system runs. The ids filled out in each attribute must be declared in the same file, which contains the specification for the replaceable components used in the system (see Section 4.5.3).

4.5.2 Communication, Connectivity, and Temporal Aspects

By introducing a network and considering sensory effects metadata processing time in mulsemedia systems, a set of other hurdles emerge, such as network delay, jitter, packet loss, sensory effects metadata transmuting, timetable building, among others, which can have an impact on the quality of experience of users. Therefore, timing issues within the integration of different applications to deliver sensory effects must be considered so that they do not affect performance, as portrayed in Section 3.1.2. Given the plurality of the users' acceptable times (Section 2.3), a mulsemedia system should be able to consider the delay introduced by different technologies. Furthermore, some sensory effects such as scent, wind, and flavor have a tendency to linger (in the atmosphere or on the tongue). Therefore, if the mulsemedia system is able to calibrate and adapt the delivery of the sensory effects accordingly, it will produce better results.

PlaySEM SER 2 takes it into consideration by allowing a manual adjustment in time considering the technologies used to deliver the sensory effects. This is processed in execution time when the processing is being performed. The system will work on its main timetable to compensate potential delays introduced by the components chosen for a specific profile. Within the declaration of each device, communication, and connectivity protocol, there is an attribute called `Delay`, as shown in the following excerpt. It shall be adjusted in milliseconds. To this end, times provided in the literature and in devices specifications should be considered.

```
<device>
   <id>exhaliaScentDevice</id>
   ...
   <properties>
      <delay>800</delay><!-- DEFINE IT IN MS -->
   </properties>
</device>
```

4.5.3 Protocols and Properties

The declaration of the supported communication protocols, devices, and connectivity protocols and the addition of new ones as a way of expanding the support for new technology in PlaySEM SER 2 are linked to the file `SERenderer.xml`. The following excerpt shows how to describe a device and a connectivity type and how to connect one to another. The attribute `devices` represents a list of supported devices. A `device` must be described by filling in its `id`, used in the main configuration (see Section 4.5.1), `deviceClass`, for dynamic instantiation, and `connectivityInterface`, which refers to an id of a predeclared connectivity interface. The list of `connectivityInterfaces` is composed of the supported connectivity protocols, represented by an `id`, to identify the protocol, a `connectivityInterfaceClass`, for dynamic instantiation, if it requires `earlyConnection`, which indicates if the connection will be established when opening the system or on-demand, and finally, a list of `properties`, which may contain zero or more properties to be used by the specific class that implements a connectivity protocol (see Section 4.3.2.3). For instance, one could indicate which slot within a scent machine contains a determined aroma.

```
<!-- Supported devices -->
<devices>
   <!-- Arduino Devices (Light, wind, vibration) -->
   <device>
      <id>arduinoLightDevice</id>
      <deviceClass>
         br.ufes.inf.lprm.sensoryeffect.renderer.device.arduino.
            ArduinoLightDevice
      </deviceClass>
      <connectivityInterface>Serial</connectivityInterface><!-- SET
         CONNECTIVITY INTERFACE -->
   </device> ...
<!-- Connectivity interfaces -->
<connectivityInterfaces>
   <connectivityInterface>
      <id>Serial</id><!-- CONNECTIVITY MUST BE DECLARED TO BE USED -->
      <connectivityInterfaceClass>
         br.ufes.inf.lprm.sensoryeffect.renderer.connectivity.serial.
            SerialOutput
      </connectivityInterfaceClass>
      <earlyConnection>true</earlyConnection>
      <properties>
         <serialPort>COM3</serialPort>
      </properties>
   </connectivityInterface> ...
```

The same pattern applies to the communication services and metadata parsers. There is a list of `communicationServices` that registers all the currently supported communication protocols. An `id` should be provided to identify the scheme, and a `communicationServiceClass`, for dynamic instantiation. The same goes for the catalogue of `metadataParsers`. Section 4.5.1 shows how to set a strategy for each one before starting the system.

```
<!-- Communication services -->
<communicationServices>
   <communicationService>
      <id>coapService</id>
      <communicationServiceClass>
         br.ufes.inf.lprm.sensoryeffect.renderer.service.coap.
            SERendererCoapService
      </communicationServiceClass>
   </communicationService> ...
</communicationServices>
```

```
<!-- Metadata parsers -->
<metadataParsers>
   <metadataParser>
      <id>mpegvParser</id>
      <metadataParserClass>
         br.ufes.inf.lprm.sensoryeffect.renderer.metadata.parser.mpegv.
            MPEGVSEMParser
      </metadataParserClass>
   </metadataParser>
</metadataParsers> ...
```

4.6 Debug and Simulation

In order to check the integration between multimedia applications and PlaySEM SER 2, a mechanism for aiding the developers to find defects is offered. Debug refers to the process of finding issues with the code and repairing them. In the context of PlaySEM SER 2, this mechanism outputs messages from key moments such as receiving data and processing to follow up with the execution. It makes the process of discovering problems easier, finding which part of the code it refers to, and fixing the problems. It can be switch on/off by configuration before the system runs (see Section 4.5.1).

Another mechanism to find problems and correcting them is through the use of mock classes. They work like virtual devices and use the console as a mean of connectivity to print the outcome of the commands. For instance, instead of releasing a real scent at a specific time of a movie, a scent mock class would print the aroma and its intensity at this point. It is also useful to simulate the results of a metadata file. Nonetheless, new mock classes could be created to send commands to virtual environments that simulate the sensory effects. PlaySEM SER 2 provides by default one mock class for each one of the supported types of effects (see Section 4.3.2.2).

4.7 Concluding Summary

This chapter introduced PlaySEM SER 2, a framework that follows the precepts of the conceptual architecture for mulsemedia delivery systems presented in Chapter 3. The framework addresses software engineering aspects to support different end-user applications and sensory effect devices. This framework supports multi-communication and multi-connectivity protocols, multi-standards, and allows the accommodation of new technology relying on its set of architectural and design patterns to be applied successfully. It provides services to mulsemedia applications through a transparent communication broker and is able to work with timeline- and event-based applications. Moreover, it presents a set of configurable parameters to customize the solution according to the needs imposed by mulsemedia applications and offers a mechanism to calibrate delay for each component it works with. In order to test new implementations on top of PlaySEM SER 2, it also provides a debug and flexible simulation scheme.

To the extent that technology for delivering sensory effects evolves, changeless systems may become disposable hence costly. PlaySEM SER 2 does not constrain the design of mulsemedia systems. Although the framework deals with systems heterogeneity internally, it is not antagonistic to SDKs and APIs. Therefore, PlaySEM SER 2 can accommodate them to control devices if necessary.

Currently, the framework does not support content/context adaptation according to user preferences. Alternatively, it could be solved inside mulsemedia applications themselves. Another way to cope with it would be through the implementation of some standardization, such as MPEG-V Part 2, which provides means to describe user preferences. However, an input mechanism for supporting user preferences would have to be provided by multimedia applications, whereas mulsemedia renderers would need to interpret this metadata and adapt rendering devices accordingly.

Support for more than one device for the same sensory effect at the same time can be implemented. However, this requires a change in classes of devices. For example, if there is a wind device declared in your configuration file, but there are two wind fans in the environment, the class that manages the wind device can consider two outputs. To overcome this issue, future work needs to be undertaken to support a list of devices for each sense rather than just one class. This would diminish coupling and increase cohesion.

Further issues are related to safety and security—they have not been tackled yet. Owing to the fact that the incorporation of taste in mulsemedia applications is still very much in its infancy, support for this kind of sensory effect has not been implemented yet. Nonetheless, as soon as work on this effect advances, it can be added to PlaySEM SER 2 by following the design strategies described in this chapter. A more important issue is related to the state of awareness in mulsemedia systems. For example, the perception of heat in cold environments would be different from warmer environments. This implies the calibration of the environment by capturing its state through sensors and adjusting the sensory effects in execution time, which is not supported by the framework yet.

In the next chapter, results from technical and subjective experiments over the framework are described and the findings discussed.

5 Experiments and Results

This chapter describes and discusses results on this work under varied outlooks. First, case studies present heterogeneous real-world scenarios of usage in which the conceptual architecture is materialized into a framework in order to empirically validate it. Following that, a study on temporal aspects related to the communication between event-based applications (where delay plays a role) and the framework is presented. Different communication protocols are analyzed in order to provide time references for the design of mulsemedia systems. Then, the chapter ends with a subjective experiment whereby users evaluated an existent monolithic mulsemedia system and this framework. From their reported answers, a statistical comparison is made to report whether the latter's architecture has an impact on user-perceived QoE or it is indifferent.

5.1 Case Studies on Variability with PlaySEM SER 2

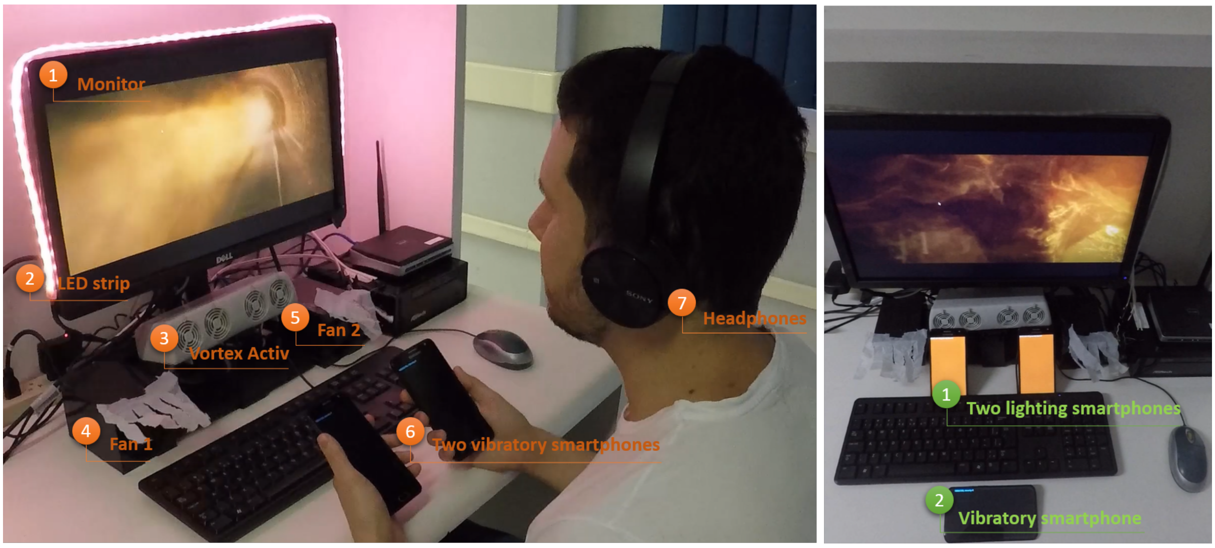
Saleme, Santos, and Ghinea (2019a) presented different scenarios of usage in which evolutions of PlaySEM SER have dealt with variability in mulsemedia systems. With an emphasis on PlaySEM SER 2, this section introduces three case studies (two from (SALEME; SANTOS; GHINEA, 2019b)) where the current framework has been used to cope with heterogeneity at different levels. Furthermore, details on their configurations are also described. The aim is to make evident PlaySEM SER 2's capability of adapting to different conditions changing its parameters.

5.1.1 Video Clip Enriched with External Light, Smell, Vibration, and Wind

This mulsemedia system is composed of DIY (Do It Yourself) devices, smartphones, and a commercial scent device called Vortex Activ. A video clip annotated with MPEG-V is played on the PlaySEM SE Video Player and displayed on a monitor. Its sensory effects are provided to PlaySEM SER 2 following the interaction described in Section 4.4.1. Figure 15 shows the output devices in the environment. On the left side, there are seven elements that work together using heterogeneous technologies to deliver audiovisual (monitor and headphones), lighting (LED strip), wind (fan 1 and 2), vibration (two smartphones), and scent (Vortex Activ) effects. On the right side, the sensory effects are the same. However, the output comes from different devices using other technologies, that is, lighting is displayed on two smartphones, whereas vibration is delivered by just one smartphone.

The connection with audiovisual devices is managed by the multimedia application whereas the sensory effects are of PlaySEM SER 2's responsibility. The following excerpt from

Figure 15 – Video clip enhanced with lighting, wind, vibration, and scent effects. On the left side, the first set of devices to deliver these sensory effects. On the right side, similar devices that deliver the same effects using different technologies.



Source: (SALEME; SANTOS; GHINEA, 2019b).

PlaySEM SER 2's configuration file presents a list of parameters that will be instantiated for the scenario of usage on the left side of Figure 15. This shall be set up as described in Section 4.5.1. The communication between the multimedia application and PlaySEM SER 2 is established via UPnP and the metadata parser used is to deal with MPEG-V.

```
<!-- Main configuration -->
<communicationServiceBroker>upnpService</communicationServiceBroker>
<metadataParser>mpegvParser</metadataParser>
<lightDevice>arduinoLightDevice</lightDevice>
<windDevice>arduinoWindDevice</windDevice>
<vibrationDevice>sedroidVibrationDevice</vibrationDevice>
<scentDevice>vortexScentDevice</scentDevice>
```

The devices provided in the initial configuration come from the next excerpt. The Arduino-based light and wind device use a serial connectivity interface with PlaySEM SER 2. The two smartphones that deliver vibrations are connected via Bluetooth whereas the scent device uses a proprietary serial connectivity interface (SerialFTD2XX) for chips manufactured by FTDI, which is embedded in Vortex Activ.

```
<!-- Lighting device -->
<device>
   <id>arduinoLightDevice</id>
   <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.arduino.
      ArduinoLightDevice</deviceClass>
   <connectivityInterface>Serial</connectivityInterface> ...
<!-- Wind device -->
```

```
<device>
   <id>arduinoWindDevice</id>
   <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.arduino.
      ArduinoWindDevice</deviceClass>
   <connectivityInterface>Serial</connectivityInterface> ...
<!-- Vibration device -->
<device>
   <id>sedroidVibrationDevice</id>
   <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.sedroid.
      SedroidVibrationDevice</deviceClass>
   <connectivityInterface>Bluetooth</connectivityInterface> ...
<!-- Scent device -->
<device>
   <id>vortexScentDevice</id>
   <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.vortex.
      VortexScentDevice</deviceClass>
   <connectivityInterface>SerialFTD2XX</connectivityInterface> ...
```

For each connectivity interface, specific properties are indicated in order to supply PlaySEM SER 2 with information about where to send messages and which classes it should use to handle them. For instance, the serial connectivity uses port ’COM3’ under baud rate ’9600’. For the proprietary serial connectivity, an *id* (67330049) is provided to match to the scent device that is plugged. For the Bluetooth connection, the number of devices that will be connected and their URL for connection are provided. The smartphones are Android-based ones which run software that listens to connections and render sensory effects according to the messages received from PlaySEM SER 2.

```
<!-- Serial connectivity -->
<connectivityInterface>
   <id>Serial</id>
   <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.renderer.
      connectivity.serial.SerialOutput</connectivityInterfaceClass>
   <properties>
      <serialPort>COM3</serialPort>
      <baudRate>9600</baudRate> ...
<!-- Serial FTD2XX -->
<connectivityInterface>
   <id>SerialFTD2XX</id>
   <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.renderer.
      connectivity.serial.FTD2XXOutput</connectivityInterfaceClass>
   <properties>
      <device01-id>67330049</device01-id> ...
<!-- Bluetooth -->
<connectivityInterface>
   <id>Bluetooth</id>
   <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.renderer.
```

```
        connectivity.bluetooth.BluetoothOutput</connectivityInterfaceClass>
    <properties>
        <number-of-devices>2</number-of-devices>
        <device01-connection-url>btspp://A816D0063584:5;authenticate=false;
            encrypt=false;master=false</device01-connection-url>
        <device02-connection-url>btspp://A816D006351E:5;authenticate=false;
            encrypt=false;master=false</device02-connection-url> ...
```

The following fragment of PlaySEM SER 2's configuration file shows the changes in the scenario of usage on the right side of Figure 15. Just as in the previous scenario, both the light and vibration devices are Android-based smartphones.

```
<!-- Main configuration -->
<lightDevice>sedroidLightDevice</lightDevice>
<vibrationDevice>sedroidVibrationDevice</vibrationDevice>
```

Whereas in the previous scenario just Bluetooth was used to connect to the smartphones, under this scenario Wi-Fi is configured to deliver lighting effects as follows.

```
<!-- Lighting device -->
<device>
    <id>sedroidLightDevice</id>
    <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.sedroid.
        SedroidLightDevice</deviceClass>
    <connectivityInterface>WiFi</connectivityInterface> ...
<!-- Vibration device -->
<device>
    <id>sedroidVibrationDevice</id>
    <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.sedroid.
        SedroidVibrationDevice</deviceClass>
    <connectivityInterface>Bluetooth</connectivityInterface> ...
```

Analogous to the first scenario, for each connectivity interface, communication parameters are supplied. The following excerpt presents parameters for the WiFi connection that requires the number of devices and the address of them. For Bluetooth, as a different device was used to the former case, a new URL was provided and just one device was set up.

```
<!-- Wi-Fi -->
<connectivityInterface>
    <id>WiFi</id>
    <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.renderer.
        connectivity.wifi.WiFiOutput</connectivityInterfaceClass>
    <properties>
        <number-of-devices>2</number-of-devices>
        <device01-address>90.0.0.102:8080</device01-address>
        <device02-address>90.0.0.107:8080</device02-address> ...
<!-- Bluetooth -->
```

```
<connectivityInterface>
   <id>Bluetooth</id>
   <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.renderer.
      connectivity.bluetooth.BluetoothOutput</connectivityInterfaceClass>
   <properties>
      <number-of-devices>1</number-of-devices>
      <device01-connection-url>btspp://4088058F7E2E:5;authenticate=false;
         encrypt=false;master=false</device01-connection-url> ...
```

This case study comes to light to show the flexibility of PlaySEM SER 2 in making slight changes in its configuration to accommodate different profiles of usage with heterogeneous interfaces for connectivity. This has proven useful to take advantage of multifaceted devices such as smartphones to provide sensory effects. Furthermore, this made evident the needlessness to either change the framework's code or install additional SDKs and APIs that eventually could be required for some devices.

5.1.2 Smell-intensive System

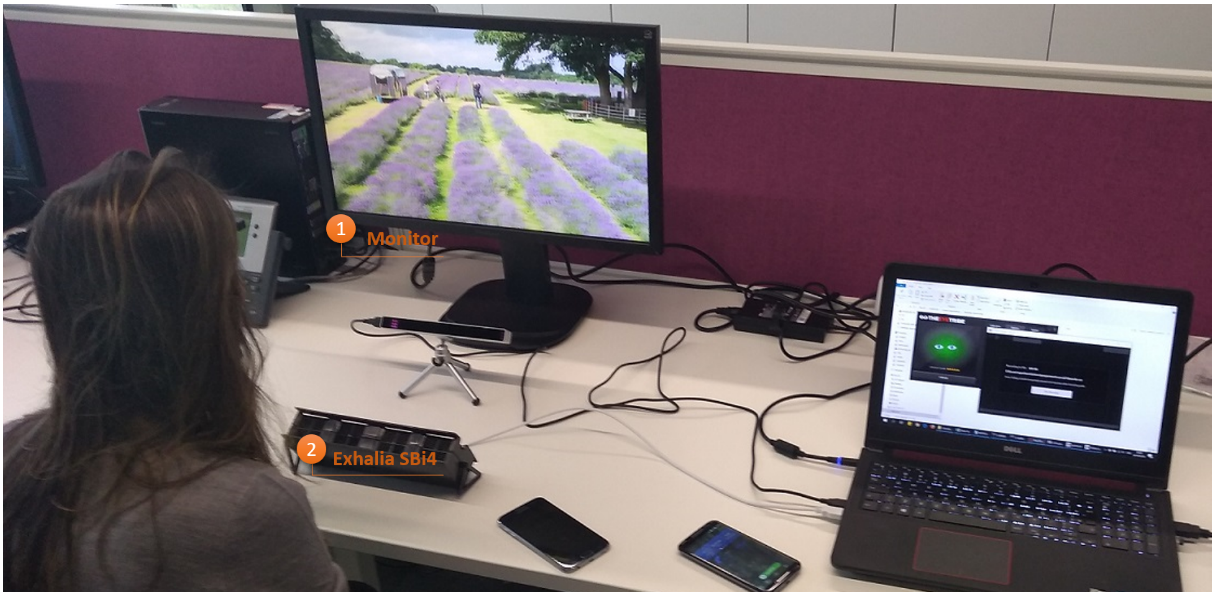
This scenario of usage involves just one scent device to deliver different intensities of smell—Exhalia SBi4. This is another proprietary device that can be managed by PlaySEM SER 2. However, the framework goes beyond the SDK provided by Exhalia and provides a mechanism to switch on all of its four fans that blow scent crystals to deliver the smell. Figure 16 shows this environment. There is a monitor that presents a video clip annotated with MPEG-V running on PlaySEM SE Video Player and a scent device. PlaySEM SER 2 listens to the video player, processes SEM, and deals with the scent device in this case. As in the former section, the interaction between the multimedia application and PlaySEM SER 2 is as described in Section 4.4.1.

The following fragment displays Exhalia SBi4 set up for the initialization in conformance with what is described in Section 4.5.1. The other devices apart from the scent device are mock classes that exhibit the outcome in an output console if their sensory effect types are included in the presentation.

```
<!-- Main configuration -->
<communicationServiceBroker>upnpService</communicationServiceBroker>
<metadataParser>mpegvParser</metadataParser>
<lightDevice>mockLight</lightDevice>
<windDevice>mockWind</windDevice>
<vibrationDevice>mockVibration</vibrationDevice>
<scentDevice>exhaliaScentDevice</scentDevice>
```

Exhalia SBi4 is connected to the framework through a USB interface as parameterized in the next excerpt. A property called 'increasedIntensity' determines whether the scent device will emit a specific scent considering all the slots for cartridges for increased intensity. In case it

Figure 16 – Smell-intensive system—audiovisual content along with scent delivered through four emitters at the same time using USB connectivity.



Source: (SALEME; SANTOS; GHINEA, 2019b).

is ‘false,’ the scent device works with one cartridge of scent at a time under the same intensity.

```
<!-- Scent device -->
<device>
   <id>exhaliaScentDevice</id>
   <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.exhalia.
      ExhaliaScentDevice</deviceClass>
   <connectivityInterface>Usb</connectivityInterface>
   <properties>
      <increasedIntensity>true</increasedIntensity> ...
```

Then, USB properties to reach and control Exhalia SBi4 are given. They should be collected from the operating system and then set up properly on PlaySEM SER 2 as follows.

```
<!-- USB -->
<connectivityInterface>
   <id>Usb</id>
   <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.renderer.
      connectivity.usb.UsbOutput</connectivityInterfaceClass>
   <properties>
      <vendorId>0x1781</vendorId>
      <productId>0x07D0</productId>
      <usbInterface>0</usbInterface>
      <inEndPoint>81</inEndPoint>
      <outEndPoint>1</outEndPoint>
      <timeout>5000</timeout> ...
```

Just like in the previous section, the framework rules out the use of SDKs and APIs to control the proprietary device in the smell-intensive system. By adding a mechanism not supported by the supplier to emit scent through all the cartridges at the same time, it increases the utility of the device. Furthermore, from the preceding case study to this one, the only difference with regard to PlaySEM SER 2 is that a configuration profile was created to meet the specific needs for this scenario of usage, discarding code changes.

5.1.3 360° VR Mulsemedia System

This mulsemedia system is a 360° VR-based one and provided support to the work of Covaci et al. (2019b) and Comsa et al. (2019). An HMD with a portable scent diffuser attached to it and a wind device are employed to engage users in a 360° environment augmented with smell and airflow. Video clips annotated with MPEG-V are played on 360° Unity mobile application installed on a smartphone attached to the HMD. Like the previous case studies, sensory effects are provided to PlaySEM SER 2 following the interaction described in Section 4.4.1. Figure 17 displays the set of output devices put on a user. The Samsung Galaxy S6 smartphone attached to a Samsung Gear VR glass reproduces audiovisual content and the sensory effects of smell and airflow are delivered respectively by a portable scent diffuser (with scent crystal balls inside and a mini smell fan) and a fan.

Figure 17 – A 360° VR mulsemedia system that includes audiovisual content displayed on an HMD integrated with a fan and a portable scent diffuser.

Source: Created by the author.

The configuration for starting PlaySEM SER 2 in this scenario of usage shall be set up as described in Section 4.5.1. The communication between the 360° Unity mobile application and PlaySEM SER 2 is established via WebSocket and the metadata parser used is to deal with

MPEG-V. The following fragment displays the wind and scent device lined-up with mock classes that exhibit the outcome in an output console for light and vibration.

```
<!-- Main configuration -->
<communicationServiceBroker>websocketService</communicationServiceBroker>
<metadataParser>mpegvParser</metadataParser>
<lightDevice>mockLight</lightDevice>
<windDevice>arduinoWindDevice</windDevice>
<vibrationDevice>mockVibration</vibrationDevice>
<scentDevice>blunoScentDevice</scentDevice>
```

Following that, the next fragment of code presents particulars on the devices and their connectivities. Both Arduino- and Bluno-based devices are set up through two different serial connectivity interfaces but with different identification.

```
<!-- Wind device -->
<device>
   <id>arduinoWindDevice</id>
   <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.arduino.
      ArduinoWindDevice</deviceClass>
   <connectivityInterface>Serial</connectivityInterface> ...
<!-- Scent device -->
<device>
   <id>blunoScentDevice</id>
   <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.bluno.
      BlunoScentDevice</deviceClass>
   <connectivityInterface>Serial2</connectivityInterface> ...
```

Finally, the two serial connectivities are configured to send messages to handle properly the devices. Whereas the wind device uses port 'COM3' under baud rate '9600' in this scenario, the scent diffuser assigns port 'COM4' under baud rate '115200' as follows.

```
<!-- Serial for Arduino-->
<connectivityInterface>
   <id>Serial</id>
   <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.renderer.
      connectivity.serial.SerialOutput</connectivityInterfaceClass>
   <properties>
      <serialPort>COM3</serialPort>
      <baudRate>9600</baudRate> ...
<!-- Serial for Bluno -->
<connectivityInterface>
   <id>Serial2</id>
   <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.renderer.
      connectivity.serial.SerialOutput</connectivityInterfaceClass>
   <properties>
      <serialPort>COM4</serialPort>
      <baudRate>115200</baudRate> ...
```

This scenario of usage showed an innovative 360° VR application integrated to PlaySEM SER 2. By doing so, the framework's variability is evinced from a different perspective of presentation, no more restricted to a monitor as in the previous two case studies. Once more, only proper configurations were performed on PlaySEM SER 2 to support this scenario.

5.2 Analysis of Response Time in Networked Event-based Mulsemedia Systems

Interactive event-based mulsemedia applications are those in which every millisecond is precious, that is, as soon as an event occurs in an interactive application, a mulsemedia renderer has to deliver some types of sensory effects, such as haptics, as swiftly as possible without spoiling users QoE as evinced by the tolerable delays for multisensory interactions reported in Chapter 2 (Section 2.1.2).

The framework described in Chapter 4 brings an optimized way of interacting with event-based applications, introduced originally by Saleme, Santos, and Ghinea (2018). At the time, Saleme and colleagues enhanced the work of Saleme, Celestrini, and Santos (2017), which presented times for the interaction between a gestural interactive application and PlaySEM SER in real-time. Precisely, it was found out a time-range between 27ms and 67ms on average to complete the whole process of recognizing gestures, transmitting signals over a wired network (using UPnP), and processing and rendering sensory effects. Moreover, the authors pointed out aspects that should be improved in mulsemedia systems architecture so as to enhance performance.

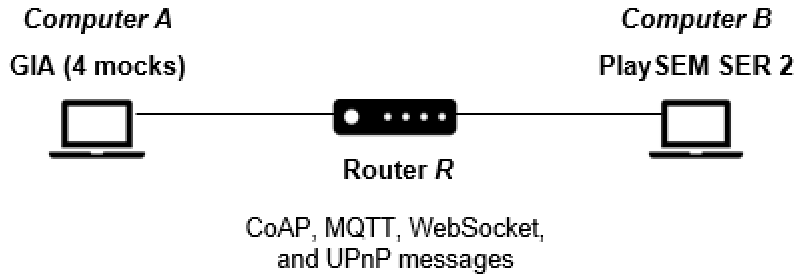
Indeed, when introducing network elements in mulsemedia systems, there is a concern that its resources (data, protocols, etc.) can have a negative impact on performance. Therefore, the analysis presented in this section investigates how PlaySEM SER 2 performs under different protocols (CoAP, MQTT, UPnP, and WebSocket) over the communication broker, making a comparison between them, in order to provide time references for the design of mulsemedia systems.

5.2.1 Materials

5.2.1.1 Devices

The hardware setup used in the experimental evaluation consists of two laptops and a router. The first laptop was an Intel Core i7-6700HQ, 16GB RAM, 254GB SSD, running Windows 10 64 bits, and a simulated gestural interactive application to send MPEG-V scripts to the framework. The second laptop was an Intel Core i7-3537U, 4GB RAM, 128 GB SSD, running Windows 8.1 64 bits. As shown in Figure 18, they were connected at 100 Mbps via cable to the router Wireless Dual Band AC750 Archer C20, which also supports the standards IEEE 802.11ac/n/a at 5GHz and IEEE 802.11b/g/n at 2.4GHz.

Figure 18 – Test-bed of the experiment with for mock applications (one for each protocol), the router, and PlaySEM SER 2.



Source: Created by the author.

5.2.1.2 Protocols

Four protocols were used: COAP, MQTT, UPnP, and WebSocket. The first two are lightweight communication ones recurrently used in real-time Cyber-Physical Systems and the IoT, whereas UPnP was previously supported by PlaySEM SER (SALEME; CELESTRINI; SANTOS, 2017) and WebSocket is often employed in real-time web applications. These protocols are described in detail in Chapter 4 (Section 4.3.2.1).

5.2.1.3 MPEG-V Scripts

The MPEG-V scripts used in this experiment come from the work of Santos, Neto, and Saleme (2015), which presents a multisensory system for theatrical environments whereby the actor triggers sensory effects by gesture on the stage. These scripts describe a set of sensory effects to be rendered after a gesture recognition and are called actions. Table 3 presents them along with their description, type, and number of effects.

5.2.1.4 Software

For the experiment, four applications simulating a gestural interactive application were created, one for each protocol mentioned in Section 5.2.1.2, due to their particular way of working. PlaySEM SER version 2.0.0 was employed, running under a simulated mode, which signifies that the output of the commands for playing sensory effects is exhibited on screen, instead of rendering them with the devices. The following Java implementation of the protocols was used: Californium CoAP framework 2.0.0-M4 (CoAP), Moquette 0.10 (MQTT), Embedded Jetty 9.4.0.v20161208 (WebSocket), and Cling Core 2.0.1 (UPnP). CoAP ran on its *Confirmable Message* mode whereas MQTT on its *Exactly once*. For UPnP and Websocket the QoS level was

Table 3 – Actions and their respective attributes of sensory effects.

		Sensory Effects		
Action	Description	Lighting	Wind	Vibration
ACT02	It turns wind on.	0	1	0
ACT03	It increases wind by 50%.	0	1	0
ACT04	It decreases wind by 50%.	0	1	0
ACT05	It stops wind.	0	1	0
ACT06	It simulates ray of light and earthquake.	5	0	2
ACT07	It generates a rainbow.	8	0	0
ACT08	It stops lighting.	1	0	0
ACT09	It stops vibration.	0	0	1

Source: Created by the author.

set to default. A parameter was defined in the PlaySEM SER's configuration file to indicate which protocol would run in each instance. Wireshark 2.4.2 was used to capture the communication between the mock applications and PlaySEM SER 2 over the network.

5.2.2 Experimental Design

Network protocol was the independent variable with four conditions: CoAP, MQTT, UPnP, and WebSocket. PlaySEM SER 2 was used in all conditions, thus, a related design (repeated measures) was employed for this experiment. The dependent variable was the response time, which in this case means the time spent by an event cue sent from the mock applications until reaching PlaySEM SER 2 over the network.

5.2.3 Procedure

As reported in Chapter 4 (Section 4.4.2), there are two steps for the integration between the event-based applications and PlaySEM SER 2. During the first step, all the MPEG-V scripts are conveyed to the framework and pre-processed in a loop until there are no more scripts. To recap, this pre-processing phase converts SEM into specific commands to handle the sensory effect devices, creating a scheduling list. The response time of this phase is not the object of interest because this process runs when the applications are loading, for instance, at the time in which a splash screen is exhibited. The second step occurs in real-time—an event occurs in an application and triggers sensory effects such as those described in Table 3. It works on the basis of request/response, which consists of sending an identification of the event occurred to deliver the sensory effects. As a result, users may be directly affected by time response depending on the type of interaction and the selected protocols. This is precisely what is measured for each protocol. The following tasks were performed in order to carry out this quantitative experiment.

1. Computer *A* was set up to run the application mocks, whereas computer *B* configured to run the PlaySEM SER 2. Both were connected to a router *R*, which was isolated from the internet and other devices. This was to avert jitter and packet loss successfully.

2. Fifty iterations of the message *SetPlayEvent* were triggered. Once there were 8 actions (Table 3), the total number of iterations was 400.

3. The communication between the applications over the network was captured, and its log stored in a text file.

4. The set of data was converted into tabular sheets. For the CoAP protocol communication, the difference between the message *CON* sent by the client on the endpoint *SetPlayEvent* and the message *ACK* sent by the server was measured. For MQTT, the difference between *Publish Message* on the topic *SetPlayEvent* and *Publish Complete*. For UPnP, the difference between *POST* on the service *SetPlayEvent* and *HTTP/1.1 200 OK* was measured. Finally, for Websocket, the difference between *Websocket Text* containing a JSON (JavaScript Object Notation) object *SetPlayEvent* and the returned message $[ACK]$ was also measured.

5.2.4 Results

Data were analyzed with the Statistical Package for the Social Sciences (SPSS) for Windows version (release 25.0), which provides mechanisms to perform complex statistical data analysis in simple steps. A Friedman test was used to know whether there are any statistically significant differences between the response time distributions of the four protocols (CONOVER, 1998). This test was employed because data were not normally distributed, as assessed by Shapiro-Wilk's test ($p < .0005$) for all protocols and visually inspected through a Q-Q plot (normal quantile) displayed in Figure 19.

Therefore, the null hypothesis for the Friedman test is:

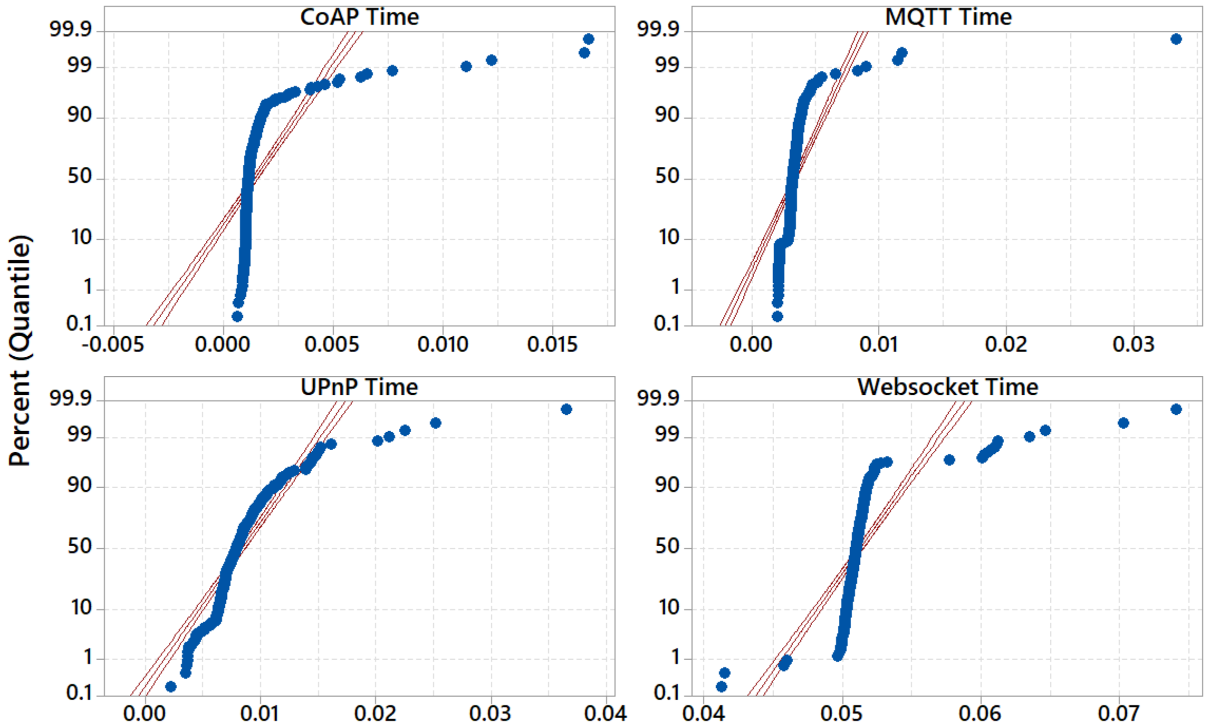
> H_0: CoAP, MQTT, UPnP, and WebSocket response time distributions using PlaySEM SER 2 are the same.

The alternative hypothesis is:

> H_A: at least two of the protocols' response time distributions using PlaySEM SER 2 differ.

A significance level of $p < .05$ was adopted for this study. All figures in this section use individual standard deviations to calculate the intervals unless otherwise stated.

Figure 19 – Q-Q plot for visual inspection of the protocols' sample quantiles to the corresponding theoretical quantiles (Normal 95% CI).



Source: Created by the author.

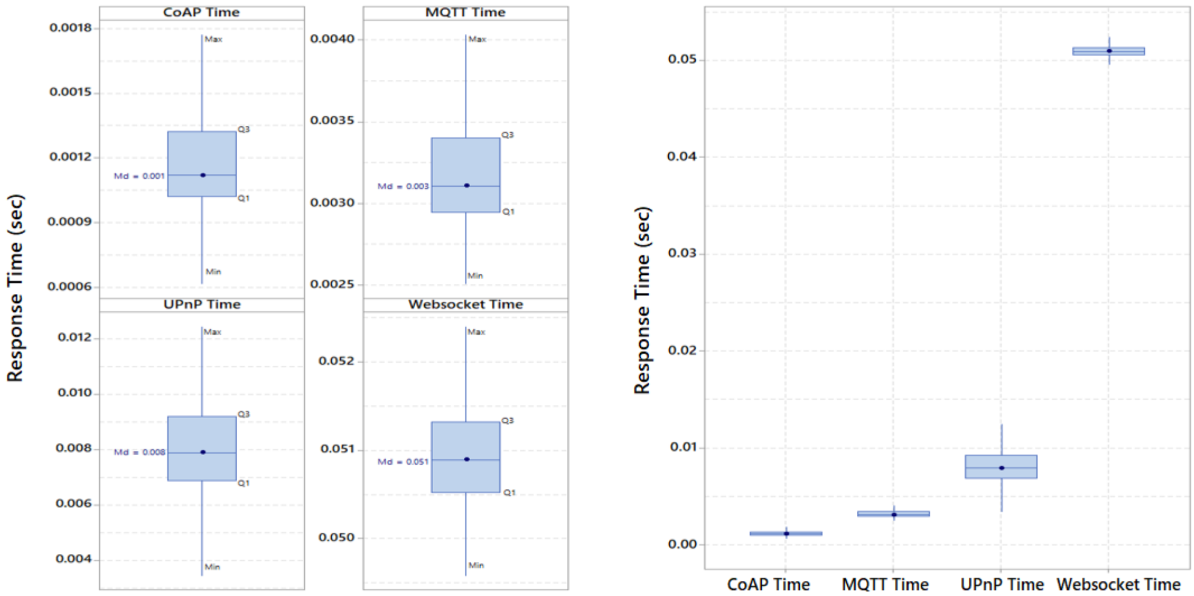
5.2.4.1 Is There Any Evinced Difference in Response Time on the Protocols?

A Friedman test was run to determine if there were differences in response time for the protocols CoAP, MQTT, UPnP, and WebSocket, using PlaySEM SER 2, during a 400-round test. Pairwise comparisons were performed with a Bonferroni correction for multiple comparisons. Response time was significantly different for all the protocols assessed, $\chi^2(3) = 1{,}167.312$, $p < .0005$. Post hoc analysis revealed statistically significant differences in response time from CoAP (*Md* = 0.001) to MQTT (*Md* = 0.003) ($p < .0005$), CoAP (*Md* = 0.001) to UPnP (*Md* = 0.008) ($p < .0005$), CoAP (*Md* = 0.001) to WebSocket (*Md* = 0.051) ($p < .0005$), MQTT (*Md* = 0.003) to UPnP (*Md* = 0.008) ($p < .0005$), MQTT (*Md* = 0.003) to WebSocket (*Md* = 0.051) ($p < .0005$), and UPnP (*Md* = 0.008) to WebSocket (*Md* = 0.051) ($p < .0005$). Therefore, the results reject the null hypothesis and accept the alternative hypothesis.

Figure 20 shows a box-and-whisker plot, usually designed to present non-normal distribution, with the central tendency for each protocol represented by medians. Table 4 reports the values.

Despite having outliers, the data points are not spread out over a broader range of values, which indicates that response time for those protocols follows a pattern.

Figure 20 – Box-and-whisker plot for each protocol assessed with PlaySEM SER 2 with quartiles.



Source: Created by the author.

Table 4 – Response time central tendency for each protocol (*n*=400).

	Response Time (sec)						
	Minimum	Q1	*Md*	Q3	Maximum	Range	IQR
CoAP	0.001	0.001	0.001	0.001	0.017	0.016	0.000
MQTT	0.002	0.003	0.003	0.003	0.033	0.031	0.000
UPnP	0.002	0.007	0.008	0.009	0.036	0.034	0.002
Websocket	0.041	0.051	0.051	0.051	0.074	0.033	0.001

Source: Created by the author.

5.2.4.2 Protocol Recommendation

Unlike timeline applications, in which some sensory effects can start before a particular scene, it is hard to know deterministically when an event will occur during the execution of an event-based one, such as a game, a VR/AR software, or any other interactive application. Thus, synchronism must be tight, and response to an interaction should be given as soon as possible in order to avoid delays, which might negatively impact users' expectation.

The results revealed that the selection of the protocol will have an influence on performance. Protocols commonly used in CPS such as CoAP and MQTT tend to deliver a response in a quite short time being the first one even faster. Compared to CoAP, UPnP was nearly 8 times slower, but it still remains as an option working around 8 ms. WebSocket hit 51ms on the median. If, on the one hand, it makes the process of integration easier for web applications, on the other hand, it could be infeasible for them to work efficiently in these scenarios. More research is needed with regard to the impact of delay in event-based mulsemedia systems for different types of sensory effects. For instance, this time could be reasonable for lighting but not for vibration. The jury is still out on the issue of to what degree this kind of delay plus devices' delivery time will impact the QoE of users.

In the event that developers devise their own communication protocol, they might enhance even more the response time interval in networked event-based mulsemedia systems. However, the interest in this study is in scenarios where interoperability and reusability prevail, and developers could take advantage of PlaySEM SER 2 to integrate their multimedia applications into mulsemedia systems through industrial standards. Therefore, the conclusion is that developers are recommended to use CoAP for improving transmission response time in similar cases when developing non-web applications. Although WebSocket provides support for mulsemedia web applications, developers should be cautious when using it in networked event-based mulsemedia systems.

5.2.5 Findings

This study made a comparison between different protocols (CoAP, MQTT, UPnP, and WebSocket) that convey information between mulsemedia applications to PlaySEM SER 2.

Four mock applications were created for each one. The interaction between the two sides is given by two steps: (i) a loop sequence to send SEM from the mulsemedia application to the framework; and (2) a real-time communication to start rendering sensory effects after an occurrence of an event in the mulsemedia former. This communication strategy, presented in Chapter 4 (Section 4.4.2), was an improvement in the face of works that transmit SEM whenever an event happens. Once SEM is processed once and then invoked through a cue over the network, it does save resources by reducing the overhead of repeated information transmitted.

By adding lightweight communication protocols, such as CoAP and MQTT, this study concludes that response time is boosted in relation to UPnP and WebSocket and are recommended for networked event-based mulsemedia systems. There were statistically significant differences in response time between those protocols. Furthermore, these time references can be used for the design of mulsemedia systems with temporal restrictions.

Finally, the limitations of this study are the specific test-bed—time may vary in an uncontrolled mulsemedia environment—and that it does not consider devices' activation time, which may differ from brand to brand, electro-mechanical features, and intrinsic characteristics of each sensory effect.

5.3 On the Effects of Software Architecture on Perceived QoE by Users

In mulsemedia systems, QoE is related to the perception of end-users, as described in Chapter 2 (Section 2.1.3). Thus, subjective studies are required to assess the level of enjoyment/annoyance experienced by users. This study compares user QoE whilst consuming mulsemedia from SEMP, which relies on a coupled-based architecture (monolithic component) (WALTL et al., 2013), and PlaySEM SER 2, a decoupled-based architecture presented in Chapter 4. Both PlaySEM SER 2 and SEMP have proven to enhance QoE but in different setups (WALTL; TIMMERER; HELLWAGNER, 2010a; JALAL et al., 2018b; COVACI et al., 2019b). The hypothesis is that if there is no difference between the reported QoE for SEMP and PlaySEM SER 2 using the same devices described in (WALTL; TIMMERER; HELLWAGNER, 2010a), then, the interconnections behind PlaySEM SER 2's architecture are not perceived by users. Therefore, it would be feasible to bind different mulsemedia technologies in a customizable way without decreasing levels of user QoE.

5.3.1 Participants

Forty participants (20 males and 20 females) aged from 16 to 55 participated in the experiment. Participants self-reported as being computer literate. No participant reported epilepsy (a disorder in which nerve cell activity in the brain is disturbed, causing seizures), which may be triggered by the exposure to certain patterns or backgrounds on a television screen. As such, none were excluded from taking part in the study.

5.3.2 Materials

5.3.2.1 Devices

The devices used in the experiment consisted of an LG 21.5 inch screen monitor, Philips amBX lighting devices, amBX fans, an amBX rumble device, Sony MDR-ZX330BT headphones, a mini-PC, a laptop, and a router, as depicted in Figure 21. The laptop was a quad-core Intel Core i5-5200U, 8 GB RAM, 512 GB SSD, running Windows 10 64 bits. The mini-PC was an Intel Atom CPU 330 processor, 2GB RAM, 80 GB hard drive, running Windows 7 32 bits. Both the laptop and the mini-PC had HDMI output to be connected to the screen monitor and were connected via cable to the router, a D-link model WBR-1310, supporting IEEE 802.11b/g wireless clients and 10/100 Ethernet.

5.3.2.2 Videos

Each subject viewed five different 1-2 minute long videos: (i) *Babylon*, an action movie trailer that features mainly gunshots and explosions; (ii) *Earth*, a video clip that shows animals

Figure 21 – On the left side, the setup of the QoE experiment. On the right, the devices turned on.

Source: Created by the author.

and nature as though the viewer were flying over them; (iii) *Formula 1*, a footage of Formula 1 racing exhibition; (iv) *Rambo*, another action movie trailer with gunshots and explosions; and (v) *Wo ist Klaus?*, a commercial advertisement. Figure 22 contains snapshots of the five videos.

Figure 22 – A glance of the five videos used in the QoE experiment: (i) Babylon, (ii) Earth, (iii) Formula 1, (iv) Rambo, and (v) Wo ist Klaus?

(i) Babylon (ii) Earth (iii) Formula 1

(iv) Rambo (v) Wo ist Klaus?

Source: Created by the author.

The videos for this experiment were from a public mulsemedia dataset provided by Waltl et al. (2012). Timmerer et al. (2012) found out that the videos *Babylon*, *Earth*, *Formula 1*, *Rambo*, and *Wo ist Klaus?* had the highest MOS in their experiments, which were performed over this dataset. Table 5 shows the parameters of the videos.

Table 5 – Videos used in the QoE experiment, their attributes, and the quantity of sensory effects for light, wind, and vibration.

Name	Resolution (WxH@FPS)	Bit-rate (kbit/s)	Length (sec)	Light / Wind / Vibration
Babylon	1280x544@24	6975	118.42	Auto/20/13
Earth	1280x720@25	7070	66	Auto/24/1
Formula 1	1280x720@25	5527	116.2	Auto/41/4
Rambo	1280x544@24	6486	58.1	Auto/3/7
Wo ist Klaus?	1024x576@30	4534	59.16	Auto/12/4

Source: (WALTL et al., 2012).

5.3.2.3 Software

The videos were annotated with MPEG-V. The laptop was used to play them using SEMP (version 0.12.1). PlaySEM SE Video Player (version 1.1.0) played the videos on the mini-PC whilst PlaySEM SER 2 (version 2.0.0) operated on the laptop.

5.3.3 Experimental Design

The mulsemedia system was the independent variable; specifically it had two values: PlaySEM (SER 2 and SE Video Player integrated through UPnP) and SEMP. A between-subjects (unrelated) study design was adopted where different people test each condition (GREENE; D'OLIVEIRA, 2005). The dependent variables are given by various aspects of the user experience, as encompassed by a QoE questionnaire, the details of which are provided in the following.

5.3.4 QoE Questionnaire

The QoE questionnaire comprised twenty questions targeting the user mulsemedia experience. The response to each question was expressed on a 5-point Likert scale, as detailed below.

Q01. I like the visual quality of the video.
Q02. I like the audio quality.
Q03. The audio is synchronized with the video.
Q04. I enjoy the content of the video.
Q05. The wind effects enhance the sense of reality.
Q06. The wind effects are synchronized with the video.
Q07. The wind effects are distracting.
Q08. The wind effects are annoying.

Q09. I enjoy watching the video with wind effects.
Q10. The vibration effects enhance the sense of reality.
Q11. The vibration effects are synchronized with the video.
Q12. The vibration effects are distracting.
Q13. The vibration effects are annoying.
Q14. I enjoy watching the video with vibration effects.
Q15. The external lighting effects enhance the sense of reality.
Q16. The external lighting effects are synchronized with the video.
Q17. The external lighting effects are distracting.
Q18. The external lighting effects are annoying.
Q19. I enjoy watching the video with external lighting effects.
{Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree}.
Q20. Please rate the overall experience watching the videos enhanced with wind, vibration, and external lighting effects.
{Very Bad, Bad, Neutral, Good, Excellent}.

5.3.5 Procedure

The experiment followed the practices described in Section 2.1.3.1, which presents mulsemedia experiences and evaluation methods. After being welcomed to the experimental room, participants were firstly briefed on the purpose of the experiment and asked whether they had any questions. Assuming that they were happy (accepted the disclaimer) to go ahead with the experiments (and any questions they might have had, had been satisfactorily answered),

Table 6 – Allocation of participants to the videos in the QoE assessment.

Id	Video 1	Video 2	Video 3	Video 4	Video 5
1	Formula 1	Babylon	Earth	Rambo	Wo ist Klaus?
2	Babylon	Earth	Rambo	Wo ist Klaus?	Formula 1
3	Earth	Rambo	Wo ist Klaus?	Formula 1	Babylon
4	Rambo	Wo ist Klaus?	Formula 1	Babylon	Earth
5	Wo ist Klaus?	Formula 1	Babylon	Earth	Rambo
6	Wo ist Klaus?	Rambo	Earth	Babylon	Formula 1
7	Rambo	Earth	Babylon	Formula 1	Wo ist Klaus?
8	Earth	Babylon	Formula 1	Wo ist Klaus?	Rambo
9	Babylon	Formula 1	Wo ist Klaus?	Rambo	Earth
10	Formula 1	Wo ist Klaus?	Rambo	Earth	Babylon

Source: Created by the author.

participants were asked to sit in front of the monitor. Then, they answered a pre-questionnaire on demographic information and experienced a 20-seconds trial in which they were asked whether they could perceive the external lighting, vibration, and wind coming from the devices. The introduction of the subjective quality assessment, the consent form, and the questionnaire on demographic information are described in APPENDIX C.

After that, each video was then played out to participants, with its corresponding SEM. Participants wore headsets to avoid being distracted by the noise of the devices working. Moreover, the room lights were switched off so that they could perceive the external lighting effects. So as to counteract order effects, the presentation order of the clips was varied. Table 6 describes the order for the first ten participants. This allocation was then cyclically repeated for the remainder of the user sample for each system. Participants were, however, unaware of what particular system they were experiencing. After watching each clip, participants completed the QoE questionnaire detailed in Section 5.3.4. Finally, at the end of the experiment, qualitative opinions on the overall experience were also collected from participants. On average, it took 25-30 minutes per participant to complete the experiment.

5.3.6 Results

All responses from the QoE questionnaire in section 5.3.4 where the 5 Likert scale items were mapped to the numeric values of 1..5 for analysis purposes. Data were analyzed with SPSS for Windows version (release 25.0). An independent-samples t-test was used to know if a difference exists between the means of two independent groups (WINTER; DODOU, 2010), and therefore, it tests the following hypothesis:

$$\mathrm{H_0}: \mu_{\mathrm{playsem}} = \mu_{\mathrm{semp}}$$
$$\mathrm{H_A}: \mu_{\mathrm{playsem}} \neq \mu_{\mathrm{semp}}$$

In words, the null hypothesis ($\mathrm{H_0}$) states that PlaySEM and SEMP MOS are equal in the population, whereas the alternative hypothesis ($\mathrm{H_A}$) states that PlaySEM and SEMP MOS are not equal in the population.

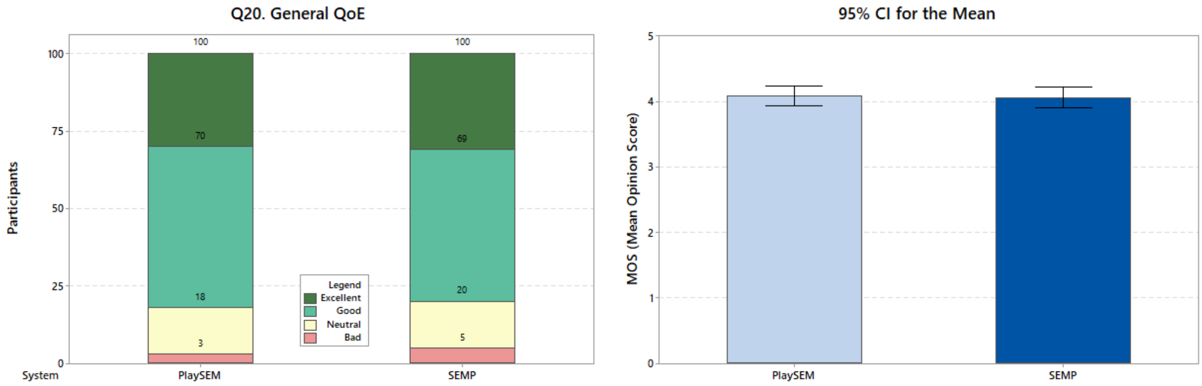
Additionally, a 2 (PlaySEM/SEMP) x 5 (Babylon/Earth/Formula 1/Rambo/Wo ist Klaus?) ANalysis Of VAriance (ANOVA), suitable to determine whether there is an interaction effect between two independent variables (NORMAN, 2010), was applied to analyze participants responses.

For both tests, a significance level of $p < .05$ was adopted for this study. All figures in this section use individual standard deviations to calculate the intervals unless otherwise stated. Though the results are reported throughout this section, Table 7 presents the MOS by system for each dependent variable with their respective standard deviation and standard error of the mean, and Table 8 provides details for the t-test for equality of means. Furthermore, Table 9 presents the MOS by system and video with their respective standard deviation and Table 10 reports tests of between-subjects effects. All these tables are introduced in APPENDIX C (Section C.4).

5.3.6.1 How Did the Groups Perceive the Experience?

Figure 23 shows a stacked bar chart and the MOS with a confidence interval of 95% for *Q20*—overall experience watching the videos enhanced with wind, vibration, and external lighting effects. A number of 80% of participants or more perceived the overall experience as positive (good or excellent). An independent-samples t-test was run to determine if there were differences in it between the groups using PlaySEM (4.09 ± 0.753) and SEMP (4.06 ± 0.814). There was no statistically significant differences for *Q20*, $t(38) = .270$, $p = .787$. Therefore, the test fails to reject the null hypothesis for the perceived overall experience.

Figure 23 – Stacked bar chart of evaluations and comparison of MOS for Q20—overall experience watching the videos with sensory effects.

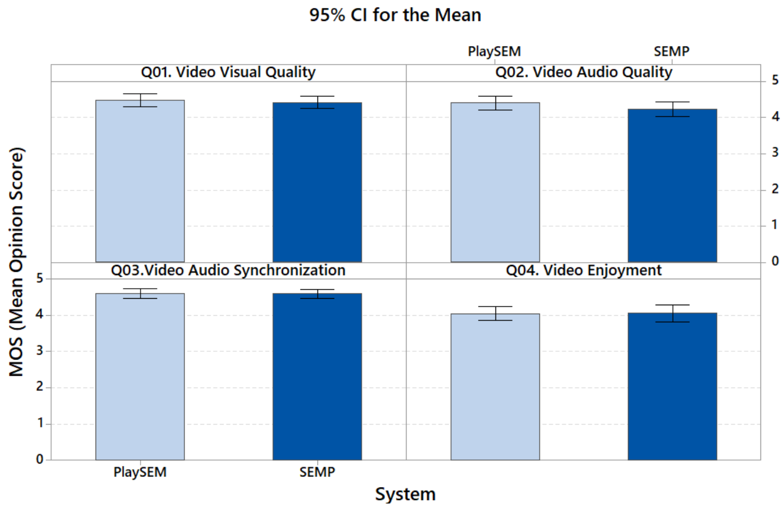


Source: Created by the author.

In Section 5.2, temporal aspects of PlaySEM SER 2 were discussed. Although the section is concerned with event-based applications, in timeline ones, the commands from mulsemedia applications to PlaySEM SER 2 go through the network. This process introduces a similar time of transmission when synchronizing the main media with sensory effects taking into account those protocols, creating inter-media skew. Indeed, synchronization does not have to be 100% precisely accurate, but any imperfection should not be noticeable by humans (MONTAGUD et al., 2018). To find out this, this study included questions about synchronization individually for each main group of aspects (audiovisual, wind effects, vibration effects, and external lighting effects), which are reported in the sequence with other dependent variables.

As for audiovisual experience and content (*Q01..Q04*), Figure 24 presents slight differences for Q02 between the groups. Independent-samples t-tests were run to determine if there were differences in perceived visual quality, audio quality, audio synchronization, and enjoyment. There were no statistically significant differences for *Q01*, $t(38) = .566$, $p = .572$, *Q02*, $t(38) = 1.239$, $p = .217$, *Q03*, $t(38) = .112$, $p = .911$, and *Q04*, $t(38) = -.065$, $p = .948$. Based on these results, the tests fail to reject the null hypothesis for aspects related to the audiovisual experience.

Figure 24 – Comparison of MOS for Q01..Q04—related to audiovisual experience and content.

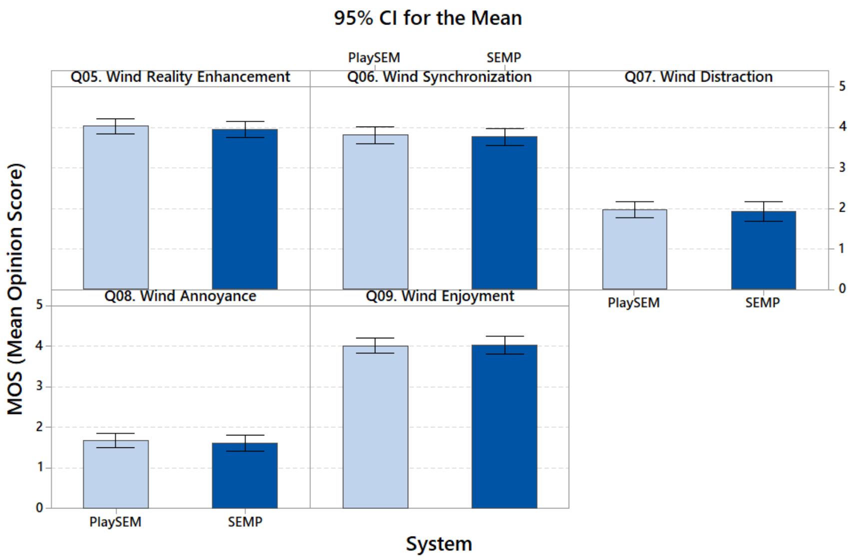


Source: Created by the author.

Figure 31 (APPENDIX C, Section C.5) shows that 88% of participants in both groups agree or strongly agree on the statement "I like the visual quality of the video." With respect to audio quality, 83% of participants perceived it positively (agree or strongly agree) on SEMP, whereas 87% was the number for PlaySEM. The audiovisual content was synchronized according to the perception of 96% of participants (agree or strongly agree). As for the content, 74% of participants agreed or strongly agreed that they enjoyed it on SEMP, whereas 73% on PlaySEM.

The result for the group of questions related to wind (*Q05..Q09*) is showed in Figure 25. There is little variation between the data. To confirm this, independent-samples t-tests

Figure 25 – Comparison of MOS for Q05..Q09—wind effects.



Source: Created by the author.

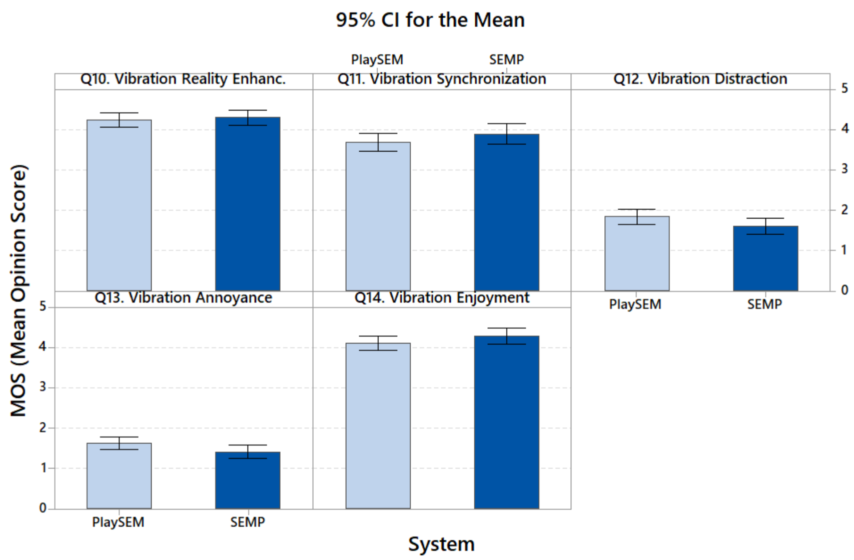
were ran to determine if there were differences in wind perceived reality enhancement, wind synchronization, distraction, annoyance, and enjoyment caused by wind effects. There were no statistically significant differences for *Q05*, $t(38) = .561$, $p = .576$, *Q06*, $t(38) = .268$, $p = .789$, *Q07*, $t(38) = .261$, $p = .795$, *Q08*, $t(38) = .446$, $p = .656$, and *Q09*, $t(38) = -.14$, $p = .889$. Thus, the tests fail to reject the null hypothesis considering wind effects factors.

Figure 32 (APPENDIX C, Section C.5) brings a positive outlook for wind effects for both systems, with more than 70% agreeing or strongly agreeing that wind enhances reality. With regard to synchronization of wind with the scenes in the videos, 73% of participants pointed out that they perceived some degree (agree or strongly agree) of synchronization for PlaySEM and 65% for SEMP. Few participants reported that they agree or strongly agree that wind effect is distracting when watching the videos: 10% for PlaySEM and 20% for SEMP. The same low trend is observed for the annoyance caused by the wind effect: 6% for PlaySEM and 10% for SEMP (agree or strongly agree). As for enjoyment of watching the videos with wind effects, 73% of participants agreed that they enjoyed it on PlaySEM, whereas 69% on SEMP.

Figure 26 presents the means for vibration effects (*Q10..Q14*). Independent-samples t-tests were ran to determine if there were differences in vibration perceived reality enhancement, vibration synchronization, distraction, annoyance, and enjoyment caused by vibration effects. There were no statistically significant differences for *Q10*, $t(38) = -.447$, $p = .655$, *Q11*, $t(38) = -1.277$, $p = .203$, *Q12*, $t(38) = 1.773$, $p = .078$, *Q13*, $t(38) = 1.91$, $p = .058$, and *Q14*, $t(38) = -1.329$, $p = .185$. Therefore, the tests also fail to reject the null hypothesis considering vibration.

Vibration was the most welcomed effect. Figure 33 (APPENDIX C, Section C.5) shows that more than 80% agree or strongly agree that the vibration effect enhances the sense of

Figure 26 – Comparison of MOS for Q10..Q14—vibration effects.

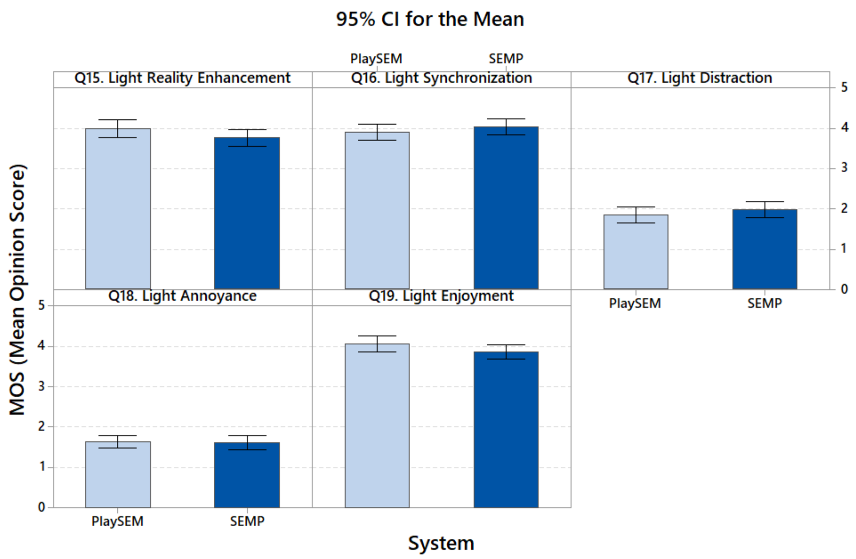


Source: Created by the author.

reality. As for synchronization, 63% and 73% agreed or strongly agreed that vibration effects were in sync with the videos respectively for PlaySEM and SEMP. Considering participants that took it as neutral, the numbers change to 79% for PlaySEM and 80% for SEMP. Very few participants agreed or strongly agreed that wind effects were distracting (16% for PlaySEM and 15% for SEMP) or annoying (11% for PlaySEM and 12% for SEMP). Finally, 76% and 80% of participants agreed or strongly agreed that they enjoyed watching the videos with vibration effects.

As far as external lighting effects, Figure 27 depicts the results for *Q15..Q19*. Independent-samples t-tests were ran to determine if there were differences in the perceived external lighting reality enhancement, synchronization, distraction, annoyance, and enjoyment caused by this type of effect. There were no statistically significant differences for *Q15*, $t(38) = 1.553$, $p = .122$, *Q16*, $t(38) = -.919$, $p = .359$, *Q17*, $t(38) = -.922$, $p = .358$, *Q18*, $t(38) = .173$, $p = .863$, and *Q19*, $t(38) = 1.6$, $p = .111$. The tests fail to reject the null hypothesis with regard to external lighting effects.

Figure 27 – Comparison of MOS for Q15..Q19—external lighting effects.



Source: Created by the author.

Figure 34 (APPENDIX C, Section C.5) presents that 73% (PlaySEM) and 55% (SEMP) agreed or strongly agreed that external lighting effects enhance reality whilst watching the videos. What is remarkable is the number of 36% of neutral evaluations for SEMP. For both systems, 72% or more participants agreed or strongly agreed that the external lighting effects were on time. Less than 10% of participants agreed or strongly agreed that light was distracting or annoying for both systems. The number of participants who agreed or strongly agreed that they enjoyed watching the videos with external lighting effects reached 69% for PlaySEM and 59% for SEMP. Again, a high number of neutral opinions was reported. This turn, for both systems: 27% for PlaySEM and 39% for SEMP. These numbers show that participants liked the least external lighting effects in comparison with wind and vibration on both systems.

5.3.6.2 Interaction Between System and Video

The interaction between factors of the independent variables aims at answering if the effect of one of the variables differs depending on the level of the other variable. In this case, the analysis consists in assessing if there is a significant change in the dependent variable when changing the videos for each system.

A graph of MOS for the interaction between system and video is presented in Figure 28, encompassing all the questions. A two-way ANOVA was conducted that examined the effect of system and video on user-perceived QoE. There was no statistically significant interaction between them for *Q01*, $F(4,190) = 1.035$, $p = .39$, *Q02*, $F(4,190) = 1.802$, $p = .13$, *Q03*, $F(4,190) = .966$, $p = .427$, *Q04*, $F(4,190) = .652$, $p = .626$, *Q05*, $F(4,190) = .583$, $p = .675$, *Q06*, $F(4,190) = .737$, $p = .568$, *Q07*, $F(4,190) = .287$, $p = .886$, *Q08*, $F(4,190) = .57$, $p = .685$, *Q09*, $F(4,190) = .318$, $p = .866$, *Q10*, $F(4,190) = .551$, $p = .699$, *Q11*, $F(4,190) = .701$, $p = .592$, *Q12*, $F(4,190) = .284$, $p = .888$, *Q13*, $F(4,190) = .267$, $p = .899$, *Q14*, $F(4,190) = .489$, $p = .743$, *Q15*, $F(4,190) = .342$, $p = .849$, *Q16*, $F(4,190) = .761$, $p = .552$, *Q17*, $F(4,190) = 1.594$, $p = .178$, *Q18*, $F(4,190) = 1.005$, $p = .406$, *Q19*, $F(4,190) = .289$, $p = .885$, and *Q20*, $F(4,190) = .414$, $p = .799$.

5.3.6.3 Qualitative Answers

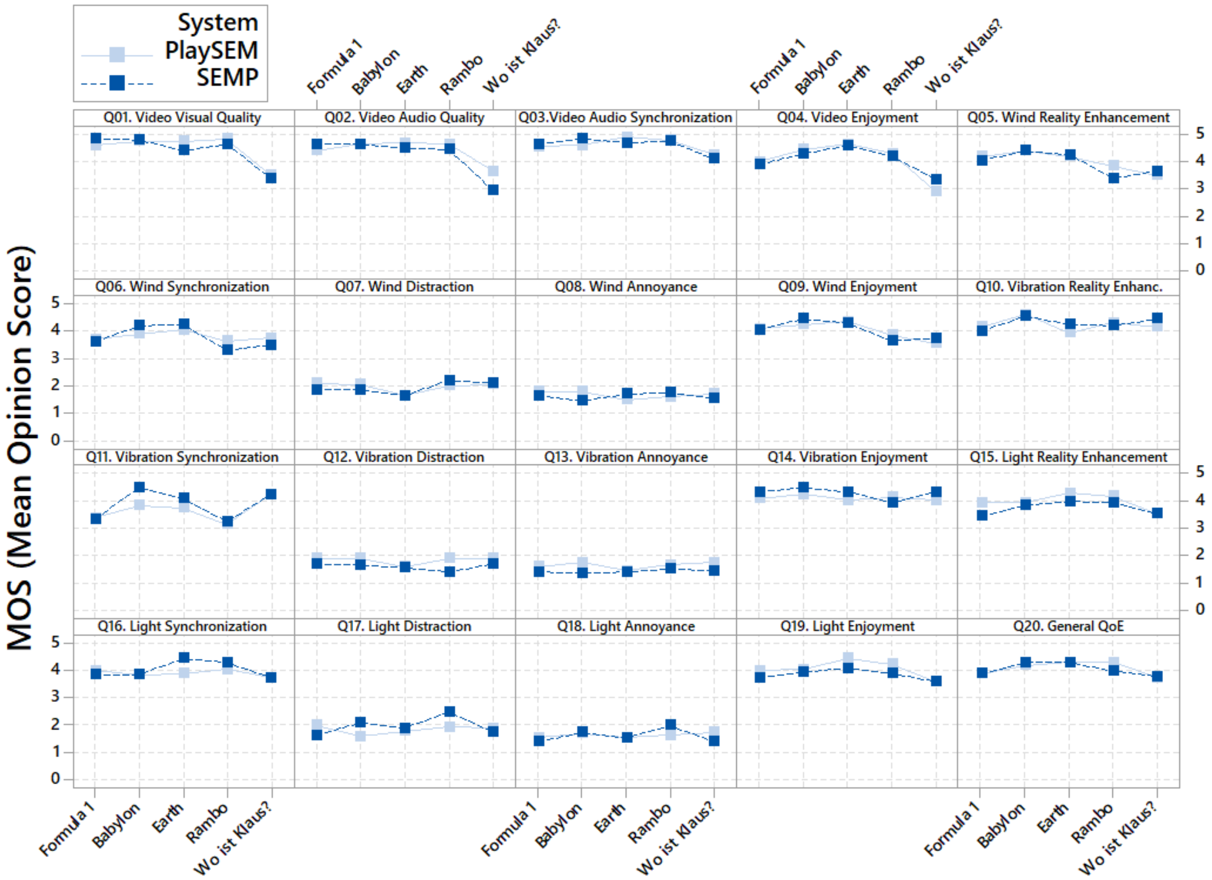
Upon the experiment's completion, the participants were asked to describe their experience. Most participants did not report previous multisensory experience, whereas those who did described that the current experience recalled memories of the previous. Few of them stated that they felt anxious because sometimes they were waiting for sensory effects that did not materialize regardless of the system. This can be related to video annotation, which is out of the scope of this investigation.

The users' comments were inextricably connected to their experience as a whole. Some participants pointed out they disliked the wind blowing on their faces, whereas others liked it. Many participants longed for some sort of warm or cold wind. A few users suggested sensory effects adaptation to their preferences. Vibration was suggested by some of the participants to be delivered on a chair or on a wearable device instead of a rumble bar. External lighting was the sensory effect the participants least argued as a must-have. Finally, nothing related to the systems themselves was captured during the final interview.

5.3.7 Findings

With the aim to compare user QoE whilst consuming mulsemedia from a monolithic system (SEMP) and a decoupled one (PlaySEM SER 2), 40 participants (20 male and 20 female) divided into 2 groups were invited to take part in this subjective experiment. Each group watched

Figure 28 – Comparison of MOS for the interaction between system and video for Q01..Q20.



Source: Created by the author.

5 different videos in a randomized order on PlaySEM SER 2 and SEMP without being aware of it. An amBX Premium Kit (wind fans, vibration bar, and external lighting) was employed to render the sensory effects for both systems.

This study concludes that no statistically significant difference in user-perceived QoE was found between the two systems under different perspectives such as overall experience watching the videos with sensory effects, audiovisual experience and content, wind effects, vibration effects, external lighting effects. The levels of perceived quality, reality, synchronization, distraction, annoyance, and enjoyment were not remarkably affected when the system was changed.

Moreover, an analysis of the interaction between system and video did not evince that there is a statistically significant change in user-perceived QoE when changing the videos for each system. From these conclusions, this study has shown that PlaySEM SER 2's architecture, which allows the possibility of combining different technologies in a customizable fashion, can be leveraged in mulsemedia environments without decreasing levels of user QoE in comparison with a monolithic system.

6 Conclusion

Low-cost and innovative technologies and devices that take into consideration humans' five Aristotelian senses have allowed the design of systems hitherto unprecedented, named mulsemedia. Researchers have exploited it to pave new ways of human-computer interaction in areas such as entertainment, healthcare, education, culture, and marketing, under varied perspectives involving the combination of different senses, configurations, and ever-evolving mulsemedia devices. This has come along with fast-paced new user interfaces, such as gestural and multi-touch interfaces, voice processing, and brain-computer interfaces, to name a few, and multi-purpose applications. These systems have undeniably improved users' QoE over traditional forms of multimedia, which is usually restricted to audiovisual content. However, it is unfortunate that mulsemedia applications and devices designed for the same purpose are hardly ever interchangeable, which may make the design of these experiences demanding, time-consuming, and costly.

This work reflects upon the integration between them in order to design reusable mulsemedia systems. In particular, this thesis presents a conceptual architecture and a framework that take into account recurrent problems in mulsemedia delivery systems without compromising reusability. The conceptual architecture is proposed upon challenges and requirements for mulsemedia delivery systems that stem from gaps and shortcomings identified in related work. It explores abstract techniques to deal with variability of scenarios, which include recommendations to cope with variation of end-user applications and sensory effect devices through the support and reuse of protocols of communication and connectivity, and sensory effects metadata standardization. The framework implements the precepts of the conceptual architecture whilst dealing with low-level practical aspects. In order to evaluate them, case studies, performance, and subject experiments were undertaken.

In this concluding chapter, the research questions in the introduction are revisited and answered.

How can researchers and developers design/leverage reusable mulsemedia systems for different contexts considering varied end-user applications and heterogeneous devices?

As shown in Chapter 2, many hardware and software solutions have been created for mulsemedia, with very few approaches to dealing with variability of scenarios of usage by reusing software. There have been identified incipient proposals of conceptual architectures and frameworks inextricably restricted to MPEG-V, IoT approaches applied to mulsemedia, and SDKs and APIs to interface with specific hardware. A plethora of issues have hindered the development of models and tools that takes variability of scenarios of usage into account as reviewed in related work. Notably, high coupling of software, lack of support for varied protocols to interact with applications and devices, absence of solutions able to work with timeline- and event-based multimedia applications, unavailability of mechanisms to support more than one SEM standard, scarcity of means to compensate delays introduced for some hardware or connectivity and communication protocols, hardware heterogeneity tackled isolated by some SDKs or APIs just in haptic, lack of schemes to adapt solutions to different profiles of usage without changing their code, and code reliance strictly on existent technologies, protocols, and standards without caring for future extensibility.

From this outlook, building and integrating such complex systems in an adaptable fashion impose many challenges that involve multifunctionality (to provide operations for different applications to support multisensory effects) and reusability (to accommodate these changes constantly), reactivity and timeliness to ensure reliable mechanisms for quick response time under temporal constraints, and manageability and configurability to deal with software and hardware heterogeneity with minimal or no coding. Thus, in Chapter 3, this work proposed a conceptual architecture underpinned on requirements (that provide input of information for reasoning about mulsemedia delivery system) to cope with these challenges and thus addresses variability of scenarios of usage regardless of technology. Researchers and developers can leverage this conceptual architecture to design new mulsemedia systems that deal with variation of end-user applications and sensory effect devices through the support and reuse of different protocols of communication and connectivity, and SEM standards. It grounded on the separation of concerns by dividing a mulsemedia system into five main layers: (i) *Mulsemedia Applications*, which isolates multimedia issues; (ii) *Communication Broker*, which provides the reuse of mulsemedia services provided by SER's through different communication protocols; (iii) *Sensory Effects Processing*, which copes with metadata processing regardless of the SEM standard; (iv) *Connectivity Layer*, which allows the reuse of different connectivity protocols for sensory effect devices independently of implementation of commands for devices; and (v) *Sensory Effects Rendering*, to represent devices that deliver sensory effects to end-users. This conceptual architecture does not intend to provide technical details nor naming conventions, but it does propose a representation where mulsemedia systems could be built once and extended continually to support different scenarios of usage in a tailored fashion taking into consideration its flexibility to expand and reuse its components. Furthermore, it is depicted to be understandable and graspable, therefore formal validation is not introduced at current to avoid compromising these principles rather particular to non-critical systems.

Upon this conceptual architecture, as shown in Chapter 4, a framework has been built with a lower level of architectural details so that researchers and developers can either undertake mulsemedia experiments or take it as a reference to follow the conceptual architecture. This framework supports multi-communication and multi-connectivity protocols, multi-standards, and allows the accommodation of new technology relying on its set of architectural and design patterns to be applied successfully. It provides services to mulsemedia applications through a transparent communication broker and is able to work with timeline- and event-based applications.

Moreover, it presents a set of configurable parameters to customize the solution according to the needs imposed by mulsemedia applications, offers a mechanism to calibrate delay for each component it works with, and provides a debug and flexible simulation scheme. In this thesis, it has been presented the implementation of different case studies to demonstrate the framework's capability to adapt itself to different scenarios of usage through configuration, as described in Chapter 5 (Section 5.1). These studies include different profiles of usage successfully undertaken to adjust the framework for (i) a mulsemedia environment comprised of video clips enriched with external light, smell, vibration, and wind delivered a system composed of a media player compatible with MPEG-V, DIY devices, smartphones, and a commercial scent device; (ii) a scenario of usage whereby a proprietary device is handled by the framework to deliver different intensities of smell; and (iii) a 360° VR mulsemedia system composed of an HMD display with a portable scent diffuser attached to it and a wind device to engage users in a 360° environment augmented with smell and airflow.

How can networked event-based mulsemedia systems have their performance improved to avoid undesired delays, which would eventually spoil user QoE?

Depending on the interaction, the time it takes for applications to produce an expected outcome may have a significant impact on the way that users perceive mulsemedia experiences. For instance, when examining timestamps in haptics, tolerable delays are most often below one hundred milliseconds. The framework, derived from the conceptual architecture, relies on network resources, which may eventually have an influence on the system's performance. Thus, a study on the framework's efficiency to interact with event-based applications was undertaken and presented in Chapter 5 (Section 5.2). In particular, a comparison between different protocols (CoAP, MQTT, UPnP, and WebSocket) that convey information from four mock networked event-based mulsemedia applications to the framework was performed. The results showed statistically significant differences in response time between those protocols with Md=1 ms for CoAP, Md=3 ms for MQTT, Md=7 ms for UPnP, and Md=51 ms for WebSocket under the same network conditions.

Furthermore, although times from one point to another were measured, the framework introduces an optimized way of interacting with event-based applications in which SEM is transmitted before real-time operations. Thereby, not only does it save network resources by sending metadata just once instead of whenever an event occurs, but also allows event-based applications to send short signals when an action is necessary from the framework by indicating just the identification of the event associated with the previously processed SEM. Therefore, the study makes two suggestions to improve networked event-based mulsemedia systems' performance. First, whenever possible, give preference to lightweight communication protocols, such as CoAP and MQTT, in relation to UPnP and WebSocket. Second, pre-process SEM before real-time communication to avoid unnecessary overhead transmitted over the network. It should be acknowledged that this study was carried out under a controlled environment and does not consider devices' activation time, which may differ from brand to brand, electro-mechanical features, and intrinsic characteristics of each sensory effect. Nevertheless, this study can be used as time references to design networked event-based mulsemedia systems with temporal restrictions.

Do users perceive mulsemedia experiences differently when mulsemedia systems are monolithic or have a decoupled approach?

Assessing users' experience is one of the most challenging tasks for mulsemedia researchers to perform due to a plethora of social, psychological, and technical variables involved in it. The way that mulsemedia features are addressed will have an impact on end-users. Thus, this question is complementary to that of how researchers and developers can design/leverage reusable mulsemedia systems for different contexts considering varied end-user applications and heterogeneous devices. In particular, it compares user-perceived QoE whilst consuming mulsemedia from a monolithic system and the framework derived from the conceptual architecture presented in this thesis. In order to find out the answer to this question, forty participants split into independent groups for each system watched five different videos enriched with sensory effects and evaluated them under different perspectives such as overall experience watching the videos with sensory effects, audiovisual experience and content, wind effects, vibration effects, external lighting effects. Since the monolithic system supported only one set of sensory effect devices to deliver wind, vibration, and external lighting, the framework was set up for the same equipment to avoid the effects of lurking variables, which may affect comparisons.

From this prospect, this study did not find a statistically significant difference in user-perceived QoE, as shown in Chapter 5 (Section 5.3), that is, the levels of perceived quality, reality, synchronization, distraction, annoyance, and enjoyment were not remarkably affected when the system changed. Furthermore, it was not found significant changes in these dependent variables whilst alternating the videos. Therefore, the answer to this question taking into account the limitations of this study is negative—users do not perceive mulsemedia experiences differently when mulsemedia systems are monolithic or have a decoupled approach. In other terms, it is feasible to integrate different technologies in a customizable fashion, therefore leveraging the design of networked mulsemedia systems relying on the conceptual architecture and the framework without decreasing levels of user QoE in comparison with a monolithic system. Thus, researchers and developers can exploit different combinations of applications and devices in mulsemedia system using this work free of concerns that they could affect users QoE adversely. A caveat, however, is that the decoupled approach underlined in this question is inextricably associated with this conceptual architecture and used a specific setup. Therefore, variations of it should be carefully analyzed.

6.1 Limitations

Although the findings of this study have provided evidence to answer the research questions, there is a need to be careful when interpreting the results. As for the conceptual architecture, it is based on constructs, that is, this work is limited to present a theoretical model containing various conceptual elements and to validate it through case studies where different pieces are fitted together. Thus, the assumption that each element can be measured to be improved is not taken into account. Deeper and focused research to examine formal requirement measures should be undertaken towards this direction with the caveat that the trade-off between formality and architecture's assimilation should not be compromised to avoid its adoption. Furthermore, the conceptual architecture is not concerned with user preferences, adaptation, safety, and security. Rather, it focuses on the way that mulsemedia delivery systems should be built. This reflects on the framework, which instantiates those concepts and does not support content/context adaptation according to user preferences. Alternatively, end-user applications could take on this role, acknowledging that this strategy would not promote reuse. As this is a relevant topic to user QoE, it is discussed with more detail in future work.

As for the study on networked event-based mulsemedia systems performance, first, the number of assessed protocols were limited to 4 to encompass current lightweight ones (CoAP and MQTT) and others used on a home network (UPnP) and the Internet (Websocket). To the extent that technology evolves, other protocols should be considered. Second, another limitation is that it was measured the time from end-user applications to the framework and not the whole chain until reaching the devices. The rationale behind it is that devices connected through a serial port are ordinary in mulsemedia systems and add negligible delays, then, mock devices were used. In order to measure the time it takes to carry information from the framework to remote devices connected through Wi-Fi and Bluetooth, for instance, a remote clock synchronization with a precision of milliseconds should be employed. This is not straightforward as it is still a challenge in computer science although there exist some algorithms that aim to coordinate otherwise independent clocks. Third, this study is limited to a specific test-bed—time may vary in an uncontrolled mulsemedia environment—and that it does not consider devices' activation time, which may differ from brand to brand, electro-mechanical features, and intrinsic characteristics of each sensory effect. Finally, with the advent of the new generation of low-latency networks, such as Wi-Fi 6 and 5G, the issue of latency when exchanging messages between end-user applications, the framework, and devices tends to be mitigated. Meanwhile, this work provides time references to be used for the design of mulsemedia systems with temporal restrictions.

With regard to the QoE experiment, other systems could not be compared because they are not freely available. Towards this issue, videos were varied not only in content but also in the order they were presented to the participants. Furthermore, the survey was split into different categories to examine any discrepancies and lurking variables that could potentially bias the results. Another acknowledged limitation was that the monolithic system did not support other devices different from the kit used in the experiment. Having an additional group of participants exposed to another set of devices for both systems could strengthen the results.

6.2 Future Work

This work caters for a conceptual architecture and a framework to aid researchers. Nevertheless, a plethora of other research and technical challenges leaves an open room for future research, which are described and explained as follows:

- **Calibration of devices before reproducing sensory effects**. At the moment, researchers and practitioners alike have not stabilized mulsemedia environments before delivering the sensory effects. This implies that end-users under different environmental conditions may experience the same content otherwise. This is rather clear in the context of temperature effects. If a user is in an acclimatized (18 °C) room consuming mulsemedia content and another is in a stuffy and sweltering one (32 °C), will they perceive the thermal experience provided by the mulsemedia content the same way? Research towards this direction has to be undertaken in order to find out user QoE under different environmental conditions, and in case of differences, a proposal to calibrate the devices with stronger or weaker power than that of the specified by authors in the SEM file. However, this would have to be carefully analyzed for different types of sensory effects to try to establish rules of calibration. In terms of the temperature device, if the mulsemedia system detects that the environment is warm, it would intensify the thermal device until it reaches the temperature specified in the SEM. How would it be with respect to other sensory effects? How could the environment be monitored and adjusted constantly? Does this calibration improve users' QoE?
- **Support for user monitoring during experiments.** A trendy research topic in multimedia is how to understand users' QoE from objective metrics. This involves capturing physiological data from users to obtain concealed data behind subjective experiments. Cameras, eye-tracking devices, heart-rate monitoring wrists, and electroencephalography headsets are examples of equipment to do this end. However, integrating them to users QoE experiments is not straightforward. If it provides an SDK or API, researchers and developers struggle to adapt them to their system; if data is captured separately by third-party software not integrated to the mulsemedia system, researchers have to synchronize them manually when the experience starts and finishes. Both of them take time, which could be spent on research instead. Therefore, a proposal that integrates sensors to capture users' physiological data automatically in mulsemedia systems would save not only researchers time, but also organize this data in a fashion that is ready to be analyzed. This could potentially boost objective metrics in mulsemedia experiments and, as a result, grow more interest in developing QoE models from objective measures.

- **User preferences adaptation**. During the subjective experiment carried out in this work, it was commonplace to hear from users about their preferences. Statements such as “I would prefer weaker wind” or “The vibration could be stronger” show that users have different expectations. The standard MPEG-V (part 2) takes this consideration into account and proposes to adjust the system accordingly. However, little has been investigated in terms of users’ preference in mulsemedia systems. Thus, another path to study is mechanisms to adapt systems to the users’ taste. Real-time feedback from emotion analysis incorporated in mulsemedia systems could be useful in this context. Data-driven approaches using artificial intelligence algorithms to analyze users’ behavior whilst consuming mulsemedia and to make insightful inferences to adapt mulsemedia systems to users’ expectations would increase the odds of delivering better experiences. Moreover, this could eventually provide data to create a mulsemedia behavioral dataset from previous experiences to be taken into consideration to design new systems or to produce new adjusted mulsemedia content based on input that considers users profiles as an input. Within this scenario, users’ safety and security should be addressed. Safety has to do with how the sensory effects are delivered (strength, intensity, direction, etc.) that could eventually affect users’ health and are linked to adapt the system to users’ preference. On the other hand, security is a concern related to data of users, which should not be exposed to arbitrarily.

- **Mulsemedia delivery systems scalability and synchronization**. Many current mulsemedia systems described in the literature deliver sensory effects using one or two devices per sense. To the extent the technology evolves, it is likely to have a plethora of vibration actuators spread along users’ body, and granular wind and scent coming from plenty of devices arranged in an immersive environment. Indeed, some of these types of new devices have already started emerging but limited to one sense using specific SDKs and APIs. When doing this using a network and a mulsemedia renderer to control synchronization and response time, it is unknown how current communication and connectivity protocols can efficiently support massive use. This raises questions such as: How scalable current mulsemedia delivery systems have to be to support more than 100 actuators at the same time, for instance? Would it require new specific protocols to deliver sensory effects on time? Would it demand new mechanisms to deal with the synchronization of these actuators?

The flourishing of mulsemedia experiences will ultimately rely on user acceptance, which in turn, permeates not only the levels of enjoyment that they feel but also the way that technology delivers mulsemedia. Therefore, this work introduced a conceptual architecture to develop reusable mulsemedia delivery systems and an implementation of it by means of a framework evaluated from the point of view of users QoE. Whilst recognizing their current limitations, it is hoped that mulsemedia researchers and developers can leverage them to carry out their experiments in heterogeneous settings, opening new tracks to investigate methods to create wealthier experiences that can lead to advancements in human-computer interaction.

Bibliography

ADELSTEIN, B. D.; LEE, T. G.; ELLIS, S. R. Head tracking latency in virtual environments: psychophysics and a model. In: SAGE PUBLICATIONS SAGE CA: LOS ANGELES, CA. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. [S.l.], 2003. v. 47, n. 20, p. 2083–2087. Cited 2 times on pages 19 and 25.

ADEMOYE, O. A.; GHINEA, G. Synchronization of olfaction-enhanced multimedia. *IEEE Transactions on Multimedia*, v. 11, n. 3, p. 561–565, April 2009. ISSN 1520-9210. Cited 5 times on pages 18, 25, 26, 27, and 33.

ADEMOYE, O. A.; GHINEA, G. Information recall task impact in olfaction-enhanced multimedia. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, ACM, v. 9, n. 3, p. 17, 2013. Cited 2 times on pages 18 and 33.

ADEMOYE, O. A.; MURRAY, N.; MUNTEAN, G.-M.; GHINEA, G. Audio masking effect on inter-component skews in olfaction-enhanced multimedia presentations. *ACM Trans. Multimedia Comput. Commun. Appl.*, ACM, New York, NY, USA, v. 12, n. 4, p. 51:1–51:14, ago. 2016. ISSN 1551-6857. Cited on page 18.

AKHTAR, Z.; SIDDIQUE, K.; RATTANI, A.; LUTFI, S. L.; FALK, T. H. Why is multimedia quality of experience assessment a challenging problem? *IEEE Access*, p. 1–1, 2019. ISSN 2169-3536. Cited on page 19.

AMORES, J.; MAES, P. Essence: Olfactory interfaces for unconscious influence of mood and cognitive performance. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017. (CHI '17), p. 28–34. ISBN 978-1-4503-4655-9. Cited on page 35.

AMORES, J.; MAES, P. Essence video showcase: Olfactory interfaces for unconscious influence. In: ACM. *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. [S.l.], 2017. p. 471–471. Cited on page 35.

AMORIM, M. N. de; SALEME, E. B.; NETO, F. R. de A.; SANTOS, C. A. S.; GHINEA, G. Crowdsourcing authoring of sensory effects on videos. *Multimedia Tools and Applications*, v. 78, n. 14, p. 19201–19227, Jul 2019. Cited 2 times on pages 18 and 23.

ANTONAKOGLOU, K.; XU, X.; STEINBACH, E.; MAHMOODI, T.; DOHLER, M. Toward haptic communications over the 5g tactile internet. *IEEE Communications Surveys Tutorials*, v. 20, n. 4, p. 3034–3059, Fourthquarter 2018. ISSN 1553-877X. Cited 2 times on pages 19 and 26.

BALZAROTTI, N.; BAUD-BOVY, G. Hpge: An haptic plugin for game engines. In: SPRINGER. *International Conference on Games and Learning Alliance*. [S.l.], 2018. p. 330–339. Cited 2 times on pages 17 and 39.

BANKS, A.; GUPTA, R. *MQTT Version 3.1.1, OASIS Standard*. 2014. <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>. Online; accessed 19 October 2017. Cited on page 62.

BARTOCCI, S.; BETTI, S.; MARCONE, G.; TABACCHIERA, M.; ZANUCCOLI, F.; CHIARI, A. A novel multimedia-multisensorial 4d platform. In: *2015 AEIT International Annual Conference (AEIT)*. [S.l.: s.n.], 2015. p. 1–6. Cited 2 times on pages 18 and 41.

BENKO, H.; HOLZ, C.; SINCLAIR, M.; OFEK, E. Normaltouch and texturetouch: High-fidelity 3d haptic shape rendering on handheld virtual reality controllers. In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2016. (UIST '16), p. 717–728. ISBN 978-1-4503-4189-9. Cited on page 32.

BROY, M. The 'grand challenge' in informatics: Engineering software-intensive systems. *Computer*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 39, n. 10, p. 72–80, out. 2006. ISSN 0018-9162. Cited 2 times on pages 20 and 45.

BRUNNSTRÖM, K.; BEKER, S. A.; MOOR, K. D.; DOOMS, A.; EGGER, S.; GARCIA, M.-N.; HOSSFELD, T.; JUMISKO-PYYKKÖ, S.; KEIMEL, C.; LARABI, M.-C.; LAWLOR, B.; CALLET, P. L.; MÖLLER, S.; PEREIRA, F.; PEREIRA, M.; PERKIS, A.; PIBERNIK, J.; PINHEIRO, A.; RAAKE, A.; REICHL, P.; REITER, U.; SCHATZ, R.; SCHELKENS, P.; SKORIN-KAPOV, L.; STROHMEIER, D.; TIMMERER, C.; VARELA, M.; WECHSUNG, I.; YOU, J.; ZGANK, A. *Qualinet White Paper on Definitions of Quality of Experience*. 2013. Fifth Qualinet meeting, Novi Sad, March 12, 2013. Available at: <https://hal.archives-ouvertes.fr/hal-00977812>. Cited on page 26.

BUJAS, Z.; SZABO, S.; KOVAČIĆ, M.; ROHAČEK, A. Adaptation effects on evoked electrical taste. *Perception & Psychophysics*, Springer, v. 15, n. 2, p. 210–214, 1974. Cited on page 36.

BUSCHMANN, F.; HENNEY, K.; SCHMIDT, D. *Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing*. [S.l.]: John Wiley & Sons, 2007. ISBN 0470059028, 9780470059029. Cited on page 59.

CANNA, A.; PRINSTER, A.; FRATELLO, M.; PUGLIA, L.; MAGLIULO, M.; CANTONE, E.; PIROZZI, M. A.; SALLE, F. D.; ESPOSITO, F. A low-cost open-architecture taste delivery system for gustatory fmri and bci experiments. *Journal of Neuroscience Methods*, v. 311, p. 1 – 12, 2019. ISSN 0165-0270. Cited 2 times on pages 17 and 40.

CARTER, T.; SEAH, S. A.; LONG, B.; DRINKWATER, B.; SUBRAMANIAN, S. Ultrahaptics: multi-point mid-air haptic feedback for touch surfaces. In: ACM. *Proc. of the 26th annual symposium on User Interface Software and Technology*. [S.l.], 2013. p. 505–514. Cited on page 32.

CHA, J.; EID, M.; SADDIK, A. E. Touchable 3d video system. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, ACM, v. 5, n. 4, p. 29, 2009. Cited on page 32.

CHO, H.-Y. *Event-Based control of 4D effects using MPEG RoSE*. Dissertação (Mestrado) — School of Mechanical, Aerospace and Systems Engineering. Korea Advanced Institute of Science and Technology. Master's Thesis, 2010. Cited 2 times on pages 18 and 40.

CHOI, B.; LEE, E.-S.; YOON, K. Streaming media with sensory effect. In: *Information Science and Applications (ICISA), 2011 International Conference on*. [S.l.: s.n.], 2011. p. 1–6. Cited 2 times on pages 18 and 37.

CHU, J. H.; HARLEY, D.; KWAN, J.; MCBRIDE, M.; MAZALEK, A. Sensing history: Contextualizing artifacts with sensory interactions and narrative design. In: *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. New York, NY, USA: ACM, 2016. (DIS '16), p. 1294–1302. ISBN 978-1-4503-4031-1. Available at: <http://doi.acm.org/10.1145/2901790.2901829>. Cited on page 17.

CHU, S.; DOWNES, J. J. Odour-evoked autobiographical memories: Psychological investigations of proustian phenomena. *Chemical Senses*, v. 25, n. 1, p. 111–116, 2000. Cited on page 33.

CLAISSE, C.; PETRELLI, D.; DULAKE, N.; MARSHALL, M.; CIOLFI, L. Multisensory interactive storytelling to augment the visit of a historical house museum. In: *Proceedings of the 2018 Digital Heritage International Congress*. IEEE, 2018. Available at: <http://shura.shu.ac.uk/22646/>. Cited on page 17.

COMSA, I.; TRESTIAN, R.; GHINEA, G. 360° Mulsemedia Experience over Next Generation Wireless Networks - A Reinforcement Learning Approach. In: *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. [S.l.: s.n.], 2018. p. 1–6. ISSN 2472-7814. Cited on page 37.

COMSA, I.-S.; SALEME, E. B.; COVACI, A.; MESFIN, G.; TRESTIAN, R.; SANTOS, C. A. S.; GHINEA, G. Do I Smell Coffee? The Tale of a 360° Mulsemedia Experience. *IEEE MultiMedia*, 2019. ISSN 1070-986X. Cited 2 times on pages 22 and 83.

CONOVER, W. J. *Practical Nonparametric Statistics*. 3. ed. ed. [S.l.]: John Wiley & Sons, Inc., 1998. Cited on page 88.

CONTI, F. *The CHAI libraries*. [S.l.], 2003. Cited 3 times on pages 17, 38, and 39.

COVACI, A.; GHINEA, G.; LIN, C.-H.; HUANG, S.-H.; SHIH, J.-L. Multisensory games-based learning-lessons learnt from olfactory enhancement of a digital board game. *Multimedia Tools and Applications*, Springer, p. 1–19, 2018. Cited 2 times on pages 17 and 33.

COVACI, A.; SALEME, E. B.; MESFIN, G. A.; HUSSAIN, N.; KANI-ZABIHI, E.; GHINEA, G. How do we experience crossmodal correspondent mulsemedia content? *IEEE Transactions on Multimedia*, 2019. Cited on page 23.

COVACI, A.; TRESTIAN, R.; SALEME, E. B.; COMSA, I.-S.; MESFIN, G.; SANTOS, C. A. S.; GHINEA, G. 360° Mulsemedia: A Way to Improve Subjective QoE in 360° Videos. In: *Proceedings of the 27th ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2019. (MM '19), p. 2378–2386. ISBN 978-1-4503-6889-6. Cited 6 times on pages 19, 22, 49, 83, 92, and 126.

COVACI, A.; ZOU, L.; TAL, I.; MUNTEAN, G.-M.; GHINEA, G. Is multimedia multisensorial? - a review of mulsemedia systems. *ACM Computing Surveys*, ACM, New York, NY, USA, v. 51, n. 5, p. 91:1–91:35, set. 2018. ISSN 0360-0300. Cited 3 times on pages 17, 25, and 45.

DANIEAU, F.; FLEUREAU, J.; CABEC, A.; KERBIRIOU, P.; GUILLOTEL, P.; MOLLET, N.; CHRISTIE, M.; LÉCUYER, A. Framework for enhancing video viewing experience with haptic effects of motion. In: IEEE. *IEEE Haptics Symposium*. [S.l.], 2012. p. 541–546. Cited on page 32.

DANIEAU, F.; LÉCUYER, A.; GUILLOTEL, P.; FLEUREAU, J.; MOLLET, N.; CHRISTIE, M. Enhancing audiovisual experience with haptic feedback: a survey on hav. *IEEE transactions on haptics*, IEEE, v. 6, n. 2, p. 193–205, 2013. Cited on page 31.

DELPLANQUE, S.; CHREA, C.; GRANDJEAN, D.; FERDENZI, C.; CAYEUX, I.; PORCHEROT, C.; CALVE, B. L.; SANDER, D.; SCHERER, K. R. How to map the affective semantic space of scents. *Cognition and emotion*, v. 26 5, p. 885–98, 2012. Cited on page 33.

DOBBELSTEIN, D.; HERRDUM, S.; RUKZIO, E. inscent: A wearable olfactory display as an amplification for mobile notifications. In: *Proceedings of the 2017 ACM International Symposium on Wearable Computers*. New York, NY, USA: ACM, 2017. (ISWC '17), p. 130–137. ISBN 978-1-4503-5188-1. Cited on page 35.

DOBBELSTEIN, D.; RUKZIO, E.; HERRDUM, S. Demonstration of inscent: A wearable olfactory display as an amplification for mobile notifications. In: *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. New York, NY, USA: ACM, 2017. (UbiComp '17), p. 229–232. ISBN 978-1-4503-5190-4. Cited 2 times on pages 17 and 39.

EBRAHIMI, T. Quality of multimedia experience: Past, present and future. In: *Proceedings of the 17th ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2009. (MM '09), p. 3–4. ISBN 978-1-60558-608-3. Available at: <http://doi.acm.org/10.1145/1631272.1631275>. Cited on page 26.

EBRAHIMI, T. *Quality of Multimedia Experience: Past, Present and Future*. Keynote at ACM Multimedia, Beijing, China: [s.n.], 2009. <http://www.slideshare.net/touradj_ebrahimi/qoe>. Online; accessed 16 August 2019. Cited on page 26.

EGAN, D.; BRENNAN, S.; BARRETT, J.; QIAO, Y.; TIMMERER, C.; MURRAY, N. An evaluation of heart rate and electrodermal activity as an objective qoe evaluation method for immersive virtual reality environments. In: IEEE. *Quality of Multimedia Experience (QoMEX), 2016 Eighth International Conference on*. [S.l.], 2016. p. 1–6. Cited on page 28.

FENG, M.; LINDEMAN, R. W.; ABDEL-MOATI, H.; LINDEMAN, J. C. Haptic chairio: A system to study the effect of wind and floor vibration feedback on spatial orientation in ves. In: IEEE. *IEEE Symposium on 3D User Interfaces*. [S.l.], 2015. p. 149–150. Cited on page 33.

FETTE, I.; MELNIKOV, A. *The WebSocket Protocol. RFC 6455*. 2011. <https://tools.ietf.org/html/rfc6455>. Online; accessed 19 October 2017. Cited on page 63.

FUCCIO, R. D.; PONTICORVO, M.; FERRARA, F.; MIGLINO, O. Digital and multisensory storytelling: narration with smell, taste and touch. In: SPRINGER. *European Conference on Technology Enhanced Learning*. [S.l.], 2016. p. 509–512. Cited on page 33.

GALLACHER, C.; MOHTAT, A.; DING, S.; KÖVECSES, J. Toward open-source portable haptic displays with visual-force-tactile feedback colocation. In: IEEE. *Haptics Symposium (HAPTICS), 2016 IEEE*. [S.l.], 2016. p. 65–71. Cited 3 times on pages 17, 32, and 39.

GALSTER, M.; AVGERIOU, P. Chapter 6 - supporting variability through agility to achieve adaptable architectures. In: BABAR, M. A.; BROWN, A. W.; MISTRIK, I. (Ed.). *Agile Software Architecture*. Boston: Morgan Kaufmann, 2014. p. 139 – 159. ISBN 978-0-12-407772-0. Cited on page 19.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. *Design Patterns: Elements of Reusable Object-oriented Software*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995. ISBN 0-201-63361-2. Cited on page 59.

GHINEA, G.; ADEMOYE, O. User perception of media content association in olfaction-enhanced multimedia. *ACM Trans. on Multimedia Computing, Communications, and Applications (TOMM)*, ACM, v. 8, n. 4, p. 52, 2012. Cited on page 33.

GHINEA, G.; ADEMOYE, O. A. Perceived synchronization of olfactory multimedia. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, v. 40, n. 4, p. 657–663, July 2010. ISSN 1083-4427. Cited on page 18.

GHINEA, G.; TIMMERER, C.; LIN, W.; GULLIVER, S. R. Mulsemedia: State of the Art, Perspectives, and Challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, ACM, New York, NY, USA, v. 11, n. 1s, p. 17:1–17:23, out. 2014. ISSN 1551-6857. Cited 3 times on pages 17, 35, and 45.

GITHUT. *GitHut - A small place to discover languages in GitHub*. 2019. Available at: <https://githut.info/>. Cited on page 59.

GREENE, J.; D'OLIVEIRA, M. *Learning to use statistical tests in psychology*. [S.l.]: McGraw-Hill Education (UK), 2005. Cited on page 94.

GROHER, I.; WEINREICH, R. Supporting variability management in architecture design and implementation. In: *2013 46th Hawaii International Conference on System Sciences*. [S.l.: s.n.], 2013. p. 4995–5004. ISSN 1530-1605. Cited on page 19.

GU, X.; ZHANG, Y.; SUN, W.; BIAN, Y.; ZHOU, D.; KRISTENSSON, P. O. Dexmo: An inexpensive and lightweight mechanical exoskeleton for motion capture and force feedback in vr. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. [S.l.]: ACM, 2016. p. 1991–1995. ISBN 978-1-4503-3362-7. Cited on page 31.

HASEGAWA, K.; QIU, L.; SHINODA, H. Midair ultrasound fragrance rendering. *IEEE Transactions on Visualization and Computer Graphics*, v. 24, n. 4, p. 1477–1485, April 2018. ISSN 1077-2626. Cited on page 35.

HERRERA, N. S.; MCMAHAN, R. P. Development of a simple and low-cost olfactory display for immersive media experiences. In: ACM. *Proceedings of the 2nd ACM International Workshop on Immersive Media Experiences*. [S.l.], 2014. p. 1–6. Cited 2 times on pages 34 and 35.

HOWELL, M. J.; HERRERA, N. S.; MOORE, A. G.; MCMAHAN, R. P. A reproducible olfactory display for exploring olfaction in immersive media experiences. *Multimedia Tools and Applications*, v. 75, n. 20, p. 12311–12330, Oct 2016. ISSN 1573-7721. Cited 5 times on pages 17, 33, 34, 35, and 39.

ISCHER, M.; BARON, N.; MERMOUD, C.; CAYEUX, I.; PORCHEROT, C.; SANDER, D.; DELPLANQUE, S. How incorporation of scents could enhance immersive virtual experiences. *Frontiers in Psychology*, v. 5, 2014. ISSN 1664-1078. Cited on page 33.

ITKOWITZ, B.; HANDLEY, J.; ZHU, W. The openhaptics/spl trade/ toolkit: a library for adding 3d touch/spl trade/ navigation and haptics to graphics applications. In: *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*. [S.l.: s.n.], 2005. p. 590–591. Cited 2 times on pages 17 and 38.

JALAL, L.; ANEDDA, M.; POPESCU, V.; MURRONI, M. Qoe assessment for broadcasting multi sensorial media in smart home scenario. In: *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. [S.l.: s.n.], 2018. p. 1–5. ISSN 2155-5052. Cited 4 times on pages 18, 19, 41, and 42.

JALAL, L.; ANEDDA, M.; POPESCU, V.; MURRONI, M. QoE Assessment for IoT-Based Multi Sensorial Media Broadcasting. *IEEE Transactions on Broadcasting*, v. 64, n. 2, p. 552–560, June 2018. ISSN 0018-9316. Cited 8 times on pages 17, 19, 26, 27, 28, 49, 92, and 127.

KAKLANIS, N.; VOTIS, K.; TZOVARAS, D. Adding haptic feedback to web applications towards improving end-users' cognitive capabilities. In: *2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*. [S.l.: s.n.], 2015. p. 245–249. Cited 2 times on pages 17 and 38.

KARUNANAYAKA, K.; JOHARI, N.; HARIRI, S.; CAMELIA, H.; BIELAWSKI, K. S.; CHEOK, A. D. New thermal taste actuation technology for future multisensory virtual reality and internet. *IEEE Transactions on Visualization and Computer Graphics*, v. 24, n. 4, p. 1496–1505, April 2018. ISSN 1077-2626. Cited 2 times on pages 36 and 40.

KEIGHREY, C.; FLYNN, R.; MURRAY, S.; MURRAY, N. A qoe evaluation of immersive augmented and virtual reality speech & language assessment applications. In: IEEE. *Quality of Multimedia Experience (QoMEX), 2017 Ninth International Conference on*. [S.l.], 2017. p. 1–6. Cited on page 29.

KERVEGANT, C.; RAYMOND, F.; GRAEFF, D.; CASTET, J. Touch hologram in mid-air. In: *ACM SIGGRAPH 2017 Emerging Technologies*. New York, NY, USA: ACM, 2017. (SIGGRAPH '17), p. 23:1–23:2. ISBN 978-1-4503-5012-9. Cited on page 32.

KIM, J. R.; OSGOUEI, R. H.; CHOI, S. Effects of visual and haptic latency on touchscreen interaction: A case study using painting task. In: IEEE. *World Haptics Conf., 2017 IEEE*. [S.l.], 2017. p. 159–164. Cited 2 times on pages 19 and 25.

KIM, S.-K.; JOO, Y. S. Sensible media simulation in an automobile application and human responses to sensory effects. *ETRI Journal*, ETRI, v. 35, n. 6, p. 1001–1010, dez. 2014. ISSN 1225-6463. Cited 2 times on pages 18 and 40.

KIM, Y.; CHA, J.; RYU, J.; OAKLEY, I. A tactile glove design and authoring system for immersive multimedia. *IEEE MultiMedia*, IEEE, v. 17, n. 3, 2010. Cited on page 32.

KOLSANOV, A.; NAZARYAN, A.; IVASCHENKO, A.; KUZMIN, A. Intelligent sdk for 3d surgery simulation. In: *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*. [S.l.: s.n.], 2016. p. 384–387. Cited 2 times on pages 17 and 39.

KOVATSCH, M.; LANTER, M.; SHELBY, Z. Californium: Scalable cloud services for the internet of things with coap. In: *2014 International Conference on the Internet of Things (IOT)*. [S.l.: s.n.], 2014. p. 1–6. Cited on page 62.

KUMAGAI, M. Development of a ball drive unit using partially sliding rollers - an alternative mechanism for semi-omnidirectional motion. In: *IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2010. p. 3352–3357. Cited on page 32.

LIN, Y.; YANG, M.; LIN, Y. Low-cost 4d experience theater using home appliances. *IEEE Transactions on Multimedia*, p. 1–1, 2018. ISSN 1520-9210. Cited 2 times on pages 18 and 42.

LUQUE, F. P.; GALLOSO, I.; FEIJOO, C.; MARTÍN, C. A.; CISNEROS, G. Integration of multisensorial stimuli and multimodal interaction in a hybrid 3dtv system. *ACM Trans. Multimedia Comput. Commun. Appl.*, ACM, New York, NY, USA, v. 11, n. 1s, p. 16:1–16:22, out. 2014. ISSN 1551-6857. Cited 2 times on pages 18 and 41.

MARTIN, S.; HILLIER, N. Characterisation of the novint falcon haptic device for application as a robot manipulator. In: CITESEER. *Australasian Conference on Robotics and Automation (ACRA)*. [S.l.], 2009. p. 291–292. Cited on page 38.

MATSUKURA, H.; YONEDA, T.; ISHIDA, H. Smelling screen: development and evaluation of an olfactory display system for presenting a virtual odor source. *IEEE Trans. Vis. Comput. Graph.*, IEEE, v. 19, n. 4, p. 606–615, 2013. Cited on page 33.

MCGOOKIN, D.; ESCOBAR, D. Hajukone: Developing an open source olfactory device. In: ACM. *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. [S.l.], 2016. p. 1721–1728. Cited 3 times on pages 17, 34, and 39.

MESFIN, G. A.; HUSSAIN, N.; KANI-ZABIHI, E.; COVACI, A.; SALEME, E. B.; GHINEA, G. QoE of Cross-modally Mapped Mulsemedia: an Assessment Using Eye Gaze and Heart Rate. *Multimedia Tools and Applications*, 2019. Cited on page 23.

MEULEN, E. van der; CIDOTA, M. A.; LUKOSCH, S. G.; BANK, P. J.; HELM, A. J. van der; VISCH, V. T. A haptic serious augmented reality game for motor assessment of parkinson's disease patients. In: IEEE. *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. [S.l.], 2016. p. 102–104. Cited on page 17.

MISTRIK, I.; GALSTER, M.; MAXIM, B. R. *Software Engineering for Variability Intensive Systems: Foundations and Applications*. [S.l.]: CRC Press, 2019. Cited on page 19.

MONKS, J.; OLARU, A.; TAL, I.; MUNTEAN, G. M. Quality of experience assessment of 3d video synchronised with multisensorial media components. In: *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. [S.l.: s.n.], 2017. p. 1–6. Cited 4 times on pages 17, 26, 27, and 28.

MONTAGUD, M.; CESAR, P.; BORONAT, F.; JANSEN, J. *MediaSync: Handbook on Multimedia Synchronization*. [S.l.]: Springer, 2018. Cited on page 97.

MOON, T.; KIM, G. J. Design and evaluation of a wind display for virtual reality. In: ACM. *Proceedings of the ACM symposium on Virtual reality software and technology*. [S.l.], 2004. p. 122–128. Cited on page 32.

MORIMOTO, T. K.; BLIKSTEIN, P.; OKAMURA, A. M. [d81] hapkit: An open-hardware haptic device for online education. In: IEEE. *Haptics Symposium (HAPTICS), 2014 IEEE*. [S.l.], 2014. p. 1–1. Cited on page 32.

MURER, M.; ASLAN, I.; TSCHELIGI, M. Loll io: exploring taste as playful modality. In: ACM. *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*. [S.l.], 2013. p. 299–302. Cited on page 36.

MURRAY, N.; ADEMOYE, O. A.; GHINEA, G.; QIAO, Y.; MUNTEAN, G.; LEE, B. Olfactory-enhanced multimedia video clips datasets. In: *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*. [S.l.: s.n.], 2017. p. 1–5. ISSN 2472-7814. Cited 2 times on pages 18 and 30.

MURRAY, N.; ADEMOYE, O. A.; GHINEA, G.; MUNTEAN, G.-M. A tutorial for olfaction-based multisensorial media application design and evaluation. *ACM Comp. Surveys*, ACM, New York, NY, USA, v. 50, n. 5, p. 67:1–67:30, 2017. ISSN 0360-0300. Cited 4 times on pages 17, 29, 39, and 45.

MURRAY, N.; LEE, B.; QIAO, Y.; MUNTEAN, G.-M. Multiple-scent enhanced multimedia synchronization. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, ACM, v. 11, n. 1s, p. 12, 2014. Cited on page 34.

MURRAY, N.; LEE, B.; QIAO, Y.; MUNTEAN, G.-M. The influence of human factors on olfaction based mulsemedia quality of experience. In: *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*. [S.l.: s.n.], 2016. p. 1–6. Cited on page 18.

MURRAY, N.; LEE, B.; QIAO, Y.; MUNTEAN, G.-M. The impact of scent type on olfaction-enhanced multimedia quality of experience. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, v. 47, n. 9, p. 2503–2515, Sept 2017. ISSN 2168-2216. Cited 7 times on pages 17, 18, 25, 26, 27, 28, and 34.

MURRAY, N.; QIAO, Y.; LEE, B.; MUNTEAN, G.-M.; KARUNAKAR, A. K. Age and gender influence on perceived olfactory & visual media synchronization. In: IEEE. *Multimedia and Expo (ICME), 2013 IEEE International Conference on*. [S.l.], 2013. p. 1–6. Cited on page 34.

MURRAY, N.; QIAO, Y.; LEE, B.; KARUNAKAR, A. K.; MUNTEAN, G.-M. Subjective evaluation of olfactory and visual media synchronization. In: *Proceedings of the 4th ACM Multimedia Systems Conference*. New York, NY, USA: ACM, 2013. (MMSys '13), p. 162–171. ISBN 978-1-4503-1894-5. Cited 2 times on pages 18 and 33.

MURRAY, N.; QIAO, Y.; LEE, B.; MUNTEAN, G.-M. User-profile-based perceived olfactory and visual media synchronization. *ACM Trans. Multimedia Comput. Commun. Appl.*, ACM, New York, NY, USA, v. 10, n. 1s, p. 11:1–11:24, jan. 2014. ISSN 1551-6857. Available at: <http://doi.acm.org/10.1145/2540994>. Cited on page 18.

NAKAMOTO, T.; OTAGURO, S.; KINOSHITA, M.; NAGAHAMA, M.; OHINISHI, K.; ISHIDA, T. Cooking up an interactive olfactory game display. *IEEE Comput. Graph. and Applications*, v. 28, n. 1, p. 75–78, Jan 2008. ISSN 0272-1716. Cited on page 33.

NAKAMURA, H.; MIYASHITA, H. Development and evaluation of interactive system for synchronizing electric taste and visual content. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2012. (CHI '12), p. 517–520. ISBN 978-1-4503-1015-4. Cited on page 25.

NORMAN, G. Likert scales, levels of measurement and the "laws" of statistics. *Advances in Health Sciences Education*, v. 15, n. 5, p. 625–632, Dec 2010. ISSN 1573-1677. Available at: <https://doi.org/10.1007/s10459-010-9222-y>. Cited on page 96.

OBRIST, M.; GATTI, E.; MAGGIONI, E.; VI, C. T.; VELASCO, C. Multisensory experiences in HCI. *IEEE MultiMedia*, v. 24, n. 2, p. 9–13, Apr 2017. ISSN 1070-986X. Cited 3 times on pages 19, 44, and 45.

OBRIST, M.; VELASCO, C.; VI, C.; RANASINGHE, N.; ISRAR, A.; CHEOK, A.; SPENCE, C.; GOPALAKRISHNAKONE, P. Sensing the future of HCI: touch, taste, and smell user interfaces. *interactions*, ACM, v. 23, n. 5, p. 40–49, 2016. Cited on page 17.

PETIT, O.; VELASCO, C.; SPENCE, C. Digital sensory marketing: Integrating new technologies into multisensory online experience. *Journal of Interactive Marketing*, v. 45, p. 42 – 61, 2019. ISSN 1094-9968. Available at: <http://www.sciencedirect.com/science/article/pii/S1094996818300513>. Cited on page 17.

PICHON, A. M.; COPPIN, G.; CAYEUX, I.; PORCHEROT, C.; SANDER, D.; DELPLANQUE, S. Sensitivity of physiological emotional measures to odors depends on the product and the pleasantness ranges used. In: *Front. Psychol.* [S.l.: s.n.], 2015. Cited on page 33.

PIMENTEL, V.; NICKERSON, B. G. Communicating and displaying real-time data with websocket. *IEEE Internet Computing*, v. 16, n. 4, p. 45–53, July 2012. ISSN 1089-7801. Cited on page 63.

PLATTIG, K.-H.; INNITZER, J. Taste qualities elicited by electric stimulation of single human tongue papillae. *Pflügers Archiv*, v. 361, n. 2, p. 115–120, Jan 1976. ISSN 1432-2013. Cited on page 36.

PORCHEROT, C.; DELPLANQUE, S.; RAVIOT-DERRIEN, S.; CALVE, B. L.; CHREA, C.; GAUDREAU, N.; CAYEUX, I. How do you feel when you smell this? optimization of a verbal measurement of odor-elicited emotions. *Food Quality and Preference*, v. 21, n. 8, p. 938 – 947, 2010. ISSN 0950-3293. Cited on page 33.

PRASAD, M.; TAELE, P.; GOLDBERG, D.; HAMMOND, T. A. Haptimoto: Turn-by-turn haptic route guidance interface for motorcyclists. In: ACM. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. [S.l.], 2014. p. 3597–3606. Cited on page 31.

RAINER, B.; TIMMERER, C. A generic utility model representing the quality of sensory experience. *ACM Trans. Multimedia Comput. Commun. Appl.*, ACM, New York, NY, USA, v. 11, n. 1s, p. 14:1–14:17, out. 2014. ISSN 1551-6857. Cited on page 29.

RAINER, B.; WALTL, M.; CHENG, E.; SHUJAU, M.; TIMMERER, C.; DAVIS, S.; BURNETT, I.; RITZ, C.; HELLWAGNER, H. Investigating the impact of sensory effects on the quality of experience and emotional response in web videos. In: *2012 Fourth International Workshop on Quality of Multimedia Experience*. [S.l.: s.n.], 2012. p. 278–283. Cited 3 times on pages 17, 26, and 27.

RAMALHO, J.; CHAMBEL, T. Windy sight surfers: Sensing and awareness of 360° immersive videos on the move. In: *Proceedings of the 11th European Conference on Interactive TV and Video*. New York, NY, USA: ACM, 2013. (EuroITV '13), p. 107–116. ISBN 978-1-4503-1951-5. Cited on page 32.

RANASINGHE, N.; CHEOK, A. D.; NAKATSU, R. Taste/ip: The sensation of taste for digital communication. In: *Proceedings of the 14th ACM International Conference on Multimodal Interaction*. New York, NY, USA: ACM, 2012. (ICMI '12), p. 409–416. ISBN 978-1-4503-1467-1. Available at: <http://doi.acm.org/10.1145/2388676.2388768>. Cited on page 40.

RANASINGHE, N.; DO, E. Y.-L. Digital lollipop: Studying electrical stimulation on the human tongue to simulate taste sensations. *ACM Transactions on Multimedia Computing, Comm., and Applications (TOMM)*, ACM, v. 13, n. 1, p. 5, 2017. Cited on page 36.

RANASINGHE, N.; JAIN, P.; TRAM, N. T. N.; KOH, K. C. R.; TOLLEY, D.; KARWITA, S.; LIEN-YA, L.; LIANGKUN, Y.; SHAMAIAH, K.; TUNG, C. E. W.; YEN, C. C.; DO, E. Y.-L. Season traveller: Multisensory narration for enhancing the virtual reality experience. In:

Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. New York, NY, USA: ACM, 2018. (CHI '18), p. 577:1–577:13. ISBN 978-1-4503-5620-6. Available at: <http://doi.acm.org/10.1145/3173574.3174151>. Cited on page 17.

RANASINGHE, N.; NGUYEN, T. N. T.; LIANGKUN, Y.; LIN, L.-Y.; TOLLEY, D.; DO, E. Y.-L. Vocktail: A virtual cocktail for pairing digital taste, smell, and color sensations. In: *Proceedings of the 2017 ACM on Multimedia Conference*. New York, NY, USA: ACM, 2017. (MM '17), p. 1139–1147. ISBN 978-1-4503-4906-2. Cited on page 40.

RANK, M.; SHI, Z.; HIRCHE, S. Perception of Delay in Haptic Telepresence Systems. *Presence: Teleoperators and Virtual Environments*, v. 19, n. 5, p. 389–399, oct 2010. ISSN 1054-7460. Cited 2 times on pages 19 and 26.

RÉTIVEAU, A. N.; IV, E. C.; MILLIKEN, G. A. Common and specific effects of fine fragrances on the mood of women. *Journal of Sensory Studies*, v. 19, n. 5, p. 373–394, 2005. Cited on page 33.

SALEME, E. B.; CELESTRINI, J. R.; SANTOS, C. A. S. Time Evaluation for the Integration of a Gestural Interactive Application with a Distributed Mulsemedia Platform. In: *Proceedings of the 8th ACM on Multimedia Systems Conference - MMSys'17*. New York, New York, USA: ACM Press, 2017. p. 308–314. ISBN 9781450350020. Cited 4 times on pages 23, 41, 85, and 86.

SALEME, E. B.; COVACI, A.; MESFIN, G.; SANTOS, C. A. S.; GHINEA, G. Mulsemedia DIY: A Survey of Devices and a Tutorial for Building Your Own Mulsemedia Environment. *ACM Computing Surveys*, v. 52, n. 3, p. 58:1–58:29, jun. 2019. ISSN 0360-0300. Cited 5 times on pages 17, 22, 30, 38, and 40.

SALEME, E. B.; COVACI, A.; MESFIN, G.; SANTOS, C. A. S.; GHINEA, G. Sumarização de dispositivos de efeitos multissensoriais para interações humano-computador. In: *Extended Proceedings of IHC '19: XVIII Brazilian Symposium on Human Factors in Computing Systems*. Porto Alegre, RS, Brazil: SBC, 2019. (IHC '19). Cited on page 23.

SALEME, E. B.; SANTOS, C. A. S. PlaySEM: a Platform for Rendering MulSeMedia Compatible with MPEG-V. In: *Proceedings of the 21st Brazilian Symposium on Multimedia and the Web - WebMedia '15*. New York, New York, USA: ACM Press, 2015. p. 145–148. ISBN 9781450339599. Cited 6 times on pages 18, 19, 28, 41, 58, and 127.

SALEME, E. B.; SANTOS, C. A. S.; FALBO, R. A.; GHINEA, G.; ANDRES, F. Towards a reference ontology on mulsemedia systems. In: *Proceedings of the 10th International Conference on Management of Digital EcoSystems*. New York, NY, USA: ACM, 2018. (MEDES '18), p. 23–30. ISBN 978-1-4503-5622-0. Cited on page 23.

SALEME, E. B.; SANTOS, C. A. S.; FALBO, R. A.; GHINEA, G.; ANDRES, F. MulseOnto: a Reference Ontology to Support the Design of Mulsemedia Systems. *Journal of Universal Computer Science*, v. 25, n. 9, 2019. Cited on page 23.

SALEME, E. B.; SANTOS, C. A. S.; GHINEA, G. Improving response time interval in networked event-based mulsemedia systems. In: *Proceedings of the 9th ACM Multimedia Systems Conference*. New York, NY, USA: ACM, 2018. (MMSys '18), p. 216–224. ISBN 978-1-4503-5192-8. Cited 4 times on pages 22, 41, 71, and 85.

SALEME, E. B.; SANTOS, C. A. S.; GHINEA, G. Coping with the challenges of delivering multiple sensorial media. *IEEE MultiMedia*, v. 26, n. 2, p. 66–75, April 2019. ISSN 1070-986X. Cited 4 times on pages 20, 22, 45, and 77.

SALEME, E. B.; SANTOS, C. A. S.; GHINEA, G. A mulsemedia framework for delivering sensory effects to heterogeneous systems. *Multimedia Systems*, v. 25, n. 4, p. 421–447, Aug 2019. ISSN 1432-1882. Cited 14 times on pages 19, 22, 48, 58, 61, 64, 65, 67, 69, 70, 71, 77, 78, and 82.

SALMINEN, K.; RANTALA, J.; ISOKOSKI, P.; LEHTONEN, M.; MüLLER, P.; KARJALAINEN, M.; VäLIAHO, J.; KONTUNEN, A.; NIEMINEN, V.; LEIVO, J.; TELEMBECI, A. A.; LEKKALA, J.; KALLIO, P.; SURAKKA, V. Olfactory display prototype for presenting and sensing authentic and synthetic odors. In: *Proceedings of the 20th ACM International Conference on Multimodal Interaction*. New York, NY, USA: ACM, 2018. (ICMI '18), p. 73–77. ISBN 978-1-4503-5692-3. Cited on page 35.

SANFILIPPO, F.; WEUSTINK, P. B. T.; PETTERSEN, K. Y. A coupling library for the force dimension haptic devices and the 20-sim modelling and simulation environment. In: *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*. [S.l.: s.n.], 2015. p. 000168–000173. Cited 2 times on pages 17 and 39.

SANTOS, C. A. S.; NETO, A. N. R.; SALEME, E. B. An Event Driven Approach for Integrating Multi-sensory Effects to Interactive Environments. In: *2015 IEEE International Conference on Systems, Man, and Cybernetics*. [S.l.]: IEEE, 2015. p. 981–986. ISBN 978-1-4799-8697-2. Cited 2 times on pages 24 and 86.

SARID, O.; ZACCAI, M. Changes in mood states are induced by smelling familiar and exotic fragrances. *Frontiers in psychology*, Frontiers, v. 7, p. 1724, 2016. Cited on page 33.

SCHEGGI, S.; AGGRAVI, M.; PRATTICHIZZO, D. Cooperative navigation for mixed human–robot teams using haptic feedback. *IEEE Transactions on Human-Machine Systems*, IEEE, v. 47, n. 4, p. 462–473, 2017. Cited on page 31.

SCHORR, S. B.; OKAMURA, A. M. Fingertip tactile devices for virtual object manipulation and exploration. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017. (CHI '17), p. 3115–3119. ISBN 978-1-4503-4655-9. Cited on page 32.

SERRANO, B.; BAÑOS, R. M.; BOTELLA, C. Virtual reality and stimulation of touch and smell for inducing relaxation: A randomized controlled trial. *Computers in Human Behavior*, Elsevier, v. 55, p. 1–8, 2016. Cited on page 17.

SHAH, N.; BASTERIS, A.; AMIRABDOLLAHIAN, F. Design parameters in multimodal games for rehabilitation. *Games For Health: Research, Development, and Clinical Applications*, Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 3, n. 1, p. 13–20, 2014. Cited on page 31.

SHAKERI, G.; WILLIAMSON, J. H.; BREWSTER, S. May the force be with you: Ultrasound haptic feedback for mid-air gesture interaction in cars. In: *Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. New York, NY, USA: ACM, 2018. p. 1–10. ISBN 978-1-4503-5946-7. Cited on page 32.

SHELBY, Z.; HARTKE, K.; BORMANN, C. *The Constrained Application Protocol (CoAP). RFC 7252*. 2014. <https://tools.ietf.org/html/rfc7252>. Online; accessed 19 October 2017. Cited on page 62.

SOMMERVILLE, I. *Software Engineering*. 10th. ed. [S.l.]: Pearson, 2015. ISBN 0133943038, 9780133943030. Cited on page 60.

SPENCE, C.; OBRIST, M.; VELASCO, C.; RANASINGHE, N. Digitizing the chemical senses: possibilities & pitfalls. *International Journal of Human-Computer Studies*, Elsevier, v. 107, p. 62–74, 2017. Cited on page 35.

SPENCE, C.; PIQUERAS-FISZMAN, B. Oral-somatosensory contributions to flavor perception and the appreciation of food and drink. In: *Multisensory Flavor Perception*. [S.l.]: Elsevier, 2016. p. 59–79. Cited on page 35.

SPENCER, B. S. Incorporating the sense of smell into patient and haptic surgical simulators. *IEEE Transactions on Information Technology in Biomedicine*, v. 10, n. 1, p. 168–173, Jan 2006. ISSN 1089-7771. Cited on page 17.

STONE, R. J.; GUEST, R.; MAHONEY, P.; LAMB, D.; GIBSON, C. A 'mixed reality' simulator concept for future medical emergency response team training. *Journal of the Royal Army Medical Corps*, British Medical Journal Publishing Group, v. 163, n. 4, p. 280–287, 2017. ISSN 0035-8665. Available at: <https://jramc.bmj.com/content/163/4/280>. Cited on page 17.

SUK, C. B.; HYUN, J. S.; YONG, L. H. Sensory effect metadata for smmd media service. In: PERRY, M.; SASAKI, H.; EHMANN, M.; BELLOT, G. O.; DINI, O. (Ed.). *ICIW*. [S.l.]: IEEE Computer Society, 2009. p. 649–654. Cited 2 times on pages 18 and 37.

SULEMA, Y. Mulsemedia vs. Multimedia: State of the art and future trends. In: *2016 International Conference on Systems, Signals and Image Processing (IWSSIP)*. [S.l.: s.n.], 2016. p. 1–5. Cited on page 17.

SULEMA, Y. Asampl: Programming language for mulsemedia data processing based on algebraic system of aggregates. In: AUER, M. E.; TSIATSOS, T. (Ed.). *Interactive Mobile Communication Technologies and Learning*. Cham: Springer International Publishing, 2018. p. 431–442. ISBN 978-3-319-75175-7. Cited on page 37.

TIMMERER, C.; WALTL, M.; RAINER, B.; HELLWAGNER, H. Assessing the quality of sensory experience for multimedia presentations. *Signal Processing: Image Communication*, v. 27, n. 8, p. 909 – 916, 2012. ISSN 0923-5965. Cited on page 93.

TOET, A.; SCHAIK, M. van; THEUNISSEN, N. C. M. No effect of ambient odor on the affective appraisal of a desktop virtual environment with signs of disorder. *PLOS ONE*, Public Library of Science, v. 8, n. 11, p. 1–15, 11 2013. Cited on page 33.

UPnP Forum. *UPnP Device Architecture 1.0*. 2008. <http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0.pdf>. Online; accessed 05 April 2018. Cited on page 62.

VI, C. T.; ARTHUR, D.; OBRIST, M. Tastebud: Bring taste back into the game. In: *Proceedings of the 3rd International Workshop on Multisensory Approaches to Human-Food Interaction*. New York, NY, USA: ACM, 2018. (MHFI'18), p. 1:1–1:5. ISBN 978-1-4503-6074-6. Available at: <http://doi.acm.org/10.1145/3279954.3279955>. Cited on page 17.

VI, C. T.; MARZO, A.; ABLART, D.; MEMOLI, G.; SUBRAMANIAN, S.; DRINKWATER, B.; OBRIST, M. Tastyfloats: A contactless food delivery system. In: *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. New York, NY, USA: ACM, 2017. (ISS '17), p. 161–170. ISBN 978-1-4503-4691-7. Cited 2 times on pages 36 and 40.

WALTL, M.; RAINER, B.; TIMMERER, C.; HELLWAGNER, H. An end-to-end tool chain for sensory experience based on mpeg-v. *Image Commun.*, Elsevier Science Inc., New York, NY, USA, v. 28, n. 2, p. 136–150, fev. 2013. ISSN 0923-5965. Cited 5 times on pages 17, 18, 27, 40, and 92.

WALTL, M.; TIMMERER, C.; HELLWAGNER, H. A test-bed for quality of multimedia experience evaluation of sensory effects. In: *2009 International Workshop on Quality of Multimedia Experience*. [S.l.: s.n.], 2009. p. 145–150. Cited 2 times on pages 24 and 37.

WALTL, M.; TIMMERER, C.; HELLWAGNER, H. Improving the quality of multimedia experience through sensory effects. In: *2010 Second International Workshop on Quality of Multimedia Experience (QoMEX)*. [S.l.: s.n.], 2010. p. 124–129. Cited 5 times on pages 17, 18, 26, 27, and 92.

WALTL, M.; TIMMERER, C.; HELLWAGNER, H. Increasing the user experience of multimedia presentations with sensory effects. In: *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*. [S.l.: s.n.], 2010. p. 1–4. ISSN 2158-5873. Cited on page 18.

WALTL, M.; TIMMERER, C.; RAINER, B.; HELLWAGNER, H. Sensory effect dataset and test setups. In: *Fourth International Workshop on Quality of Multimedia Experience (QoMEX'12)*. [S.l.: s.n.], 2012. p. 115–120. Cited 4 times on pages 18, 30, 93, and 94.

WALTL, M.; TIMMERER, C.; RAINER, B.; HELLWAGNER, H. Sensory effects for ambient experiences in the world wide web. *Multimedia Tools and Applications*, v. 70, n. 2, p. 1141–1160, May 2014. Available at: <https://doi.org/10.1007/s11042-012-1099-8>. Cited on page 18.

WEI, J.; WANG, X.; PEIRIS, R. L.; CHOI, Y.; MARTINEZ, X. R.; TACHE, R.; KOH, J. T. K. V.; HALUPKA, V.; CHEOK, A. D. Codine: An interactive multi-sensory system for remote dining. In: *Proceedings of the 13th International Conference on Ubiquitous Computing*. New York, NY, USA: ACM, 2011. (UbiComp '11), p. 21–30. ISBN 978-1-4503-0630-0. Available at: <http://doi.acm.org/10.1145/2030112.2030116>. Cited on page 36.

WHITMIRE, E.; BENKO, H.; HOLZ, C.; OFEK, E.; SINCLAIR, M. Haptic revolver: Touch, shear, texture, and shape rendering on a reconfigurable virtual reality controller. In: ACM. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. [S.l.], 2018. p. 86. Cited on page 32.

WINTER, J. C. F. D.; DODOU, D. Five-point likert items: t test versus mann-whitney-wilcoxon. *Practical Assessment, Research & Evaluation*, Dr. Lawrence M. Rudner, v. 15, n. 11, p. 1–12, 2010. Cited on page 96.

YAMADA, T.; YOKOYAMA, S.; TANIKAWA, T.; HIROTA, K.; HIROSE, M. Wearable olfactory display: Using odor in outdoor environment. In: IEEE. *null*. [S.l.], 2006. p. 199–206. Cited on page 33.

YANAGIDA, Y. A survey of olfactory displays: Making and delivering scents. In: *SENSORS, 2012 IEEE*. [S.l.: s.n.], 2012. p. 1–4. ISSN 1930-0395. Cited on page 33.

YANAGIDA, Y.; KAWATO, S.; NOMA, H.; TETSUTANI, N.; TOMONO, A. A nose-tracked, personal olfactory display. In: *ACM SIGGRAPH 2003 Sketches &Amp; Applications*. [S.l.: s.n.], 2003. p. 1–1. Cited on page 33.

YECIES, B. Transnational collaboration of the multisensory kind: exploiting korean 4d cinema in china. *Media International Australia*, SAGE Publications Sage UK: London, England, v. 159, n. 1, p. 22–31, 2016. Cited on page 17.

YOON, K. End-to-end framework for 4-d broadcasting based on mpeg-v standard. *Image Commun.*, Elsevier Science Inc., New York, NY, USA, v. 28, n. 2, p. 127–135, fev. 2013. ISSN 0923-5965. Cited 2 times on pages 18 and 37.

YOON, K.; KIM, S.-K.; HAN, J. J.; HAN, S.; PREDA, M. *MPEG-V: Bridging the Virtual and Real World*. 1st. ed. [S.l.]: Academic Press, 2015. ISBN 0124201407, 9780124201408. Cited on page 18.

YUAN, Z.; BI, T.; MUNTEAN, G.-M.; GHINEA, G. Perceived synchronization of mulsemedia services. *IEEE Transactions on Multimedia*, v. 17, n. 7, p. 957–966, July 2015. ISSN 1520-9210. Cited on page 25.

YUAN, Z.; CHEN, S.; GHINEA, G.; MUNTEAN, G.-M. User quality of experience of mulsemedia applications. *ACM Trans. Multimedia Comput. Commun. Appl.*, ACM, New York, NY, USA, v. 11, n. 1s, p. 15:1–15:19, out. 2014. ISSN 1551-6857. Cited 4 times on pages 17, 26, 27, and 33.

YUAN, Z.; GHINEA, G.; MUNTEAN, G.-M. Beyond multimedia adaptation: Quality of experience-aware multi-sensorial media delivery. *IEEE Transactions on Multimedia*, IEEE, v. 17, n. 1, p. 104–117, 2015. Cited on page 33.

ZOU, L.; TAL, I.; COVACI, A.; IBARROLA, E.; GHINEA, G.; MUNTEAN, G.-M. Can multisensorial media improve learner experience? In: ACM. *Proceedings of the 8th ACM on Multimedia Systems Conference*. [S.l.], 2017. p. 315–320. Cited 2 times on pages 17 and 34.

Appendix

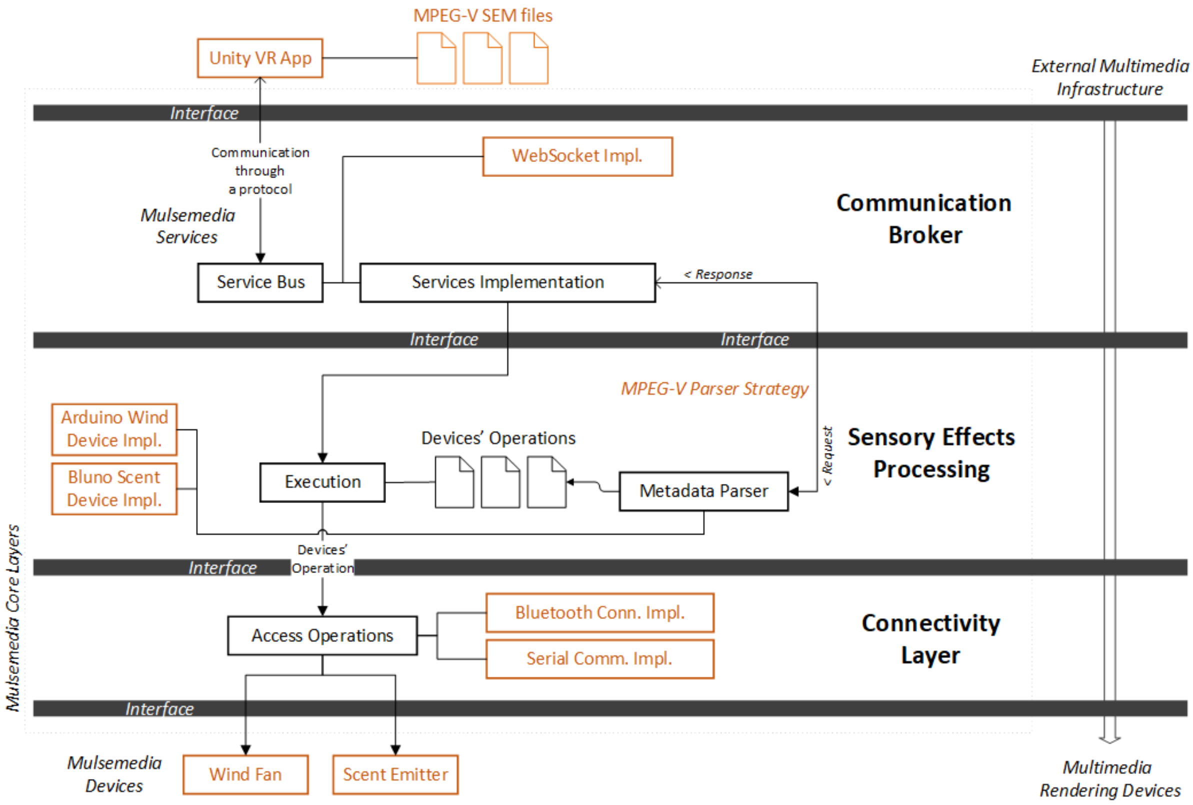
APPENDIX A – Conceptual Architecture Instantiation

This chapter shows two instantiations of the conceptual architecture presented in Chapter 3 (Section 3.2). Items painted orange are the components instantiated within the architecture.

A.1 360° Mulsemedia System

Figure 29 presents the instantiation of the 360° mulsemedia system described in the work of Covaci et al. (2019b). An HMD with a portable scent diffuser attached to it and a wind device were employed to engage users in a 360° environment augmented with smell and airflow. Video clips annotated with MPEG-V are played on 360° Unity mobile application installed on a smartphone attached to the HMD while the delivery of sensory effects is provided by PlaySEM SER 2.

Figure 29 – A 360° Mulsemedia System instantiated in the Conceptual architecture.

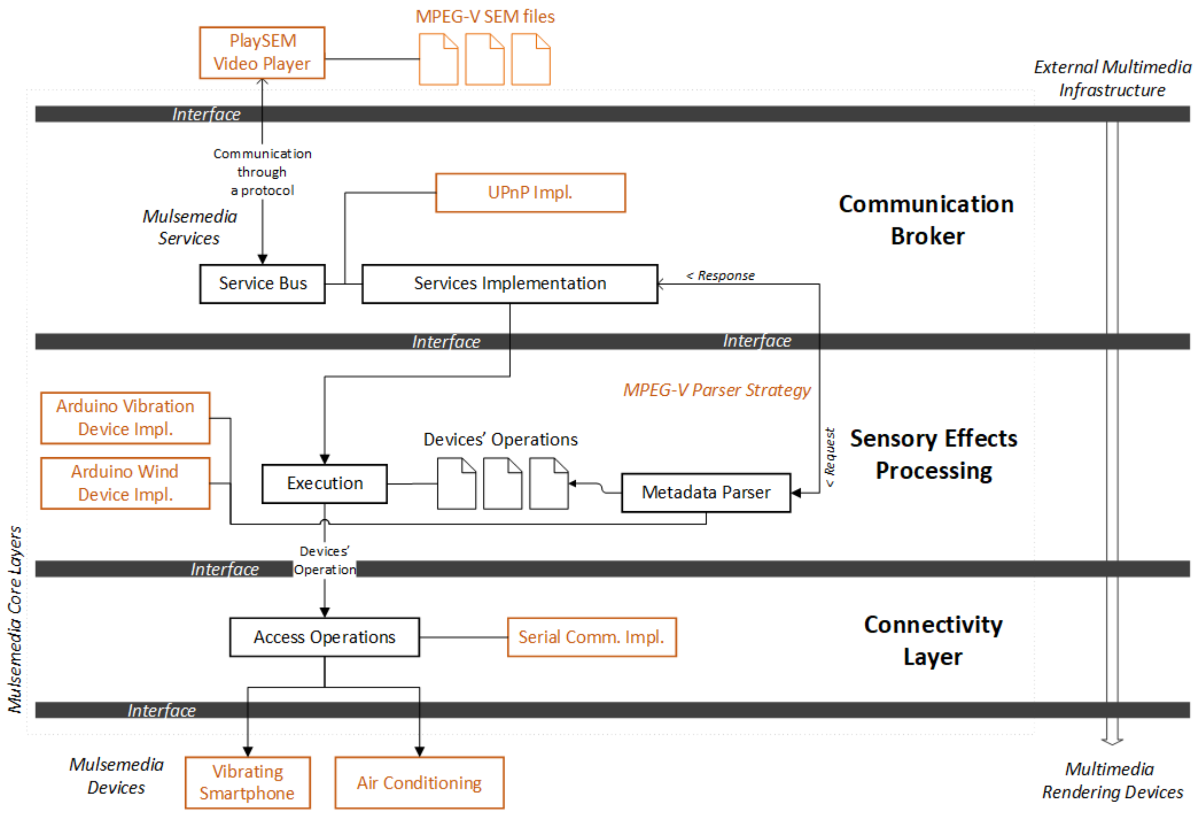


Source: Created by the author.

A.2 TV Broadcast Mulsemedia System

Figure 30 brings the TV Broadcast Mulsemedia System created by Jalal et al. (2018b). The environment included a TV, a desktop PC, three RGB Smart Philips LED light devices for lighting effect, an air conditioner for airflow, a smartphone for vibration effects, and the PlaySEM platform to play the videos and to render sensory effects described in MPEG-V. A caveat, however, is that they were hindered by a limitation in the first version of PlaySEM SER (SALEME; SANTOS, 2015), which did not support multiple connectivity protocols to integrate other devices. Thus, the authors used serial communication for the wind and vibrating device whilst the three RGB Smart Philips LED light devices were connected as part of the external multimedia infrastructure. They used a smartphone to record the screen of the TV, obtained the average colors of the video in real-time, and transmitted this information to the LED device through Wi-Fi.

Figure 30 – A TV Broadcast Mulsemedia System instantiated in the Conceptual architecture.



Source: Created by the author.

APPENDIX B – PlaySEM SER 2 Configuration File

This is a full example of a configuration file (`SERenderer.xml`) for the framework PlaySEM SER 2 where setup related to initialization, communication and connectivity protocols, temporal aspects, debug and simulation, and their properties, are created and described (Chapter 4, Section 4.5).

```xml
<?xml version="1.0" encoding="UTF-8"?>

<configuration>
   <!-- Main configuration -->
   <communicationServiceBroker>upnpService</communicationServiceBroker>
   <metadataParser>mpegvParser</metadataParser>
   <lightDevice>mockLight</lightDevice>
   <windDevice>mockWind</windDevice>
   <vibrationDevice>mockVibration</vibrationDevice>
   <scentDevice>mockScent</scentDevice>
   <debugMode>true</debugMode>

   <!-- Communication services -->
   <communicationServices>
      <communicationService>
         <id>coapService</id>
         <communicationServiceClass>br.ufes.inf.lprm.sensoryeffect.renderer.
            service.coap.SERendererCoapService
         </communicationServiceClass>
      </communicationService>
      <communicationService>
         <id>mqttService</id>
         <communicationServiceClass>br.ufes.inf.lprm.sensoryeffect.renderer.
            service.mqtt.SERendererMqttService
         </communicationServiceClass>
      </communicationService>
      <communicationService>
         <id>upnpService</id>
         <communicationServiceClass>br.ufes.inf.lprm.sensoryeffect.renderer.
            service.upnp.SERendererUPnPService
         </communicationServiceClass>
      </communicationService>
      <communicationService>
         <id>websocketService</id>
         <communicationServiceClass>br.ufes.inf.lprm.sensoryeffect.renderer.
```

```xml
            service.websocket.SERendererWebSocketService
         </communicationServiceClass>
      </communicationService>
   </communicationServices>

   <!-- Metadata parsers -->
   <metadataParsers>
      <metadataParser>
         <id>mpegvParser</id>
         <metadataParserClass>br.ufes.inf.lprm.sensoryeffect.renderer.
            metadata.parser.mpegv.MPEGVSEMParser
         </metadataParserClass>
      </metadataParser>
   </metadataParsers>

   <!-- Supported devices -->
   <devices>
      <!-- Mock devices - it prints the results on the console -->
      <device>
         <id>mockLight</id>
         <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.
            mock.MockLightDevice
         </deviceClass>
         <connectivityInterface>Console</connectivityInterface>
      </device>
      <device>
         <id>mockVibration</id>
         <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.
            device.mock.MockVibrationDevice
         </deviceClass>
         <connectivityInterface>Console</connectivityInterface>
      </device>
      <device>
         <id>mockWind</id>
         <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.
            mock.MockWindDevice
         </deviceClass>
         <connectivityInterface>Console</connectivityInterface>
         <properties>
            <delay>0</delay>
         </properties>
      </device>
      <device>
         <id>mockScent</id>
         <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.
            mock.MockScentDevice
         </deviceClass>
```

```
        <connectivityInterface>Console</connectivityInterface>
        <properties>
            <ScentSlot01>
                urn:mpeg:mpeg-v:01-SI-ScentCS-NS:rose
            </ScentSlot01>
            <ScentSlot02>
                urn:mpeg:mpeg-v:01-SI-ScentCS-NS:cinnamon
            </ScentSlot02>
            <ScentSlot03>
                urn:mpeg:mpeg-v:01-SI-ScentCS-NS:chocolate_dark
            </ScentSlot03>
            <ScentSlot04>
                urn:mpeg:mpeg-v:01-SI-ScentCS-NS:rubbish_acrid
            </ScentSlot04>
            <delay>0</delay>
        </properties>
    </device>

    <!-- Arduino Devices (Light, wind, vibration) -->
    <device>
        <id>arduinoLightDevice</id>
        <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.
            arduino.ArduinoLightDevice
        </deviceClass>
        <connectivityInterface>Serial</connectivityInterface>
    </device>
    <device>
        <id>arduinoVibrationDevice</id>
        <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.
            arduino.ArduinoVibrationDevice
        </deviceClass>
        <connectivityInterface>Serial</connectivityInterface>
    </device>
    <device>
        <id>arduinoWindDevice</id>
        <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.
            arduino.ArduinoWindDevice
        </deviceClass>
        <connectivityInterface>Serial</connectivityInterface>
        <properties>
            <delay>500</delay>
        </properties>
    </device>

    <!-- Vortex Scent Device -->
    <device>
        <id>vortexScentDevice</id>
```

```
    <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.
       vortex.VortexScentDevice
    </deviceClass>
    <connectivityInterface>SerialFTD2XX</connectivityInterface>
    <properties>
       <ScentSlot01>
          urn:mpeg:mpeg-v:01-SI-ScentCS-NS:burning_rubber
       </ScentSlot01>
       <ScentSlot02>
          urn:mpeg:mpeg-v:01-SI-ScentCS-NS:lavender
       </ScentSlot02>
       <ScentSlot03>
          urn:mpeg:mpeg-v:01-SI-ScentCS-NS:coffee_cream
       </ScentSlot03>
       <ScentSlot04>
          urn:mpeg:mpeg-v:01-SI-ScentCS-NS:rubbish_acrid
       </ScentSlot04>
       <delay>0</delay>
    </properties>
 </device>

 <!-- Exhalia Scent Device -->
 <device>
    <id>exhaliaScentDevice</id>
    <deviceClass>
       br.ufes.inf.lprm.sensoryeffect.renderer.device.exhalia.
          ExhaliaScentDevice
    </deviceClass>
    <connectivityInterface>Usb</connectivityInterface>
    <properties>
       <ScentSlot01>
          urn:mpeg:mpeg-v:01-SI-ScentCS-NS:burning_rubber
       </ScentSlot01>
       <ScentSlot02>
          urn:mpeg:mpeg-v:01-SI-ScentCS-NS:lavender
       </ScentSlot02>
       <ScentSlot03>
          urn:mpeg:mpeg-v:01-SI-ScentCS-NS:coffee_cream
       </ScentSlot03>
       <ScentSlot04>
          urn:mpeg:mpeg-v:01-SI-ScentCS-NS:rubbish_acrid
       </ScentSlot04>
       <delay>0</delay>
       <increasedIntensity>false</increasedIntensity>
    </properties>
 </device>
```

```
<!-- Bluno Scent Device -->
<device>
   <id>blunoScentDevice</id>
   <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.
      bluno.BlunoScentDevice
   </deviceClass>
   <connectivityInterface>WiFi</connectivityInterface>
   <properties>
      <delay>0</delay>
   </properties>
</device>

<!-- Android Devices (Light and vibration) -->
<device>
   <id>sedroidLightDevice</id>
   <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.
      sedroid.SedroidLightDevice
   </deviceClass>
   <connectivityInterface>Bluetooth</connectivityInterface>
</device>
<device>
   <id>sedroidVibrationDevice</id>
   <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.
      sedroid.SedroidVibrationDevice
   </deviceClass>
   <connectivityInterface>Bluetooth</connectivityInterface>
</device>

<!-- amBX Devices (Light, wind, vibration) -->
<device>
   <id>ambxLightDevice</id>
   <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.
      ambx.AmbxLightDevice
   </deviceClass>
   <connectivityInterface>UsbAmbx</connectivityInterface>
</device>
<device>
   <id>ambxVibrationDevice</id>
   <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.
      ambx.AmbxVibrationDevice
   </deviceClass>
   <connectivityInterface>UsbAmbx</connectivityInterface>
</device>
<device>
   <id>ambxWindDevice</id>
   <deviceClass>br.ufes.inf.lprm.sensoryeffect.renderer.device.
      ambx.AmbxWindDevice
```

```
        </deviceClass>
        <connectivityInterface>UsbAmbx</connectivityInterface>
     </device>
  </devices>

  <!-- Connectivity interfaces -->
  <connectivityInterfaces>
     <connectivityInterface>
        <id>Console</id>
        <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.
           renderer.connectivity.console.ConsoleOutput
        </connectivityInterfaceClass>
        <earlyConnection>true</earlyConnection>
     </connectivityInterface>
     <connectivityInterface>
        <id>Serial</id>
        <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.
           renderer.connectivity.serial.SerialOutput
        </connectivityInterfaceClass>
        <earlyConnection>true</earlyConnection>
        <properties>
           <serialPort>COM3</serialPort>
           <baudRate>9600</baudRate>
        </properties>
     </connectivityInterface>
     <connectivityInterface>
        <id>SerialFTD2XX</id>
        <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.
           renderer.connectivity.serial.FTD2XXOutput
        </connectivityInterfaceClass>
        <earlyConnection>true</earlyConnection>
        <properties>
           <device01-id>67330049</device01-id>
        </properties>
     </connectivityInterface>
     <connectivityInterface>
        <id>Bluetooth</id>
        <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.
           renderer.connectivity.bluetooth.BluetoothOutput
        </connectivityInterfaceClass>
        <earlyConnection>true</earlyConnection>
        <properties>
           <number-of-devices>1</number-of-devices>
           <device01-connection-url>
              btspp://B4994C5024E1:5;authenticate=false;encrypt=false;
              master=false
           </device01-connection-url>
```

```
        <device02-connection-url>
           btspp://4088058F7E2E:5;authenticate=false;encrypt=false;
           master=false
        </device02-connection-url>
        <device03-connection-url>
           btspp://A816D0063584:5;authenticate=false;encrypt=false;
           master=false
        </device03-connection-url>
        <device04-connection-url>
           btspp://A816D006351E:5;authenticate=false;encrypt=false;
           master=false
        </device04-connection-url>
     </properties>
  </connectivityInterface>
  <connectivityInterface>
     <id>WiFi</id>
     <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.
        renderer.connectivity.wifi.WiFiOutput
     </connectivityInterfaceClass>
     <earlyConnection>true</earlyConnection>
     <properties>
        <number-of-devices>1</number-of-devices>
        <device01-address>192.168.0.101:8080</device01-address>
        <device02-address>90.0.0.107:8080</device02-address>
        <device03-address>90.0.0.102:8080</device03-address>
     </properties>
  </connectivityInterface>
  <connectivityInterface>
     <id>Usb</id>
     <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.
        renderer.connectivity.usb.UsbOutput
     </connectivityInterfaceClass>
     <earlyConnection>true</earlyConnection>
     <properties>
        <vendorId>0x1781</vendorId>
        <productId>0x07D0</productId>
        <usbInterface>0</usbInterface>
        <inEndPoint>81</inEndPoint>
        <outEndPoint>1</outEndPoint>
        <timeout>5000</timeout>
     </properties>
  </connectivityInterface>
  <connectivityInterface>
     <id>UsbAmbx</id>
     <connectivityInterfaceClass>br.ufes.inf.lprm.sensoryeffect.
        renderer.connectivity.usb.UsbOutput
     </connectivityInterfaceClass>
```

```
        <earlyConnection>true</earlyConnection>
        <properties>
           <vendorId>0x0471</vendorId>
           <productId>0x083f</productId>
           <usbInterface>0</usbInterface>
           <inEndPoint>81</inEndPoint>
           <outEndPoint>2</outEndPoint>
           <timeout>5000</timeout>
        </properties>
     </connectivityInterface>
  </connectivityInterfaces>

</configuration>
```

APPENDIX C – QoE Experiment Material

This chapter presents the introduction of the subjective quality assessment (Section C.1), the consent form (Section C.2), the questionnaire on demographic information (Section C.3), and the tables with statistical descriptions (MOS) and calculations (independent samples t-test and ANOVA) for the user-perceived QoE (Section C.4) from Chapter 5 (Section 5.3).

C.1 Introduction

About the experiment.

- Objective

Assess the quality of a sequence of audiovisual content enriched with external lighting, wind, and vibration effects using different media players.

- Procedure

You will to watch 5 short videos with sensory effects synchronized with the scenes presented in each of them. At the end of each video presentation, you will be asked to complete a questionnaire indicating your perceptions over what you have just experienced. Please, be as honest as possible when answering the questions.

- Please, request the researcher a brief demonstration of the sensory effect devices working with an audiovisual demo content before going ahead.

- Does the demonstration suffice to understand what type of content will be assessed?

( ) Yes.

( ) No. Request a new demonstration.

C.2 Informed Consent Form

Dear Participant,

You have been invited to take part in an experiment about users' quality of experience over audiovisual content enhanced with multiple sensory effects. The experiment aims to collect information about your perception of audiovisual content enriched with sensory effects of lighting, wind, and vibration, using different systems.

First, you will answer a questionnaire on demographic information. Then, a set of 5 short videos with sensory effects will be presented to you. At the end of each session, you will be asked to complete another questionnaire to report your experience. Finally, the researcher will make a brief interview with you. The experiment time lasts about 20-30 minutes (10 minutes for watching the videos and the remainder is the time you take to answer the questionnaires).

Notice that the information obtained is confidential and will be used exclusively in this research. Your name will not be disclosed under any circumstances. Your answers will be stored for a period of 2 (two) years and then it will be discarded.

Your participation is voluntary, that is, it is unpaid. If you are uncomfortable to participate or experience any discomfort during the experiment, you can abort your participation immediately.

- Do you agree to participate and grant the right to use data collected in this survey?
( ) Yes, I agree.
( ) No, I do not agree.

C.3 Questionnaire on Demographic Information

Information about you.

- Your gender:
( ) Female.
( ) Male.
( ) Prefer not to say.

- Your age:
( ) 16-25.
( ) 26-35.
( ) 36-45.
( ) 46-55.
( ) 56-65.
( ) 65+.

C.4 Statistical Tables

The tables in this section present statistical descriptions and calculations for the user-perceived QoE from the subjective quality assessment presented in Chapter 5 (Section 5.3).

Table 7 presents the MOS by system for each dependent variable with their respective standard deviation and standard error of the mean. The acronyms presented in this table have the following meanings:
M: Mean.
SD: Standard deviation.
SE: Standard error of the mean.

Table 8 provides details for the independent samples t-test for equality of means in order to determine whether there is statistical evidence that the associated population means are significantly different. The acronyms presented in this table have the following meanings:
df: The degrees of freedom.
t: It indicates the obtained value of the t-statistic (obtained t-value).
Sig. (2-tailed): The p-value (statistical significance) corresponding to the given test statistic and degrees of freedom.
M Diff: The difference between the sample means.
SE Diff: The standard error for the difference between the sample means. It measures the variability of the mean difference.
95% CI Diff: The confidence interval of the difference between the sample means. It measures the variability of the mean difference.

Table 9 presents the MOS by system and video with their respective standard deviation. The acronyms presented in this table have the following meanings:
M: Mean.
SD: Standard deviation.

Table 10 reports ANOVA tests of between-subjects effects in order to test significant changes in the dependent variables when changing the videos for each system. The acronyms presented in this table have the following meanings:
df: Degree of freedom.
F: It indicates the obtained value of the f-statistic (obtained f-value).
p: The p-value (statistical significance) corresponding to the given test statistic and degrees of freedom.

Table 7 – Mean opinion score by system (*n*=100).

	PlaySEM	SEMP
	M (SD, SE)	*M (SD, SE)*
Video Visual Quality	4.49 (0.893, 0.089)	4.42 (0.855, 0.085)
Video Audio Quality	4.41 (0.944, 0.094)	4.24 (0.996, 0.100)
Video Audio Synchronization	4.62 (0.663, 0.066)	4.61 (0.601, 0.060)
Video Enjoyment	4.06 (0.973, 0.097)	4.07 (1.200, 0.120)
Wind Reality Enhancement	4.03 (0.979, 0.098)	3.95 (1.038, 0.104)
Wind Synchronization	3.81 (1.042, 0.104)	3.77 (1.072, 0.107)
Wind Distraction	1.97 (0.979, 0.098)	1.93 (1.183, 0.118)
Wind Annoyance	1.68 (0.875, 0.087)	1.62 (1.023, 0.102)
Wind Enjoyment	4.02 (0.943, 0.094)	4.04 (1.072, 0.107)
Vibration Reality Enhancement	4.23 (0.908, 0.091)	4.29 (0.988, 0.099)
Vibration Synchronization	3.68 (1.081, 0.108)	3.89 (1.238, 0.124)
Vibration Distraction	1.84 (0.918, 0.092)	1.60 (0.995, 0.099)
Vibration Annoyance	1.64 (0.785, 0.079)	1.42 (0.843, 0.084)
Vibration Enjoyment	4.12 (0.891, 0.089)	4.30 (1.020, 0.102)
Light Reality Enhancement	3.99 (1.078, 0.108)	3.76 (1.016, 0.102)
Light Synchronization	3.91 (1.006, 0.101)	4.04 (0.994, 0.099)
Light Distraction	1.85 (0.968, 0.097)	1.98 (1.025, 0.102)
Light Annoyance	1.64 (0.785, 0.079)	1.62 (0.850, 0.085)
Light Enjoyment	4.07 (0.987, 0.099)	3.86 (0.865, 0.086)
General QoE	4.09 (0.753, 0.075)	4.06 (0.814, 0.081)

Source: Created by the author.

Table 8 – Independent samples t-test for equality of means.

	df	*t*	*Sig. (2-tailed)*	*M Diff*	*SE Diff*	*95% CI Diff*
Video Visual Quality	38	0.566	0.572	0.07	0.124	[-0.174, 0.314]
Video Audio Quality	38	1.239	0.217	0.17	0.137	[-0.101, 0.441]
Video Audio Synchronization	38	0.112	0.911	0.01	0.090	[-0.167, 0.187]
Video Enjoyment	38	-0.065	0.948	-0.01	0.154	[-0.315, 0.295]
Wind Reality Enhancement	38	0.561	0.576	0.08	0.143	[-0.201, 0.361]
Wind Synchronization	38	0.268	0.789	0.04	0.149	[-0.255, 0.335]
Wind Distraction	38	0.261	0.795	0.04	0.154	[-0.263, 0.343]
Wind Annoyance	38	0.446	0.656	0.06	0.135	[-0.205, 0.325]
Wind Enjoyment	38	-0.140	0.889	-0.02	0.143	[-0.302, 0.262]
Vibration Reality Enhancement	38	-0.447	0.655	-0.06	0.134	[-0.325, 0.205]
Vibration Synchronization	38	-1.277	0.203	-0.21	0.164	[-0.534, 0.114]
Vibration Distraction	38	1.773	0.078	0.24	0.135	[-0.027, 0.507]
Vibration Annoyance	38	1.910	0.058	0.22	0.115	[-0.007, 0.447]
Vibration Enjoyment	38	-1.329	0.185	-0.18	0.135	[-0.447, 0.087]
Light Reality Enhancement	38	1.553	0.122	0.23	0.148	[-0.062, 0.522]
Light Synchronization	38	-0.919	0.359	-0.13	0.141	[-0.409, 0.149]
Light Distraction	38	-0.922	0.358	-0.13	0.141	[-0.408, 0.148]
Light Annoyance	38	0.173	0.863	0.02	0.116	[-0.208, 0.248]
Light Enjoyment	38	1.600	0.111	0.21	0.131	[-0.049, 0.469]
General QoE	38	0.270	0.787	0.03	0.111	[-0.189, 0.249]

Source: Created by the author.

Table 9 – Mean opinion score by system and video (n=20).

	PlaySEM					SEMP				
	Formula 1	Babylon	Earth	Rambo	Wo ist Klaus?	Formula 1	Babylon	Earth	Rambo	Wo ist Klaus?
	M (SD)	*M (SD)*	*M (SD)*	*M (SD)*	*M (SD)*	*M (SD)*	*M (SD)*	*M (SD)*	*M (SD)*	*M (SD)*
Video Visual Quality	4.60 (0.995)	4.75 (0.444)	4.75 (0.444)	4.85 (0.366)	3.50 (1.147)	4.85 (0.366)	4.80 (0.410)	4.40 (0.821)	4.65 (0.489)	3.40 (1.046)
Video Audio Quality	4.40 (0.940)	4.65 (0.745)	4.70 (0.733)	4.65 (0.745)	3.65 (1.137)	4.65 (0.489)	4.65 (0.587)	4.50 (0.607)	4.45 (0.826)	2.95 (1.146)
Video Audio Synchronization	4.55 (0.759)	4.60 (0.754)	4.90 (0.308)	4.80 (0.410)	4.25 (0.786)	4.65 (0.587)	4.85 (0.366)	4.70 (0.470)	4.75 (0.444)	4.10 (0.788)
Video Enjoyment	4.00 (0.858)	4.45 (0.686)	4.65 (0.587)	4.30 (0.733)	2.90 (0.912)	3.90 (1.294)	4.30 (1.129)	4.60 (0.681)	4.20 (1.152)	3.35 (1.348)
Wind Reality Enhancement	4.20 (0.951)	4.40 (0.754)	4.20 (0.768)	3.85 (1.089)	3.50 (1.100)	4.05 (1.234)	4.40 (0.681)	4.25 (0.910)	3.40 (1.095)	3.65 (0.933)
Wind Synchronization	3.70 (1.129)	3.90 (1.210)	4.05 (0.945)	3.65 (0.933)	3.75 (1.020)	3.60 (1.231)	4.20 (0.768)	4.25 (0.851)	3.30 (1.031)	3.50 (1.147)
Wind Distraction	2.10 (1.071)	2.05 (1.276)	1.65 (0.745)	2.00 (0.918)	2.05 (0.826)	1.85 (1.226)	1.85 (1.040)	1.65 (1.040)	2.20 (1.399)	2.10 (1.210)
Wind Annoyance	1.80 (0.951)	1.80 (1.005)	1.50 (0.513)	1.60 (0.821)	1.70 (1.031)	1.65 (1.137)	1.45 (0.826)	1.70 (1.261)	1.75 (1.070)	1.55 (0.826)
Wind Enjoyment	4.10 (0.968)	4.25 (0.967)	4.35 (0.587)	3.85 (0.933)	3.55 (1.050)	4.05 (1.234)	4.45 (0.686)	4.30 (0.979)	3.65 (1.182)	3.75 (1.070)
Vibration Reality Enhancement	4.15 (0.875)	4.60 (0.754)	3.95 (0.887)	4.30 (0.923)	4.15 (1.040)	4.00 (1.214)	4.55 (0.945)	4.25 (0.786)	4.20 (1.196)	4.45 (0.686)
Vibration Synchronization	3.40 (1.142)	3.85 (1.137)	3.75 (1.020)	3.15 (1.089)	4.25 (0.716)	3.35 (1.309)	4.50 (0.761)	4.10 (1.071)	3.25 (1.446)	4.25 (1.070)
Vibration Distraction	1.90 (0.852)	1.90 (1.165)	1.60 (0.754)	1.90 (0.968)	1.90 (0.852)	1.70 (1.129)	1.65 (1.137)	1.55 (0.826)	1.40 (0.821)	1.70 (1.081)
Vibration Annoyance	1.60 (0.754)	1.75 (0.967)	1.45 (0.510)	1.65 (0.813)	1.75 (0.851)	1.40 (0.821)	1.35 (0.671)	1.40 (0.681)	1.50 (1.147)	1.45 (0.887)
Vibration Enjoyment	4.10 (0.912)	4.25 (0.967)	4.05 (0.826)	4.15 (0.933)	4.05 (0.887)	4.35 (0.875)	4.50 (0.946)	4.35 (0.875)	3.95 (1.395)	4.35 (0.933)
Light Reality Enhancement	3.95 (1.099)	3.95 (1.146)	4.30 (0.801)	4.20 (1.196)	3.55 (1.050)	3.45 (1.191)	3.85 (0.933)	4.00 (0.973)	3.95 (0.999)	3.55 (0.945)
Light Synchronization	4.00 (1.026)	3.85 (0.875)	3.90 (1.021)	4.05 (1.276)	3.75 (0.851)	3.85 (1.137)	3.85 (1.089)	4.45 (0.605)	4.30 (0.865)	3.75 (1.070)
Light Distraction	2.00 (1.026)	1.60 (0.821)	1.80 (0.894)	1.95 (1.234)	1.90 (0.852)	1.65 (0.875)	2.10 (1.119)	1.90 (1.021)	2.50 (1.100)	1.75 (0.851)
Light Annoyance	1.55 (0.759)	1.70 (0.923)	1.55 (0.605)	1.65 (0.933)	1.75 (0.716)	1.40 (0.754)	1.75 (0.910)	1.55 (0.759)	2.00 (1.026)	1.40 (0.681)
Light Enjoyment	4.00 (0.973)	4.05 (0.999)	4.45 (0.686)	4.25 (1.118)	3.60 (0.995)	3.75 (0.851)	3.95 (0.759)	4.10 (0.968)	3.90 (1.021)	3.60 (0.681)
General QoE	3.90 (0.447)	4.20 (0.951)	4.30 (0.657)	4.30 (0.733)	3.75 (0.786)	3.90 (1.021)	4.30 (0.801)	4.30 (0.657)	4.00 (0.918)	3.80 (0.523)

Source: Created by the author.

Table 10 – ANOVA for between-subjects effects (interaction between system and video).

	df	*F*	*p*
Video Visual Quality	4	1.035	0.390
Video Audio Quality	4	1.802	0.130
Video Audio Synchronization	4	0.966	0.427
Video Enjoyment	4	0.652	0.626
Wind Reality Enhancement	4	0.583	0.675
Wind Synchronization	4	0.737	0.568
Wind Distraction	4	0.287	0.886
Wind Annoyance	4	0.570	0.685
Wind Enjoyment	4	0.318	0.866
Vibration Reality Enhancement	4	0.551	0.699
Vibration Synchronization	4	0.701	0.592
Vibration Distraction	4	0.284	0.888
Vibration Annoyance	4	0.267	0.899
Vibration Enjoyment	4	0.489	0.743
Light Reality Enhancement	4	0.342	0.849
Light Synchronization	4	0.761	0.552
Light Distraction	4	1.594	0.178
Light Annoyance	4	1.005	0.406
Light Enjoyment	4	0.289	0.885
General QoE	4	0.414	0.799

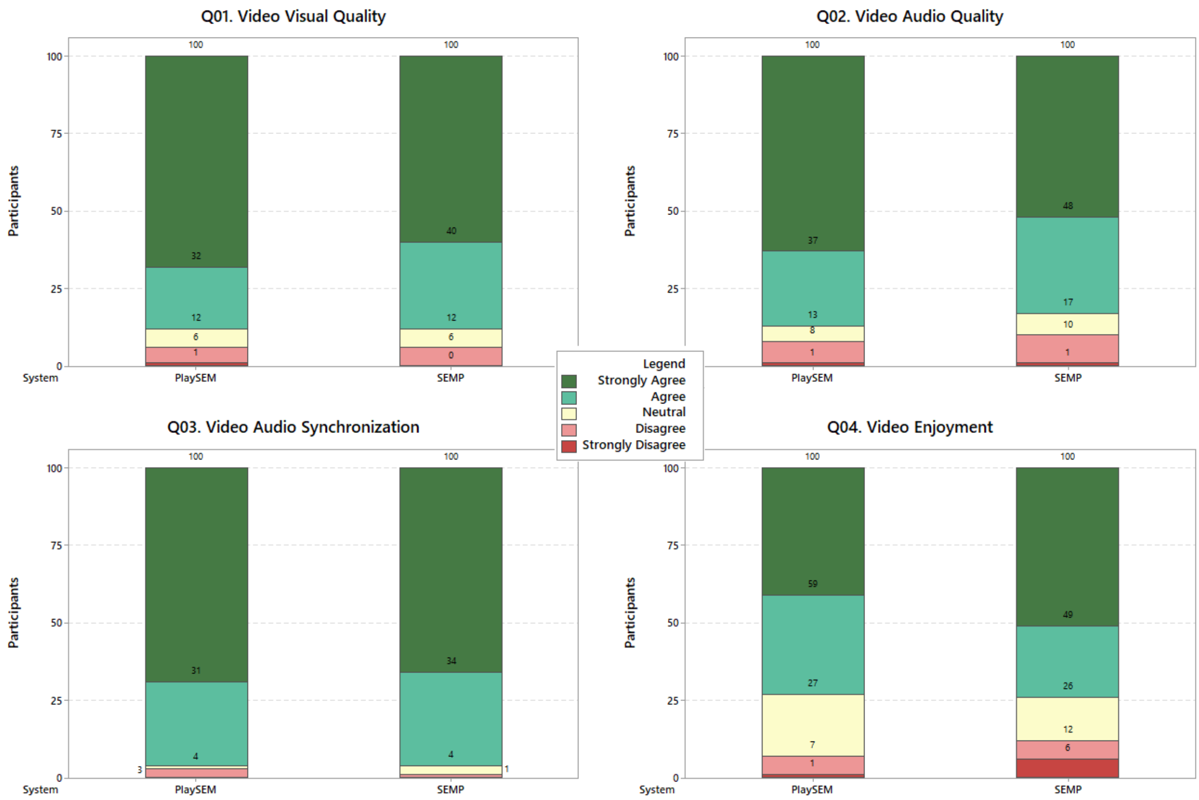
Source: Created by the author.

C.5 Complementary charts

This section presents complementary figures for the results of the subjective quality assessment described in Chapter 5 (Section 5.3).

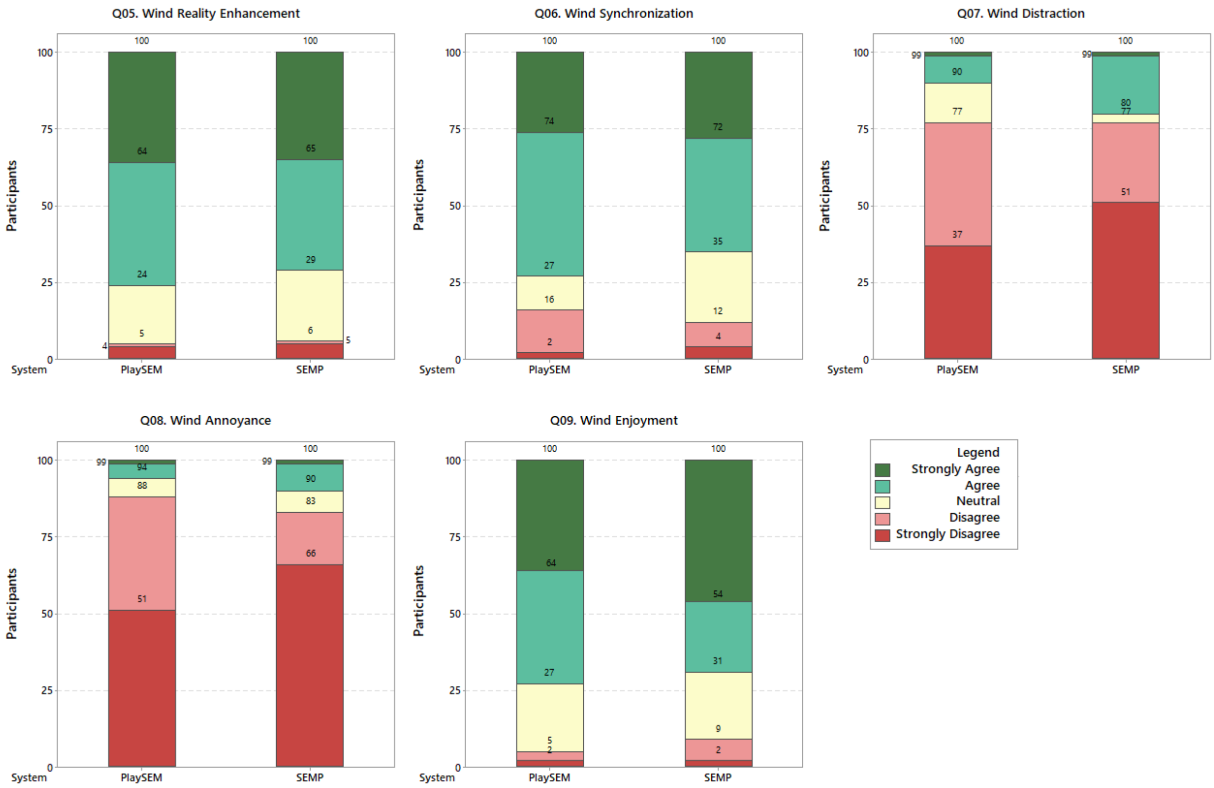
Figure 31, Figure 32, Figure 33, and Figure 34 show in stacked bar charts the number of participants and their evaluations for the questions Q01..Q04 (related to audiovisual experience and content), Q05..Q09 (related to wind effects), Q10..Q14 (related to vibration effects), and Q15..Q19 (related to external lighting effects), respectively.

Figure 31 – Stacked bar chart to compare the values of Q01..Q04—related to audiovisual experience and content.



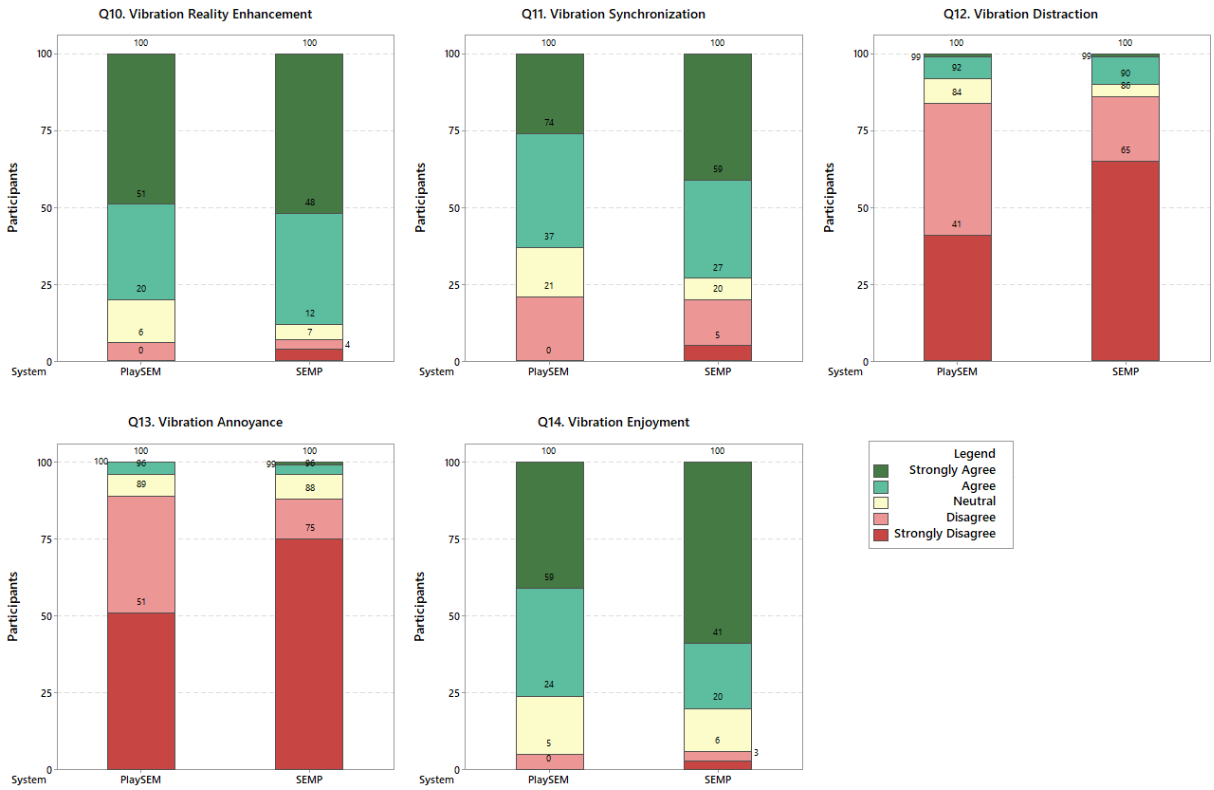
Source: Created by the author.

Figure 32 – Stacked bar chart to compare the values of Q05..Q09—wind effects.



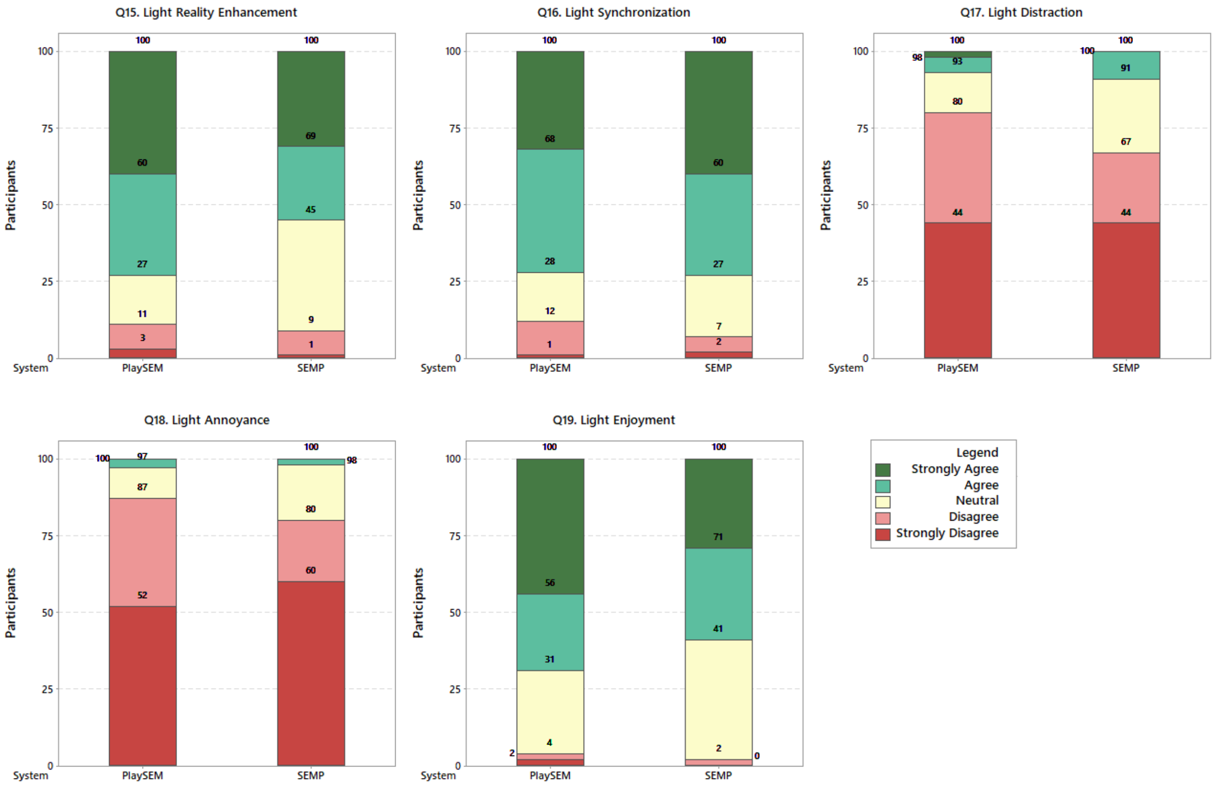
Source: Created by the author.

Figure 33 – Stacked bar chart to compare the values of Q10..Q14—vibration effects.



Source: Created by the author.

Figure 34 – Stacked bar chart to compare the values of Q15..Q19—external lighting effects.



Source: Created by the author.