PhD THESIS

Social Emotions in Multi-Agent Systems

Author: Jaime Andres Rincon Arango

Advisors: Dr. Carlos Carrascosa Casamayor
Dr. Vicente Julián Inglada

Departamento de Sistemas Informáticos y Computación,
Universidad Politècnica de València,
Valencia, Spain

December 2017

PhD THESIS

Title	Social Emotions in Multi-Agent Systems
Author:	JaimeAndres Rincon Arango
Advisors:	Dr.Carlos Carrascosa Casamayor Dr. Vicente Julián Inglada
Reviewers:	Dr. Davide Carneiro Dr. Florentino Fernández Riverola Dr. Alessandro Ricci
Examination Board:	Dra. Adriana Susana Giret Boggino Dr. Florentino Fernández Riverola Dr. Michael Ignaz Schumacher

Date of the defense: 17 January 2018

La vida es como una bicicleta.
Para mantener el equilibrio tienes que seguir adelante.

-Albert Einstein-

Agradecimientos

This is it. This is my PhD thesis and I finally finished it. In this book you can find the hard work of four years. This work is the result of the effort of several people who not only intervened directly in the experimental process, but also supported me and participated decisively in my professional training and orientation towards research.

I have to thank many people, family and colleagues with whom I work in different research centers, and that is why in this section I would like to give my thanks in my mother tongue.

En primer lugar quisiera agradecer al Dr. Carlos Carrascosa Casamayor y al Dr. Vicente Javier Julian Inglada, directores de esta tesis, por su dedicación en el día a día en la realización de la tesis, por introducirme en el mundillo de la investigación y la docencia. También por los buenos consejos y transmitirme sus conocimientos que sin lugar a duda me ayudaron a llevar a buen puerto este trabajo. Gracias por los cafés que nos tomamos en esos ratos de esparcimiento, así mismo, me disculpo por no compartir esos momentos a la hora de comer, ya que como saben no me gusta hablar mientras como. Carlos, gracias por los caramelos que te quitaba a manos llenas. Ustedes me han enseñado mucho y me han convertido en una mejor persona.

Al Dr. Paulo Novais, mi supervisor durante mi estancia en la Universidade do Minho, por recibirme con sus brazos abiertos y por prestarme su ayuda incondicional.

A mi queridos compañeros de laboratorio Maria, Elena, Jaume, Bexy, Angelo y Mohannad, por su compañía, sus risas y por volcarse conmigo cada vez que me surgia una duda. Muchos de ustedes convivieron conmigo en el laboratorio, algunas veces apretujados por el espacio, pero bien que lo pasamos. Sin ustedes nada de esto habría sido lo mismo.

A los compañeros del GTI-IA, tanto becarios como profesores. Al Departamento de Sistemas Informáticos y Computación, en especial a las secretarias por ser tan eficientes.

Las gracias más grandes a mis padres, Teresa Arango Gomez y Jaime de Jesus Rincon Quintero. Ustedes me han enseñado todo lo que soy. Esta tesis fue posible no solo gracias a mi trabajo y sacrificio, sino también al de ustedes. A mi hermana Monica y a mi sobrino Itzael, que espero algún día siga los mismos pasos de su tío. Y al resto de mi familia que de una u otra forma me apoyaron.

A Sol, mi querido amor y compañera durante este largo camino. Por tu comprensión, amor y paciencia (¡mucha!). Porque me ayudaste a seguir adelante en esos momentos en los que me sentía sin animo y desorientado.

Resumen

A lo largo de los últimos años, los sistemas multi-agente (SMA) han demostrado ser un paradigma potente y versátil, con un gran potencial a la hora de resolver problemas complejos en entornos dinámicos y distribuidos. Este potencial no se debe principalmente a sus características individuales (como son su autonomía, su capacidad de percepción, reacción y de razonamiento), sino que también a la capacidad de comunicación y cooperación a la hora de conseguir un objetivo. De hecho, su capacidad social es la que más llama la atención, es este comportamiento social el que dota de potencial a los sistemas multi-agente. Estas características han hecho de los SMA, la herramienta de inteligencia artificial (IA) más utilizada para el diseño de entornos virtuales inteligentes (IVE), los cuales son herramientas de simulación compleja basadas en agentes. Sin embargo, los IVE incorporan restricciones físicas (como gravedad, fuerzas, rozamientos, etc.), así como una representación 3D de lo que se quiere simular. Así mismo, estas herramientas no son sólo utilizadas para la realización de simulaciones. Con la aparición de nuevas aplicaciones como *Internet of Things (IoT)*, *Ambient Intelligence (AmI)*, robot asistentes, entre otras, las cuales están en contacto directo con el ser humano. Este contacto plantea nuevos retos a la hora de interactuar con estas aplicaciones. Una nueva forma de interacción que ha despertado un especial interés, es el que se relaciona con la detección y/o simulación de estados emocionales. Esto ha permitido que estas aplicaciones no sólo puedan detectar nuestros estados emocionales, sino que puedan simular y expresar sus propias emociones mejorando así la experiencia del usuario con dichas aplicaciones. Con el fin de mejorar la experiencia humano-máquina, esta tesis plantea como objetivo principal la creación de modelos emocionales sociales, los cuales podrán ser utilizados en aplicaciones

MAS permitiendo a los agentes interpretar y/o emular diferentes estados emocionales y, además, emular fenómenos de contagio emocional. Estos modelos permitirán realizar simulaciones complejas basadas en emociones y aplicaciones más realistas en dominios como IoT, AIm, SH.

Resum

Al llarg dels últims anys, els sistemes multi-agent (SMA) han demostrat ser un paradigma potent i versàtil, amb un gran potencial a l'hora de resoldre problemes complexos en entorns dinàmics i distribuïts. Aquest potencial no es deu principalment a les seues característiques individuals (com són la seua autonomia, la seua capacitat de percepció, reacció i de raonament), sinó que també a la capacitat de comunicació i cooperació a l'hora d'aconseguir un objectiu. De fet, la seua capacitat social és la que més crida l'atenció, és aquest comportament social el que dota de potencial als sistemes multi-agent. Aquestes característiques han fet dels SMA, l'eina d'intel·ligència artificial (IA) més utilitzada per al disseny d'entorns virtuals intel·ligents (IVE), els quals són eines de simulació complexa basades en agents. No obstant això, els IVE incorporen restriccions físiques (com gravetat, forces, fregaments, etc.), així com una representació 3D del que es vol simular. Així mateix, aquestes eines no són només utilitzades per a la realització de simulacions. Amb l'aparició de noves aplicacions com *Internet of Things (IOT)*, *Ambient Intelligence (AmI)*, robot assistents, entre altres, les quals estan en contacte directe amb l'ésser humà. Aquest contacte planteja nous reptes a l'hora d'interactuar amb aquestes aplicacions. Una nova forma d'interacció que ha despertat un especial interès, és el que es relaciona amb la detecció i/o simulació d'estats emocionals. Això ha permès que aquestes aplicacions no només puguen detectar els nostres estats emocionals, sinó que puguen simular i expressar les seues pròpies emocions millorant així l'experiència de l'usuari amb aquestes aplicacions. Per tal de millorar l'experiència humà-màquina, aquesta tesi planteja com a objectiu principal la creació de models emocionals socials, els quals podran ser utilitzats en aplicacions MAS permetent als agents interpretar i/o emular diferents estats emocionals i, a més, emular fenòmens

de contagi emocional. Aquests models permetran realitzar simulacions complexes basades en emocions i aplicacions més realistes en dominis com IoT, AIM, SH.

Summary

Over the past few years, multi-agent systems (SMA) have proven to be a powerful and versatile paradigm, with great potential for solving complex problems in dynamic and distributed environments. This potential is not primarily due to their individual characteristics (such as their autonomy, their capacity for perception, reaction and reasoning), but also the ability to communicate and cooperate in achieving a goal. In fact, its social capacity is the one that draws the most attention, it is this social behavior that gives potential to multi-agent systems. These characteristics have made the SMA, the artificial intelligence (AI) tool most used for the design of intelligent virtual environments (IVE), which are complex agent-based simulation tools. However, IVE incorporates physical constraints (such as gravity, forces, friction, etc.), as well as a 3D representation of what you want to simulate. Also, these tools are not only used for simulations. With the emergence of new applications such as *Internet of Things (IoT)*, *Ambient Intelligence (AmI)*, robot assistants, among others, which are in direct contact with humans. This contact poses new challenges when it comes to interacting with these applications. A new form of interaction that has aroused a special interest is that which is related to the detection and / or simulation of emotional states. This has allowed these applications not only to detect our emotional states, but also to simulate and express their own emotions, thus improving the user experience with those applications. In order to improve the human-machine experience, this thesis aims to create social emotional models, which can be used in MAS applications, allowing agents to interpret and / or emulate different emotional states, and emulate phenomena of emotional contagion. These models will allow complex simulations based on emotions and more realistic applications in domains like IoT, AIm, SH.

Contents

Part I

Introduction and Objectives

CHAPTER

1

Introduction

1.1 Motivation

Over the last few years, the role of emotions in human behaviour has been under discussion in the scientific community and has become relevant to computer science. Research in psychology has established that emotions play a key role during decision-making processes [69][16]. In fact, in a previous study was found that patients with amygdala damage, which is involved in the processing of emotions, showed a decision-making impairment [16]. Concerning the field of robotics and artificial intelligence it is expected that detection of emotions is incorporated into the reasoning process of intelligent entities.

Emotions are defined as psychophysiological reactions representing modes of adaptation to certain stimuli [88]. The experience of an emotion occurs when individuals assess a specific situation based on knowledge of their environment. Psychologically, emotions alter attention by modifying behavior. Physiologically, emotions quickly organize the responses of different biological systems such as facial expressions or the endocrine system to generate the most effective behavior.

In order to understand what emotions are, researchers have tried to identify and classify different types of emotions. From a psychological perspective, the first classification of emotions established six basic emotions [77]: sadness, happiness, surprise, disgust, fear and anger. Over time, the list was expanded and as a result seven categories were introduced: primary or basic, secondary, positive, negative, ambiguous, static and social[58]. Alternative, the wheel of emotions is considered as a reference system for the classification of emotions. This model suggests that emotions can be mixed together in a variety of ways, coming closer to what happens when an emotion occurs [82].

In the field of computer science, the impact of emotions on cognitive processes and the establishment of a more natural interaction between humans and machines have laid the foundation for the development of a new branch called affective computation. One of the challenges of affective computing has been to design computational models that incorporate the concept of emotion for a more realistic recognition. Detecting emotional information begins with the use of passive sensors to capture essential data about human behavior (i.e. facial expression or voice trough videocamera or microphone). However, recognizing emotional information requires the extraction of significant patterns from captured data by using techniques of signal and image processing. Also, to get an automatic recognition is necessary to use artificial intelligence (AI) techniques. In this sense, machine learning techniques have been used to learn and recognize emotions. Similarly, they can be used to recognize emotional states associated to physiological signals (i.e. skin resistance, heart rate or body temperature) allowing to create wearable devices capable of identify human emotions.

The detection and recognition of emotions have become a fundamental part for the design of Internet of Things (IoT), Ambient Intelligent (AIm) and Smart Home (SH) applications. These applications seek to simulate human behaviors by using human-computer interfaces in order to make the user experience more enjoyable and realistic. In most developed applications emotions are typically modeled as static information. This type of modelling can influence on the decision-making of an individual entity without taking into account possible interactions with other entities or emotional contagion among them. For instance, contagion can be used by an individual or a group of individuals to manipulate emotions resulting on the intentional affective influence by a leader in a group. In this sense, multi-agent systems (MAS) have been proposed as the ideal tool to create applications that try to simulate emotions and emotional contagion between intelligent entities. Many developers have tried to explain the social function of emotions by using MAS to facilitate the interaction between computer systems and users.

MAS is a paradigm of software engineering for the design and development of complex software systems. In these systems, autonomous software entities (agents) can solve problems that are beyond its capabilities or knowledge by interacting with each other using high-level terms, protocols, and languages[177, 120, 89]. Agents can behave like a container incorporating AI techniques (i.e. machine learning, artificial vision, speech recognition, etc.) that allow them to recognize, learn and classify patterns. Similarly, just like humans, MAS have the ability to create societies of humans and agents living in a common environment thus producing a human-agent interaction. Members of these societies can cooperate with each other and solve complex problems. Emotions are an important part in the social interaction of people [85] and therefore, in the human-agent interaction, agents need to detect and simulate human emotions to make the interaction more fluid and realistic. This way, humans and agents communicate by a human emotion language [84]. It is important that all humans and agents express their emotions in the same scale of values to calculate the social emotion and the emotional spread.

Nevertheless, from our point of view it is necessary to deepen the use of emotional models and personality models into MAS to help agents to understand, classify and simulate human emotions. In computer science, we can found different models based on psychological theories. Most well-known emotional models and personality models are the PAD model [99], the Circumplex model [142] and the BIG-FIVE or OCEAN model [122]. Existing emotional models are thought to detect and/or simulate human emotions in one entity without taking into account the interaction with other entities and the development of social emotion. Therefore, it is necessary to introduce new models to obtain the social emotion and explain how the emotion propagates within a group.

In this thesis, new social emotional models are proposed in order to facilitate the decision-making of a group of individuals. These new models can be cataloged in two categories: the static model and the dynamic model. In each model, the PAD model is used and the emotion is represented as a three-dimensional vector (3D vector). In the static model, the social emotion is calculated using the individual emotion of each entity (humans and agents). At the same time, the social emotion is used to calculate the emotional distance and to know the emotional separation between many groups. In the dynamic model, the flow of emotions within a group is determined by using

Newton's second law, and it explains how emotions move and propagate through individuals.

The integration of individual emotional models, personality model and social models allow to design more realistic simulations of human-computer interfaces. However, to perform more complex simulations where emotions play an important role it is necessary to develop tools to create environments where humans and agents can coexist. These environments are called Intelligent Virtual Environments (IVEs) [70] and they make possible to simulate real environments inhabited by autonomous intelligent entities. These types of environments are most commonly used in the design of multi-user games such as *World of Warcraft* and immerse social networks such as *Second Life*. At the same time, IVEs can be used in serious games such as [19], providing a tool to simulate training environment in military situations or improve the design of smart homes. Several approaches have been proposed for the use of MAS as a paradigm for IVEs modelling. However, there are still some open issues regarding the use of MAS since they have low generality and reuse. Furthermore, MAS lack of strong support for managing open and dynamic environments where objects are dynamically created and destroyed. In this sense, we propose the JaCalIVE framework which is based on the MAM5 meta-model [14] which describes a method to design IVEs.

Hence, in this thesis we pursue computational solutions for implementing emotional models that represent and play different emotions in AmI, IoT or SH applications. In addition, we try to implement these computational solutions in the design and simulation of emotional IVEs.

1.2 Objectives

The main objective of this thesis is to propose emotional models that allow MAS to interpret and/or emulate different emotional states and, also, emulate emotional contagion phenomena. Moreover, the integration of proposed models with Intelligent

Virtual Environments (IVE) is used to achieve more complex simulations based on emotions in an effort to improve the human-computer interaction in emotional-based application such as IoT, AIm and SH applications.

According to the main objective, the contributions of this work are organized into specific goals as describe below:

- **Study of the state-of-art of emotional modeling and multi-agent systems:** It is necessary to survey, classify, and review the existing literature on emotional modeling and its applications on MAS. Furthermore, it is necessary to include a categorization of the emotional and personality models, as well as the techniques to perform the detection and simulation of emotions.

- **Simulation tool:** The main challenge in human-computer interaction lies in the design and construction of IVEs, in which humans interact with autonomous and intelligent entities through different input and output devices. Traditionally, IVEs use the client/server paradigm, but due to their features, a distributed approach such as multi-agent systems (MAS) seems to fit in the development of components that will evolve autonomously and coordinate with environmental evolution. We aim to design a method to develop this kind of IVEs along with a supporting platform to execute them.

  - Develop and validate a simulation tool that allows to design and execute IVEs.

- **Social emotional model:** Human behavior is related with the management of emotions levels which depends on the perception and variation of stimuli from the environment. The changes in emotional state could be a useful information for human-computer interaction. However, computer systems lack of the ability to recognize and interpret such changes. Therefore, it is necessary to propose and validate social emotional models to represent different emotions.

  - Propose and validate a Social emotional model that determine the emotional state of a group of agents.
  - Propose and validate a dynamic emotional model, as a extension of the social emotional model, that allows to represent emotional contagion in a heterogeneous group of agents capable of express and/or communicate emotions.

- **Emotional-based applications** In order to test the proposed emotional models and their further integration in IVEs, it is necessary to implement some emotional-based applications for proving the interaction between humans and agents living within IVEs and using emotions as a form of communication.

1.3 Structure of the Thesis

The remainder of this thesis is structured as follows.

Part II. Selected Papers: This part consists on a collection of the main articles (conferences and journals) published by the PhD. student which support this thesis.

Part III. Discussion: This part provides a discussion about all the obtained results presented in the selected papers.

Part IV. Conclusions: This last part is devoted to present a final review of the conclusions as well as promising directions for further works.

1.4 Publications List

In this section, all the international publications related to this thesis are listed. They have been classified according to their type (journals or international conferences) as well as whether they are listed in JCR, respectively. Those publications which have been included in this document are marked with (*).

- Journals listed in JCR:
  - (*) J.A. Rincon, Emilia Garcia, V. Julian and C. Carrascosa The JaCalLive framework for MAS in IVE: a case study in evolving modular robotics Neurocomputing Accepted In press. (2017) Impact Factor: 2.392
  - (*) J.A. Rincon, J. L. Poza, V. Julian, J. L. Posadas and C. Carrascosa Extending MAM5 Meta-Model and JaCalIV E Framework to Integrate Smart Devices from Real Environments Plos ONE N. 10.1371/jo pp. 1-27. (2016). Impact Factor: 3.54
  - (*) J.A. Rincon, F. de la Prieta, D. Zanardini, V. Julian and C. Carrascosa Influencing over People with a Social Emotional Model Neurocomputing pp 47-54. (2017) Impact Factor: 2.392
  - (*) J.A. Rincon, V. Julian and C. Carrascosa Developing an Emotional-based Application for Human-Agent Societies Soft Computing A Fusion of Foundations, Methodologies and Applications Vol. 20 pp. 1-12. (2016). Impact Factor: 1.630
  - (*) J.A. Rincon, A. Costa, G. Villarubia, V. Julian and C. Carrascosa Introducing Dynamism in Emotional Agent Societies Neurocomputing. Accepted In press. (2017) Impact Factor: 2.392
  - J.A. Rincon, J. Bajo, A. Fernández, V. Julian and C. Carrascosa Using emotions for the development of Human-Agent Societies Frontiers of Information Technology & Electronic Engineering Vol. 10.1631/FI pp. 1-13. (2016). Impact Factor: 0.392
- International conferences listed in CORE:
  - Rincon, Jaime A. and Botti, Vicente and Julian, Vicente and Carrascosa, Carlos. *Extending JaCalIVE Framework to Create Virtual Worlds by Means of an OWL Ontology*. International Conference on Autonomous Agents and Multiacgent Systems, AAMAS 2017 São Paulo, Brazil,

May. Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems. CORE Rank: A.
http://dl.acm.org/citation.cfm?id=3091125.3091414

- Other international conferences:
  - J.A. Rincon, Emilia Garcia, V. Julian and C. Carrascosa Developing Adaptive Agents Situated in Intelligent Virtual Environments International Conference on Hybrid Artificial Intelligence Systems pp. 98-109. (2014)
  - J.A. Rincon, C. Carrascosa and Emilia Garcia Developing Intelligent Virtual Environments using MAM5 Meta-Model International Conference on Practical Applications of Agents and Multi-Agent Systems pp. 379-382. (2014)
  - J.A. Rincon, Emilia Garcia, V. Julian and C. Carrascosa From Virtual to Real, Human Interaction as a Validation Process for IVEs 9th International Symposium on Intelligent Distributed Computing - IDC'2015 pp. 49-59. (2015)
  - J.A. Rincon, V. Julian and C. Carrascosa Social Emotional Model 13th International Conference on Practical Applications of Agents and Multi-Agent Systems Vol. 9086 N. 13th pp. 199-210. (2015)
  - J.A. Rincon, V. Julian and C. Carrascosa Applying a Social Emotional Model in Human-Agent Societies Workshop WIHAS'15. Intelligent Human-Agent Societies. Vol. 524 pp. 377-388. (2015)
  - J.A. Rincon, V. Julian and C. Carrascosa An Emotional-based Hybrid Application for Human-Agent Societies 10th International Conference on Soft Computing Models in Industrial and Environmental Applications Vol. 368 pp. 203-214. (2015)
  - J.A. Rincon, V. Julian and C. Carrascosa Representing Social Emotions in MAS 13th International Conference on Practical Applications of Agents and Multi-Agent Systems Vol. 9086 N. 13th pp. 308-311. (2015)

- J.A. Rincon, J. L. Poza, J. L. Posadas, V. Julian and C. Carrascosa Adding real data to detect emotion by means of smart resource artifacts in MAS Advances in Distributed Computing and Artificial Intelligence Journal Vol. 5 N. 4 pp. 85-92. (2016)
- A. Costa, J.A. Rincon, V. Julian, C. Carrascosa and P. Novais Emotions Detection on an Ambient Intelligent System Using Wearable Devices Affective Computing and Context Awareness in Ambient Intelligence (AfCAI 2016).
- J.A. Rincon, A. Costa, P. Novais, V. Julian and C. Carrascosa Using non-invasive wearables for detecting emotions with intelligent agents 11th International Conference on Soft Computing Models in Industrial and Environmental Applications pp. 73-84. (2016).
- J.A. Rincon, A. Costa, P. Novais, V. Julian and C. Carrascosa Detecting Social Emotions with a NAO robot 14th International Conference on Practical Applications of Agents and Multi-Agent Systems pp. 86-289. (2016).
- J.A. Rincon, A. Costa, P. Novais, V. Julian and C. Carrascosa A Dynamic Emotional Model for Agent Societies 14th International Conference on Practical Applications of Agents and Multi-Agent Systems pp. 169-182. (2016).

1.5 Research Projects

The research work presented in this PhD Thesis was carried out in the context of the following research projects:

- **HUMBACE: HUMAN-LIKE COMPUTATIONAL MODELS FOR AGENT-BASED COMPUTATIONAL ECONOMICS.**

  - Funder: Generalitat Valenciana. *PROMETEOII/2013/019*
  - Lead Applicant: Vicente Botti Navarro
  - Years: 2013 - 2019

- **iHAS: Sociedades Humano-Agente: Diseño, Formación y Coordinación**
  - Funder: Ministerio de Economía y Competitividad *TIN2012-36586-C03-01*
  - Lead Applicant: Vicente J. Julián Inglada
  - Years: 2013 - 2015

1.6 Research Visits

During the development of this thesis, the following research visit was carried out:

- 27-02-2017 to 27-05-2017,Universidade do Minho, Portugal, research visit supervised by Prof. Paulo Novais. This research visit has been funded by International mobility grants for PhD students from the Universitat Polit‘ecnica de Val‘encia.

1.7 Funding

This PhD thesis has been funded by the FPI grant AP2013-01276, awarded to author.

Part II

Selected Papers

CHAPTER 2

The JaCalLive framework for MAS in IVE: a case study in evolving modular robotics

AUTHORS:
J. A. RINCON, EMILIA GARCIA, V. JULIAN AND C. CARRASCOSA
{*jrincon, mgarcia, vinglada, carrasco*}*@dsic.upv.es*
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN
UNIVERSIDAD POLITÉCNICA DE VALENCIA
CNO/ DE VERA SN
46022 VALENCIA, SPAIN

Abstract

This paper presents a framework specially designed for the execution and adaptation of Intelligent Virtual Environments. This framework, called JaCalIVE, facilitates the development of this kind of environments managing in an efficient and realistic way the evolution of parameters for the adaptation of the physical world. The framework includes a design method and a physical simulator which is in charge of giving the Intelligent Virtual Environment the look of the real or physical world, allowing to simulate physical phenomena such as gravity or collision detection. The paper also includes a case study which illustrates the use of the proposed framework as an evolutive algorithm which allows the automatic adaptation of modular robots.

2.1 Introduction

Nowadays, having software solutions at one's disposal that enforce autonomy, robustness, flexibility and adaptability of the system to develop is completely necessary. The dynamic agents organizations that auto-adjust themselves to obtain advantages from their environment seems a more than suitable technology to cope with the development of this type of systems. These organizations could appear in emergent or dynamic agent societies, such as grid domains, peer-to-peer networks or other contexts where agents dynamically group together to offer compound services as in Intelligent Virtual Environments (IVE). An IVE is a virtual environment simulating a physical (or real) world, inhabited by autonomous intelligent entities [7].

Today, this kind of applications are between the most demanded ones, not only as being the key for multi-user games such as *World Of Warcraft*[1] (with more than 7

[1]http://eu.battle.net/wow

million of users in 2013)[2] but also for immersive social networks such as *Second Life*[3] (with 36 million accounts created in its 10 years of history)[4]. It is in the development of these huge IVEs where the need of a quick and easy-to-use modelling toolkit arises.

These kinds of IVEs are addressed to a huge number of simultaneous entities, so they must be supported by highly scalable software. This software has also to be able to adapt to changes, not only of the amount of entities but also of their users needs. Technology currently used to develop this kind of products lacks of elements facilitating the adaptation and management of the system. Traditionally, this kind of applications use the client/server paradigm, but due to their features, a distributed approach such as multi-agent systems (MAS) seems to fit in the development of components that will evolve in an autonomous way and coordinated with the own environment's evolution. In the last decade, MAS technology has been successfully employed in similar large scale distributed systems such as Robocup Rescue simulation [75].

This paper presents the JaCalIVE[5] (Jason Cartago implemented Intelligent Virtual Environment) framework. It provides a method to develop this kind of IVEs along with a supporting platform to execute them. JaCalIVE is based on the MAM5 meta-model [14] which describes a method to design IVEs.

MAM5 is based in the A&A (Agent & Artifact) meta-model [109] that describes environments for MAS as populated not only by agents, but also for other entities that are called *artifacts*. The A&A meta-model promotes the modelling and engineering of agent societies and MAS environment as first-class entities. According to MAM5 meta-model, an IVE is composed of three important parts: artifacts, agents and physical simulation. Artifacts are the elements in which the environment is modelled. Agents are the IVE intelligent part. The physical simulation is in charge of giving the

[2]http://www.statista.com/statistics/276601/number-of-world-of-warcraft-subscribers-by-quarter/
[3]http://www.secondlife.com
[4]http://massively.joystiq.com/2013/06/20/second-life-readies-for-10th-anniversary-celebrates-a-million-a/
[5]http://jacalive.gti-ia.dsic.upv.es/

IVE the look of the real or physical world, allowing to simulate physical phenomenal such as gravity or collision detection.

In order to evaluate the proposed framework we have chosen a case study consisting in the implementation of an evolutive algorithm which allows the automatic creation of modular robots optimized for a specific task. Specifically, the implemented system simulates a genetic algorithm where robots can interact among them in order to change its shape by joining other modules or environment objects. In that sense, one simple modular robot can change its shape and create a complex robot depending on the movement required. The aim of each robot is to minimize the distance between its real movement and the movement defined by the fitness function. During the simulation robots will evolve or will be destroyed following the rules of an evolutive algorithm. The modular robots obtained at the end of the simulation will be the better adapted and the most appropriated to do the required movement.

The rest of the paper is organized as follows: Section 2.2 summarizes the most important related work. Section 2.3 describes the JaCalIVE framework. Section 2.4 presents the proposed case study based on the automatic evolution of modular robotics developed using JaCalIVE. Finally, Section 6.7 summarizes the main conclusions of this work.

2.2 Related work

This section summarizes the most relevant techniques and technology that the JaCalIVE framework integrates in order to design and simulate IVEs. These techniques allow JaCalIVE to develop IVEs that are realistic, complex, adaptable, and with autonomous and rational entities. First, some concepts about IVEs are presented, to continue commenting about Multi-Agent Systems concepts, as platforms and methodologies relevant to the present work. Finally, this section presents the MAM5 meta-model, as it is the starting point for the present work to model IVEs in MAS terms.

2.2.1 IVE

Currently, there is an increasing interest in the application of IVEs in a wide variety of domains. IVEs have been used to create advanced simulated environments [175], [83], [174] in so different domains as education [9], entertainment [123], [5], [141], e-commerce [45], health [160], [90] and use to VR-based simulations [140].

One of the key features of any IVE is to offer a high level of user immersion. In order to achieve that, it is necessary that the IVE has the ability of simulating physical conditions of the real world such as gravity, friction and collisions. Besides, in order to increase the graphical realism, the physical simulators should include dynamic and static objects that inhabitat the IVE in a three-dimensional environment. A comparative analysis of some of the most important developed physical simulation tools can be found in [22]. The conclusion of this analysis shows that, of the open source engines, the *JBullet*[6] engine provided the best results overall, outperforming even some of the commercial engines. In this kind of simulators in which it is pretended to create a robotic simulation, it is important to give to the developer a physical engine. This engine allows to create constraints in the environment. These constraints pretend to create realistic simulations. It is very common to find simulators in which the physical engine is coupled to the stage of visualization. Some examples of this kind of simulators are Gazebo [76], V-REP[7], and Molecubes [178]. Gazebo uses ODE physical engine for physics simulation. Gazebo (similarly to aforementioned Stage simulator) can be controlled using the Player API and from Robot Operating System (ROS). V-REP is a general-purpose simulator that is used to create general robotics simulators. Whilst Molecubes [178] simulator is a more specific one, addressed to modular robotics (it was developed for Molecubes modular robots). It is based on *NVidia PhysX* and the visualization is implemented using the OGRE library. These simulators do not consider the use of multi agent systems. All these simulators employ either ODE or Bullet physics engine and this physical

[6] jbullet.advel.cz/
[7] www.coppeliarobotics.com/

engine is associated to the 3D rendering process. As an example of IVEs populated by agents (MAS) DIVAs 4.0 [6, 3] framework has to be underlined, as it allows to simulate an environment populated by agents, decoupled from them (although this environment / simulation is not based on agent-technology as in the proposal here presented).

2.2.2 Multi-Agent Systems

Until now, we have highlighted the importance of giving *realism* to IVEs, which would enable the user to have the desired level of immersion. This realism is provided by the physical simulation and 3D visualization, but this is only one part of a virtual environment. To be an IVE, a virtual environment needs to give entities with the intelligence to enhance the user's immersion.

MAS is one of the most employed artificial intelligence technique for modeling IVEs. This is mainly due to the characteristics that agents have, such as autonomy, proactivity, reactivity and sociability. But this does not mean that no other AI techniques can be used within MAS for IVE development. An agent can include as a decision-making mechanism other algorithms that improve the deliberative process such as reinforcement learning [51, 39], genetic algorithms [73], markov models [43], classification [40], neuronal networks [70] or use any method of hybrid artificial intelligence systems [35] etc.

However, when modeling an environment it is necessary to take into account that not all the entities are agents. The A&A meta-model [128, 127] describes a methodology for modeling environments using artifacts. Artifacts represent the first level of abstraction when modeling environments. This is mainly due to the clear differentiation of the entities which are in systems of this kind. This differentiation can determine which items are objects (Artifacts) and which are intelligent entities (Agents).

The BDI model (Belief - Desire - Intention) [23, 125, 53] is the most well-known and used agent model when designing intelligent agents. This model is based on logic and psychology, which creates symbolic representations of beliefs, desires and intentions of the agents. The beliefs are the information the agent has about the environment. This information can be updated at each time step or not. This obsolescence of the used information forces the agent to perform deliberative processes. Desires are the possible actions that the agent could make. This does not mean that every desire of an agent has to be performed. Finally, intentions represent the actions that the agent has decided to perform. These actions may be goals that have been delegated to the agent or may be the result of previous deliberation processes.

Different approaches have been devised in order to develop MAS. One of the first tools used for implementing agents is the JADE platform. JADE has been used for the development of *JGOMAS (Game Oriented Multi -Agent System based on Jade)* [12, 10, 11]. JADE does not directly provide a BDI model but there is an extension called *JADEX* allowing developers to design BDI-oriented MAs incorporating the representation of beliefs, desires and intentions. JADEX has been used for modeling environments like the presented in [18]. *Jason* is another development tool used for MAS programming which also integrates the BDI model.

In our proposal we employ Jason as the programming toolkit for our BDI agents [167]. The main reason to employ JASON is its full integration with CArtAgO (Common ARTifact infrastructure for AGents Open environments) [126]. CArtAgO is a framework/infrastructure for modeling artifacts which can run virtual environments. This framework allows the implementation of open work-spaces, which facilitates the creation of distributed environments.

2.2.3 MAM5

MAM5 [14] is a model to design IVEs based in the A&A meta-model. It is addressed to be used by an IVE designer, that wants to design an IVE based on a multi-agent system. As it is intended to be distributed, the human interface part of the system is decoupled from the intelligent part, being only this last one the part designed by means of MAM5. To have this two parts distributed facilitates the developing, gives more flexibility to the final applications (allowing different interfaces to be connected and at the same time) and allows to scale the final system (thinking on massive applications with a huge number of users and/or agents).

This model classifies the entities in the design into two different sets (as seen in Figure 2.1). The first one is related to all the entities that do not have any physical representation in the IVE (Non Virtually Physical Situated), whilst the second one is formed by all the entities having a representation inside the IVE (Virtually Physical Situated). Inside the former set there are Agents, Artifacts, and Workspaces following the A&A definition. In a similar way, inside the last set there are IVE Artifacts and Inhabitant Agents that are situated in the virtual environment (in fact, the Inhabitant Agent will have an IVE Artifact representing its body in the IVE), and IVE Workspaces, representing the virtual place, and the laws defining and governing such places.

MAM5 meta-model not only allows to differenciate between virtual represented entities and not virtual represented, but it also incorporates the definition of the physical restrictions and properties in the modelling of the environment and of their inhabiting entities, respectively. That is, the designer may define the different IVE Laws governing the different IVE Workspaces in the IVE (representing the physical laws of the real world) and, he may also define the different physical properties of the entities populating such virtual environment (mass, length, ...).

2.3 JaCalIVE

In the last years, there have been different approaches for using MAS as a paradigm for modelling and engineering IVEs, but they have some open issues: low generality and then reusability; weak support for handling full open and dynamic environments

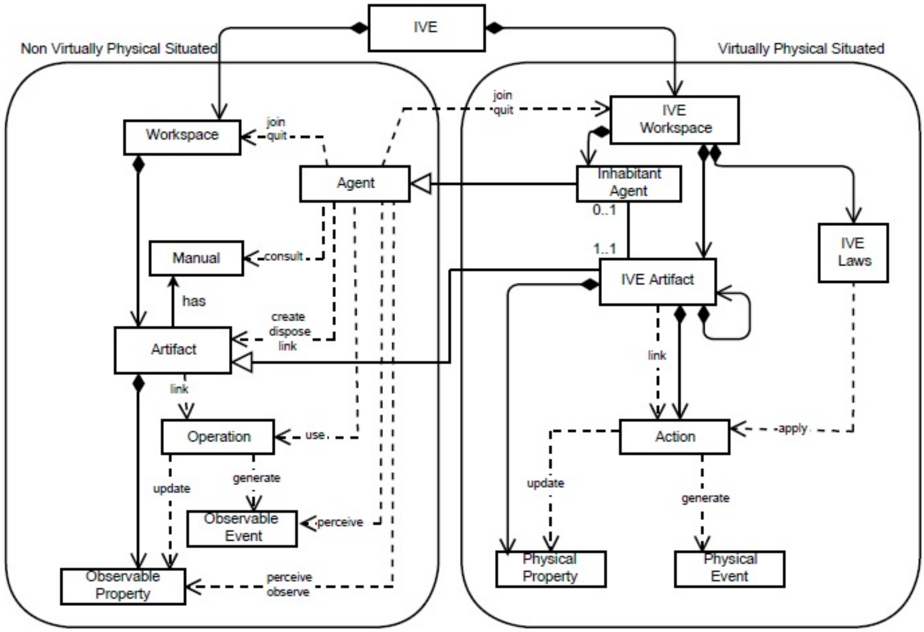


Figure 2.1: MAM5 meta-model of an IVE based onfigures A&A

where objects are dynamically created and destroyed. Those issues were addressed in developing JaCalIVE [133][139][132]. In JaCalIVE we can differentiate two parts, a virtual part and a non-virtual part. Each one of these parts are represented in a different workspace: a workspace including the entities that have a virtual representation called *IVE Workspace* and the other called *Workspace* including the entities that don't have a virtual representation. Each workspace is different but there exist communication between them. In the *IVE Workspace* the developer can design all entities that have a virtual representation (i.e Robot, characters bodies) and in the *Workspace* the developer can design all entities that don't have a virtual representation (i.e an agent that control a database or an agent that control the access). To differentiate which entities could be in this two parts, JaCalIVE incorporates a series of icons that facilitate the designed proses (Figure 2.2).

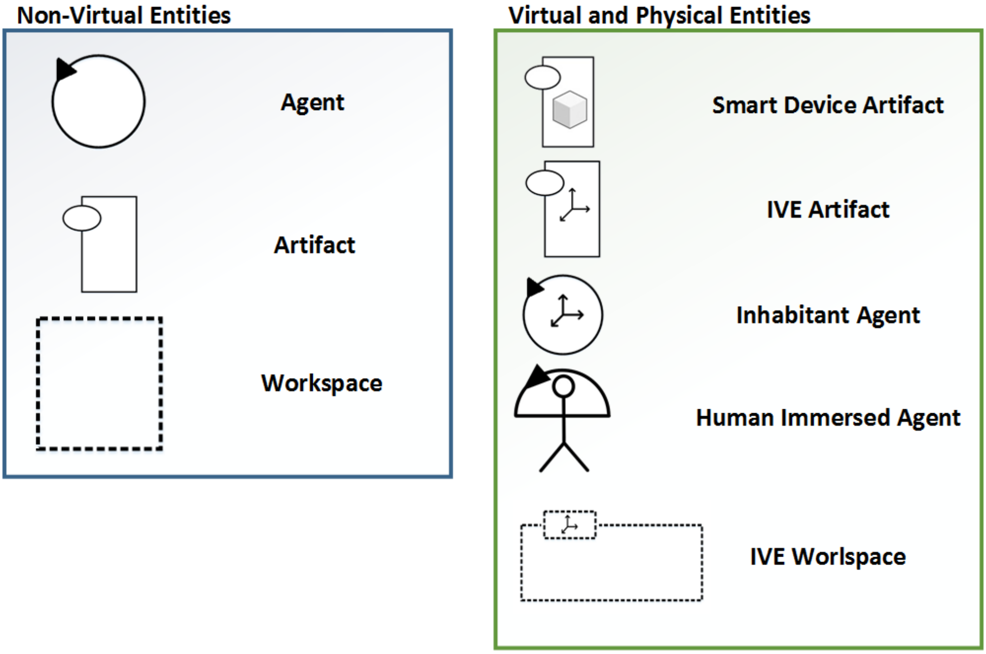


Figure 2.2: Type of Entities that Compose an IVE according to JaCalIVE

These icons facilitate the developer to distinguish between the two parts composing an IVE in the development process. At the same time, JaCalIVE incorporates two entities that allow to connect the IVE with the real world. This connection gives to the developer the positivity of capturing information of the real world and/or actuate in the real world. Following the MAM5 meta-modle, in JaCalIVE exist the following entities:

1. **Non-Virtual Entities**: The set of entities don't have a virtual representation.

    - **Agent:** represents the intelligent part of the autonomous pro-active entities in the system.
    - **Artifact:** represents the basic object that don't have a virtual representation.

2. **Virtual Entities**

   - **Inhabitant Agent:** representing agents that are physically situated in the virtual environment. This agent has a body in the IVE and could interact with other entities.
   - **IVE Artifact:** representing artifacts that are physically situated in the virtual environment. This artifact can be a body of an inhabitant agent or simply an object that could be used for the inhabitant agents.
   - **Human-Immersed Agent:** This is an special entity because it is a virtual representation of the human in the IVE. In the perception process, the Human-Immersed Agent associates a human in its process and follows him, capturing all information that the human can express. This agent incorporates process of machine learning as face detection and face classification.
   - **Smart Device Artifact:** This is a physical artifact. This artifact interacts with the real world. Commonly, it is a physical element that is composed of sensors and/or actuators, allowing to the Inhabitant Agent or the Human-Immersed Agent to interact with the real world.

At the same time, JaCalIVE incorporates a compiling process that produces agents and artifacts templates in which the developer programs the different behaviours of each entity. Figure 2.3 shows the steps that should be followed in order to develop an IVE according to the JaCalIVE framework:

1. Model: The first step is to design the IVE. JaCalIVE provides an XSD based on MAM5 meta-model. According to it, an IVE can be composed of two different types of workspaces depending on whether they specify the location of its entities (`IVE_Workspaces`) or not (`Workspaces`). It also includes the specification of agents, artifacts and the norms that regulate the physical laws of the IVE Workspaces.

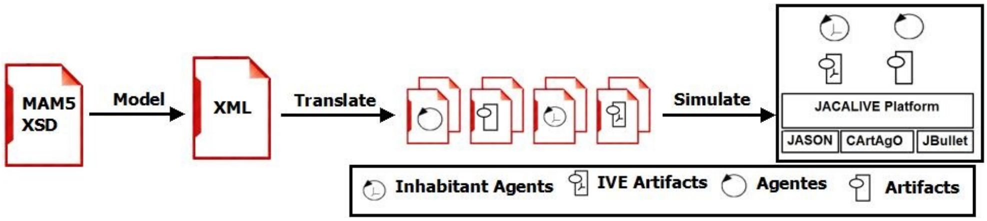


Figure 2.3: IVE developing general scheme using JaCalIVE.

2. Translate: The second step is to automatically generate code templates from design. One file template is generated for each agent and artifact. JaCalIVE agents are rational agents based on JASON. The artifacts representing the virtual environment are based on CArtAgO. The developer must complete these templates and then the IVE is ready to be executed.

3. Simulate: Finally the IVE is simulated. As is shown in Figure 2.3, JaCalIVE platform uses JASON, CArtAgO and JBullet. JASON offers support for BDI agents that can reason about their beliefs, desires and intentions. CArtAgO offers support for the creation and management of artifacts. JBullet offers support for physical simulation. JaCalIVE platform also includes internal agents (JASON based) to manage the virtual environment.

2.4 Case Study: Evolving modular robotics

2.4.1 Description

In this section a case study based on modular robots is described to show the versatility of JaCalIVE framework. Modular robots [151, 176] are robots mainly characterized by their ability to reconfigure their modules and changing their shape [115]. Each module of a robot is an independent entity that can be joined to other modules. This feature allows each robot to adapt its shape dynamically to changes in the environment. Currently, a wide range of domains of application are using modular robotics. For example, they are being used to search for missing persons in earthquakes [55] and to carry out space exploration [62]. These domains need advanced virtual simulation environments like the ones MAM5 and JaCalIVE allow to test their implementations. Moreover, simulations as the one presented in this paper can be used as testbeds for new adaptive algorithms, cooperative algorithms, Swarm Robotics, and so on.

Our case study implements a modular robots simulation that is composed of ten modular robot models. Each one of these modular robots are composed of individual pieces, each one with different possibilities, so that a robot must be composed at least of two of such pieces allowing it to move.

In this case, the whole system will behave as a genetic algorithm, where the chromosomes will be the different robots. Figure 2.4 shows a chromosome, that is, a modular robot, composed of three parts. Robots will move randomly finding pieces or other robots in the environment. Each robot r_i will have a different movement function f_i, being the goal of the system to approach as much as possible to a fitness function that models the robot movement as a sinus, that is, the goal is to minimise $g_i(x)$, being $g_i(x) = f_i(x) - sin(x)$. The two characteristic operations of a genetic algorithm are modeled:

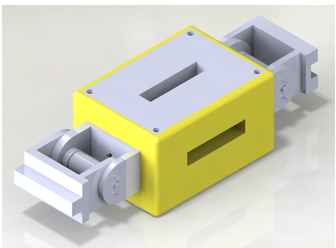


Figure 2.4: Chromosome represented by a modular robot

- Crossing: in this operation, two robots will join to form a different robot. This operation will make that the new robot will also have a new movement function as a result of composing the two functions of the crossed robots. Figure 2.5 shows an example of two robots formed each one of 3 pieces crossing and the result of this operation.

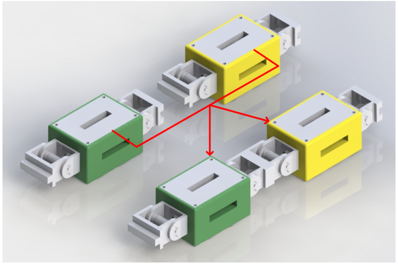


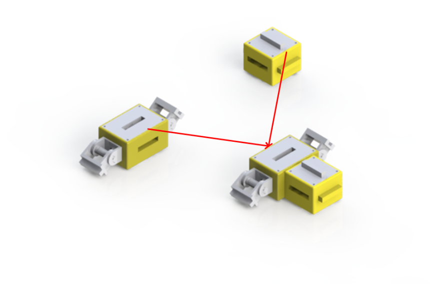
Figure 2.5: Crossing between two chromosome

- Mutation: In a genetic algorithm, mutation operations affect only one chromosome. In our case, these operations affect only one robot. There are two different kinds of mutations in our system: the first one is the one when a robot passes next to an individual piece and attaches it to its body (an example is shown in Figure 2.6(a)); whilst the last one is the one made internally in the robot when two pieces forming its body swap their places in it (as it is shown in the example of the Figure 2.6(b)).

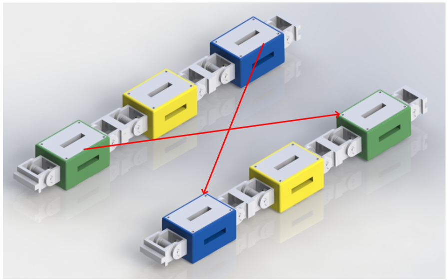
2.4.2 Developed System

The robots are within a map that is configured in the modeling process. In the tests made the map has a size of 300x300 without any wall. Robots can interact among them in order to change its shape by joining other modules or environment objects, that is, by moving and applying crossing and mutation operations as described above. In that sense, one simple modular robot can change its shape and create a complex robot depending on the environment requirements. The case study represents an environment where robots can evolve their shape and their movement.

Each one of the robots is modelled as an $Inhabitant\_Agent$, being a popullation of



(a) Chromosome mutation type 1



(b) Chromosome mutation type 2

Figure 2.6: Different kinds of chromosome mutation in our case study.

10 agents constant in the system. In the beginning of the execution, a random number of pieces are created and placed in the virtual environment, being each one of them an $IVE\_Artifact$.

Following the above mentioned process, the XML files are defined. The main parts of this XML are: an $IVE\_Workspace$ called $apodoRobot\_Workspaces$ as shown in Figure 2.7(a); a random set of $IVE\_Artifacts$ as shown in Figure 2.7(b); and ten $Inhabitant\_Agents$ as shown in Figure 2.7(c).

```xml
<VIRTUAL>
 <IVE_WORKSPACE NAME="apodRobot_workspace">
  <IVE_ARTIFACTS>
   <ITEM NAME="BodyLeft"/>
   <ITEM NAME="BodyRight"/>
   <ITEM NAME="linkedArtifact"/>
   <ITEM NAME="unlinkedArtifact"/>
  </IVE_ARTIFACTS>
  <INHABITANT_AGENTS>
   <ITEM NAME="Robot"/>
  </INHABITANT_AGENTS>
  <IVE_LAWS>
   <ITEM NAME="Gravity"/>
  </IVE_LAWS>
 </IVE_WORKSPACE>
```

(a) IVE_Workspace

```xml
<IVE_ARTIFACT NAME="linkedArtifact" LINKEABLE="true">
 <ATTRIBUTES/>
 <PHYSICAL_PROPERTIES>
  <PERCEIVABLE>
    <VECTOR3D NAME ="position">
  <DOUBLE NAME ="x"> 100.0 </DOUBLE>
  <DOUBLE NAME ="y"> 80.0 </DOUBLE>
  <DOUBLE NAME ="z"> 0.0 </DOUBLE>
 </VECTOR3D>
 <VECTOR3D NAME ="velocity">
  <DOUBLE NAME ="x"> 1.0 </DOUBLE>
  <DOUBLE NAME ="y"> 1.0 </DOUBLE>
  <DOUBLE NAME ="z"> 1.0 </DOUBLE>
 </VECTOR3D>
 <VECTOR3D NAME ="orientation">
  <DOUBLE NAME ="x"> 1.0 </DOUBLE>
  <DOUBLE NAME ="y"> 0.0 </DOUBLE>
  <DOUBLE NAME ="z"> 0.0 </DOUBLE>
 </VECTOR3D>
    <VECTOR3D NAME ="joint">
  <DOUBLE NAME ="x"> 0.0 </DOUBLE>
  <DOUBLE NAME ="y"> 0.0 </DOUBLE>
  <DOUBLE NAME ="z"> 0.0 </DOUBLE>
```

(b) IVE_Artifacts

```xml
<INHABITANT_AGENT NAME="Robot">
 <ATTRIBUTES/>
 <BODY_ARTIFACT>
  <ITEM ID="0"/>
  <ITEM ID="1"/>
 </BODY_ARTIFACT>
 <FILE NAME="apodRobotJason.asl"/>
</INHABITANT_AGENT>...

<IVE_LAW NAME="Gravity">
 <VECTOR3D NAME="gravity">
  <DOUBLE NAME="x"> 0.0 </DOUBLE>
  <DOUBLE NAME="y"> -9.8 </DOUBLE>
  <DOUBLE NAME="z"> 0.0 </DOUBLE>...
 </VECTOR3D>
 <ACTIONS>
  <ITEM NAME="move"/>
 </ACTIONS>
</IVE_LAW>
```

(c) Inhabitant Agents

Figure 2.7: XML generated in the modelling phase of the case study.

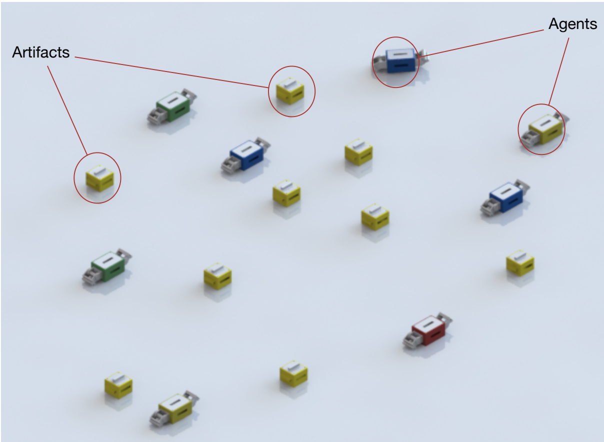
Agents and artifacts templates are generated through the compilation process, and they have been completed to have the described behaviour. Each agent is working in a cycle where it moves randomly in the environment, applying any possible operation with what it finds in its way (a crossing if it encounters another agent, a mutation type 1 if it encounters an artifact), or if it does not find any of this, it applies a mutation type 2 operation. At the end of each cycle, all the goal functions are calculated for each agent. The worst agents are *destroyed* or rather *reset*: the robot modules are reduced to the minimum expression (2 pieces) and its behaviour is also reset.

2.4.3 Execution tests and results

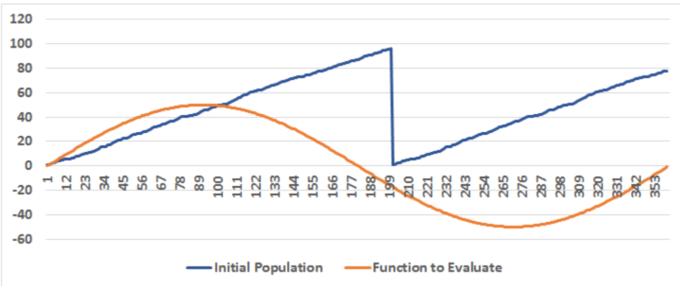
The proposed example has been tested in order to evaluate the evolution of the modular robots. The main idea was to observe the evolution of the pseudo-genetic algorithm in function of the different tests performed. As before commented, the initial population of the example has been fixed on 10 agents with a minimum capability to move around the environment and a random number of artifacts along the environment.

Figure 2.8(a) shows an initial generation of the proposed example. In the figure we can see different robot agents that will try to evolve through the crossing and mutation operations (previously defined) and a set of artifacts (individual pieces) that can be added to the body of a robot agent. At the beginning, if we apply the fitness function to each robot, there exists a high difference with respect the expected movement. Figure 2.8(b) shows the difference between the best movement achieved by a robot agent in the initial population and the desired sinus movement. As can be seen, the movement of this best initial agent can only be done along the positive y-axis.

After some iterations of the simulation, robot agents are evolving and changing their movement capabilities. Figure 2.9(a) represents and intermediate situation of the robot population after 100 iterations. In this situation we can only see a zoomed view of a part of the environment where it is placed the robot agent with the best



(a) Initial Map

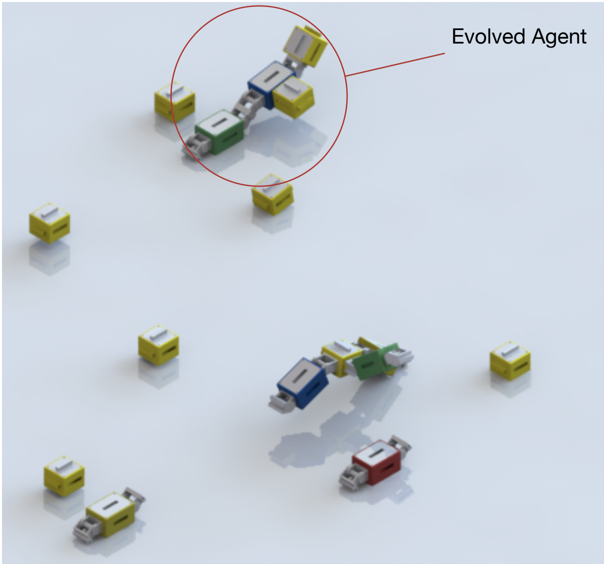


(b) Fitness evaluation

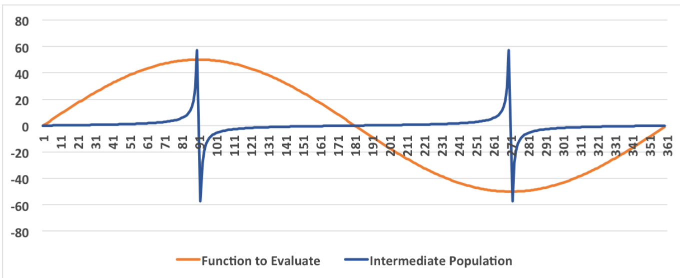
Figure 2.8: General view of the initial generation of modular robots

fitness value. As can be seen, the robot agent has evolved and it is formed by several pieces. In the same figure, we can see another robot formations that have evolved in a different way. The movement capability of these evolved robots have been improved as can be seen in Figure 2.9(b), where the best fitness value is compared again with the expected movement. In this case, the best evolved robot can move in both positive and negative y-axis.

Finally, Figure 2.10(a) shows the final scenario of the simulation after 187 iterations. In this figure, we can see the final structure of the robot agent with the best fitness



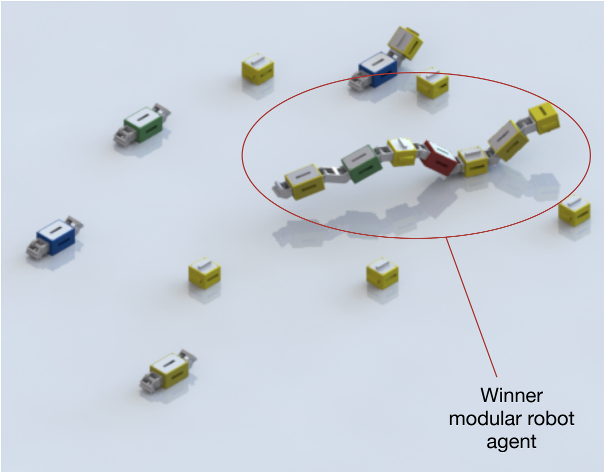
(a) intermediate Map



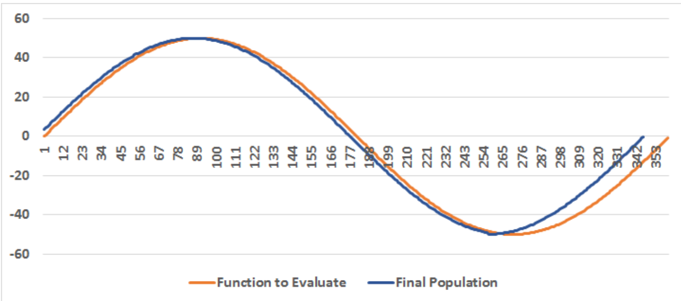
(b) Fitness evaluation

Figure 2.9: General view of the simulation during the execution

value. The robot has a structure similar to a snake and the obtained movement is very similar to the expected movement (as can be seen in Figure 2.10(b)). In this situation, the system has been able to obtain a robot configuration with the desired sine-wave movement capability.



(a) Final Map



(b) Fitness evaluation

Figure 2.10: View of the agent with the best fitness at the end of the simulation

2.5 Conclusions

In this paper we present a framework for the design and simulation of IVEs. This framework differs from other works in the sense that it integrates the concepts of agents, artifacts and physical simulation. Besides, IVEs developed using the JaCalIVE framework can be easily modified thanks to the XML modellation and the automatic code generation.

Following the MAM5 perspective, the modules used to interact with the developed IVEs are uncoupled from the rest of the system. It allows to easily integrate different kinds of modules as needed. For example, it allows to adapt the visualization render to the requirements of the specific IVE we want to simulate. To show the possibilities of such approach, a case study based on modular robotics is presented. These robots can adapt its shape to changing environment conditions. In the developed scenario, there are different modules in the environment that the agents can incorporate to their body, changing the way they move. Obtained results have shown the functionalities of the proposed framework. As far as we know, there is no other works that integrates in the same framework the above commented concepts for the development of modular robots. This makes impossible a comparison of the results. Nevertheless, for a better validation, we want to apply this framework to other application domains, like the industrial domain, where it can simulate the conditions inside a factory.

2.6 Acknowledgements

This work is partially supported by the TIN2011-27652-C03-01 and the FPI grant AP2013-01276 awarded to Jaime-Andres Rincon.

CHAPTER 3

Extending MAM5 Meta-Model and *JaCalIVE* Framework to Integrate Smart Devices from Real Environments

AUTHORS:
J. A. RINCON, JOSE-LUIS POZA-LUJAN, V. JULIAN, JUAN-LUIS POSADAS-YAG AND C. CARRASCOSA
{jrincon, vinglada, carrasco}@dsic.upv.es
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN
UNIVERSIDAD POLITÉCNICA DE VALENCIA
CNO/ DE VERA SN
46022 VALENCIA, SPAIN
INSTITUTE OF CONTROL SYSTEMS AND INDUSTRIAL COMPUTING (AI2)

3.1 Abstract

This paper presents the extension of a meta-model ($MAM5$) and a framework based on the model ($JaCalIVE$) for developing intelligent virtual environments. The goal of this extension is to develop augmented mirror worlds that represent a real and virtual world coupled, so that the virtual world not only reflects the real one, but also complements it. A new component called a smart resource artifact, that enables modelling and developing devices to access the real physical world, and a human in the loop agent to place a human in the system have been included in the meta-model and framework. The proposed extension of $MAM5$ has been tested by simulating a light control system where agents can access both virtual and real sensor/actuators through the smart resources developed. The results show that the use of real environment interactive elements (smart resource artifacts) in agent-based simulations allows to minimize the error between simulated and real system.

3.2 Introduction

The emergence of new virtual technologies such as *HoloLens* (https://www.microsoft.com/microsoft-hololens/en-us), in which the world is augmented using virtual objects is a first step in the creation of an *augmented world*. *Augmented worlds* were defined by Gelernter [52] as ”software models of some chunk of reality, some piece of the real world going on outside your windows.” Gelernter indicates that four keys are necessary to develop mirror worlds: a deep picture; a live picture; an agent; and history.

The *mirror world* is based on the concept of an *augmented world*, but in a *mirror world*, elements that enable high scalability are introduced. This *mirror world* integrates elements such as artificial intelligence (AI), augmented reality (AR), multi-agent systems (MAS) and mobile augmented reality. The design of this mirror world enables the creation of applications of ambient intelligence *AmI*, where the human may interact with virtual entities and these virtual entities may interact with the real world through sensors, robots, or any device that can connect with a human.

Software solutions that enforce autonomy, robustness, flexibility, and adaptability of a system under development are currently necessary. This is because the real world may always be changing and so software solutions need to change in real time. For this reason, these applications need to be robust enough to support a change in the environment and, at the same time, flexible enough to dynamically add elements.

Agent organizations that dynamically auto-adjust themselves to obtain advantages from their environment seem to be a suitable technology to cope with the development of this type of system. These organizations could appear in distributed and dynamic systems, such as grid domains, peer-to-peer networks, or other contexts where software agents dynamically group together to offer compound services as in intelligent virtual environments (IVE). An IVE is a virtual environment simulating a physical (or real) world inhabited by autonomous intelligent entities[7].

IVEs are addressed by a huge number of simultaneous entities, so they must be supported by highly scalable software. This software must also be able to adapt to changes in the number of entities and user needs. Current technology used to develop this kind of product lacks elements to facilitate the adaptation and management of the system. Traditionally, this type of application uses the client/server paradigm, but due to its features, a distributed approach such as multi-agent systems (MAS) seems compatible with the development of components that will evolve autonomously and coordinate with environmental evolution.

One approach for the development of these systems is the JaCalIVE framework. The JaCalIVE (http://jacalive.gti-ia.dsic.upv.es/)(Jason Cartago implemented Intelligent Virtual Environment) framework provides an agent-oriented method to develop IVEs along with a supporting platform to execute them. JaCalIVE is based on the MAM5 meta-model, which describes a method to design IVEs[14]. MAM5 is based in the A&A (agent &artifact) meta-model [109] that describes environments for MAS populated by agents and other entities that are called *artifacts*.

An IVE is composed of three important parts: artifacts, agents, and a physical simulation. Artifacts are the elements on which the environment is modelled. Agents are the IVE intelligent part. The physical simulation is in charge of giving the IVE the look of the real or physical world, enabling the simulation of physical phenomena such as gravity or collision detection.

This paper presents the extension of both the MAM5 meta-model and the JaCalIVE framework to model, design, and implement augmented worlds that more precisely mirror worlds. The concept of a smart resource artifact has been included in the meta-model in order to enable the design of the physical world in the environmental model. Moreover, the extended meta-model also enables the modelization of humans who can interact with other entities in the IVE. Finally, some experiments have been developed to validate the proposed extension. The example proposes the design of the optimal lighting for a road. To do this, the system tries to minimize the time that streetlights are on and the number of sensors needed to track driver itineraries.

Next, the most relevant methods, techniques, and technology used to devise mirror worlds are summarized. The starting point is a meta-model and framework for devising intelligent virtual environments. Then, it is commented on the $MAM5$ meta-model and $JaCalIVE$ framework that enables the development of IVEs in MAS terms. Some comments about accessing the physical real environment and *smart resources* are then introduced.

3.2.1 Intelligent Virtual Environment (IVE)

Currently, there is an increasing interest in the application of IVEs in a wide variety of domains. IVEs have been used to create advanced simulated environments [175, 83, 174] in domains such as: education [9], entertainment [123, 5, 141], e-commerce[45], health [160, 90] and VR-based simulations[140]. One of the key features of any IVE is that a high level of user immersion is offered. To achieve this it is necessary that the IVE has the ability to simulate physical conditions of the real world such as gravity, friction, and collisions. To increase graphical realism, physical simulators should include dynamic and static objects that populate the IVE in a three-dimensional environment. Relevant physical simulation tools include *JBullet*(http://jbullet.advel.cz/) and *Open Dynamic Engine (ODE)*(http://www.ode.org/). Another important feature of any IVE is a high level of graphic realism. Currently, there are available some well-developed graphical simulators such as *Unity 3D*(http://unity3d.com/unity),*UnrealengineUDK*(http://www.unrealengine.com/udk/) and *Cryengine*(http://www.crytek.com/cryengine). Although these simulators were initially designed for videogames, they can be used to simulate IVEs. In some IVEs, it is very important to get information of the real world. With this information, it is possible to know what is happening in the real world, allowing to create a complex simulation mixing the information of the real world and information emulated in the virtual world. This is also the idea behind augmented worlds or even mirror worlds. In some IVEs, it is very important to obtain information from the real world. With this information, it is possible to create a complex simulation mixing information from the real world and information emulated in a virtual world. This is also the idea behind augmented and mirror worlds. This type of application must also take into account that a human is immersed in the loop – producing a double immersion [146]. This double immersion enables humans to interact and communicate with agents using natural human interfaces, while at the same time, agents perceive and communicate with humans and other agents. This kind of interaction enables the creation of a *"human agent society"*, a type of application where agents offer services to humans or to other agents in an integrated environment.

3.2.2 Multi-Agent systems

Previously, it has been highlighted the importance of giving *realism* to IVEs so that users have the desired level of immersion. This realism is provided by physical simulation and 3D visualization, but this is only part of a virtual environment. To be an IVE, a virtual environment needs to give entities the intelligence to enhance the user's immersion MAS is one of the most popular artificial intelligence technique for modeling IVEs.

This is mainly due to the characteristics that agents have, such as autonomy, proactivity, reactivity, and sociability. But this does not mean that no other AI techniques can be used within MAS for IVE development. An agent can include as a decision-making mechanism other algorithms that improve the deliberative process – such as: reinforcement learning [51]; genetic algorithms [73]; Markov models [43]; classification [110]; and neuronal networks [70] or use any hybrid artificial intelligence system method [35].

However, when modeling an environment it is necessary to take into account that not all the entities are agents. The A&A meta-model [128, 127] describes a methodology for modeling environments using artifacts. Artifacts represent the first level of abstraction when modeling environments. This is mainly due to the clear differentiation among entities in systems of this kind. This differentiation can determine which items are objects (artifacts) and which are intelligent entities (agents).

The BDI model (belief - desire - intention) [23, 125, 53] is the best known and most frequently used agent model for designing intelligent agents. This model is based on logic and psychology, and creates symbolic representations of agent beliefs, desires, and intentions. The beliefs are the information the agent has about the environment. This information can be updated at each time step or not. The obsolescence of the used information forces the agent to perform deliberative processes. Desires are the actions that the agent could make. This does not mean that every desire of an agent must be performed. Finally, intentions represent the actions that the agent has decided to perform. These actions may be goals that have been delegated to the agent or may be the result of previous deliberation processes.

Different approaches have been devised to develop MAS. One of the first tools used for implementing agents is the JADE platform. JADE was used for the development of *JGOMAS (Game Oriented Multi -Agent System based on Jade)* [12, 10, 11]. JADE does not directly provide a BDI model but offers an extension called JADEX that enables developers to design BDI-oriented MAS that incorporate the representation of beliefs, desires and intentions. JADEX has been used for modeling environments such as that presented in [18]. *Jason* is another development tool used for MAS programming and also integrates the BDI model.

In this proposal Jason is employed as the programming toolkit for BDI agents[167]. The main reason for using JASON is its complete integration with CArtAgO (Common ARTifact infrastructure for AGents Open environments)[126]. CArtAgO is a framework/infrastructure for modeling artifacts that can run virtual environments. This framework allows the implementation of open work-spaces, which facilitate the creation of distributed environments.

3.2.3 MAM5

MAM5[14] is a model for designing IVEs that is based in the A&A meta-model. It is for use by IVE designers who want to design an IVE based on a multi-agent system. As it is intended to be distributed, the human interface part of the system is decoupled from the intelligent part (designed using MAM5). The fact that both parts are distributed facilitates development and gives more flexibility to the final applications (enabling different interfaces to be connected at the same time) and enables scaling the final system (massive applications with a huge number of users and/or agents).

This model classifies the entities in the design into two sets (as seen in Fig. 3.1). The first set is related to all the entities that do not have a physical representation in the IVE (non-virtually physically situated), while the second set is formed by all the entities having a representation inside the IVE (virtually physically situated). Inside the former set there are agents, artifacts, and workspace – in accordance with the A&A definition. In a similar way, inside the last set are IVE artifacts and inhabitant agents in the virtual environment (in fact, the inhabitant agent will have an IVE artifact representing its body in the IVE), and IVE workspace, representing the virtual place, and the laws defining and governing these places.

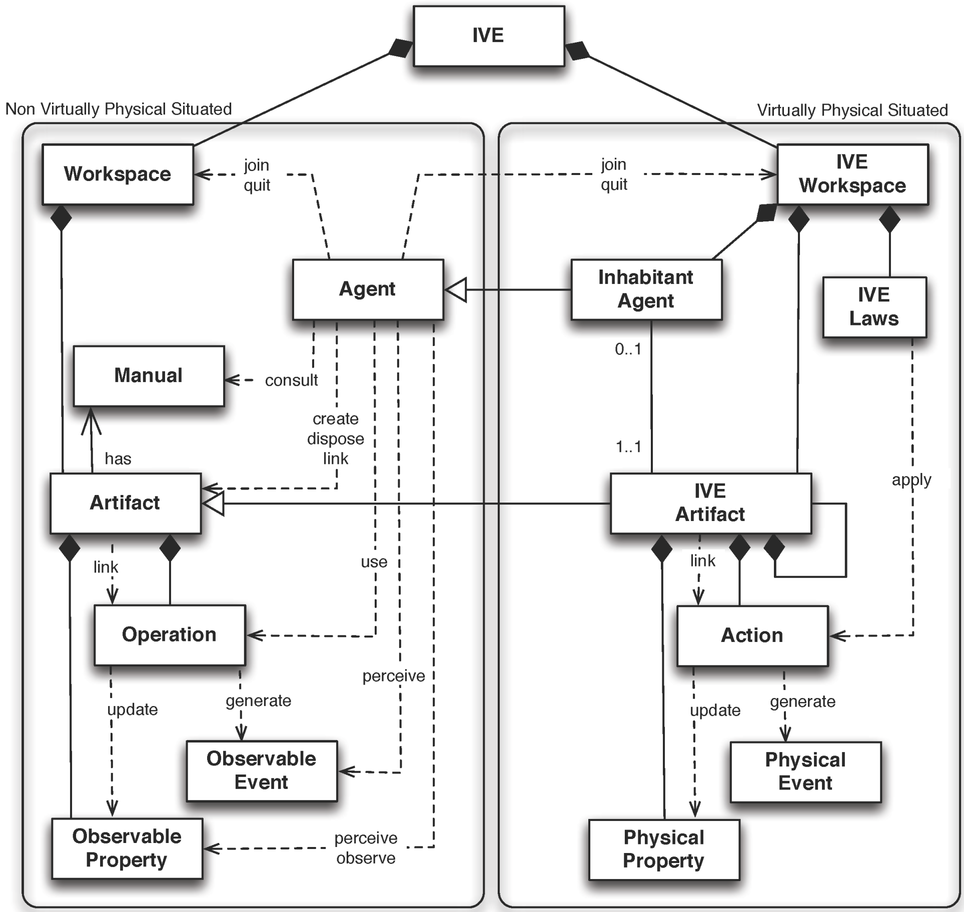


Figure 3.1: MAM5 Meta-model

The MAM5 meta-model enables a differentiation between virtually represented and non-virtually represented entities, and also incorporates the definition of physical restrictions and properties in the modelling of the environment and inhabiting entities, respectively. The designer may define the different IVE laws governing the IVE workspaces (representing the physical laws of the real world) and may also define the different physical properties of the entities populating the virtual environments (including mass and length).

Some of these physical properties are very difficult to simulate or need numerous calculations. Moreover, it is desirable to design very complex simulations and link the simulation to a real environment to produce augmented worlds, or even mirror worlds. To this end, the $MAM5$ meta-model is modified by adding two new properties as will be explained in Section *$MAM5$ and $JaCalIVE$ Extended*. To deal with this information, the concept of *smart resource artifact* is proposed to be incorporated into the meta-model.

3.2.4 JaCalIVE (Jason Cartago implemented intelligent virtual environment)

Various approaches have recently been explored for using MAS as a paradigm for modelling and engineering IVEs. However, several difficulties remain: low generality and reusability; and weak support for handling fully open and dynamic environments where objects are dynamically created and destroyed.

The $JaCalIVE$ framework is based on the MAM5 meta-model and was developed to tackle these difficulties. It provides a method to develop this type of application along with a supporting platform for execution. Fig. 3.2 shows the steps for developing an IVE according to the $JaCalIVE$ framework.

1. Model: the first step is to design the IVE. $JaCalIVE$ provides an XSD based on the MAM5 meta-model. Using this approach, an IVE can be composed of two types of workspaces depending on whether they specify the location of the entities (`IVE_Workspaces`) or not (`Workspaces`). The specification of agents, artifacts, and the norms that regulate the physical laws of the IVE

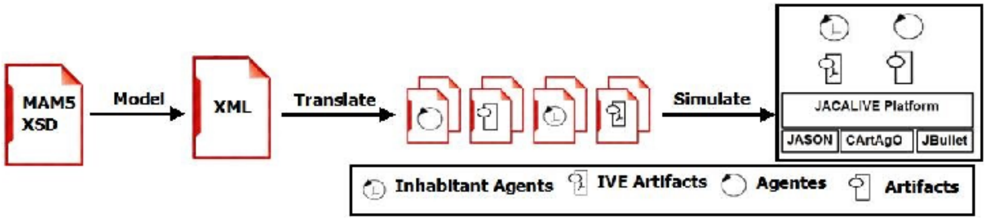


Figure 3.2: JaCalIVE General scheme

workspace is also included.

2. Translate: the second step is to automatically generate code templates from the design. This code is generated by parsing the XML file where the design is stored. One file template is generated for each agent and artifact defined in the model. As the XML file only includes a description of the agents and artifacts in the system, the template files generated include creation, accessing and communication between agents and artifacts, but they do not include any specific application code, that must be filled by the designer after the compilation process. Nevertheless, the creation of these files facilitates very much the IVE creation process. All the agents generated in this process are rational agents based on JASON, and the generated artifacts representing the virtual environment are based on CArtAgO. When the developer completes these templates with the application specific behaviour, the IVE is ready to be executed.

3. Simulate: Finally the IVE is simulated. As shown in Fig. 3.2, $JaCalIVE$ platform uses JASON, CArtAgO and JBullet. JASON offers support for BDI agents that can reason regarding their beliefs, desires, and intentions. CArtAgO offers support for the creation and management of artifacts, and JBullet offers support for physical simulation. $JaCalIVE$ platform also includes internal agents (JASON based) to manage the virtual environment.

3.2.5 Accessing the Environment

Among the distributed systems that interact with the real world, sensors and actuators are the physical interface [32] with the environment. Initially, sensors send data to clients in the same format that the data was acquired; for example, an RGB camera sends a frame with the acquired bitmap. Currently, the power the embedded systems (such as Arduino [78], Raspberry PI [165] or Beaglebone [173]) provides to sensors and actuators offers the possibility of transforming the raw data into specific information. When sensors or actuators provide processed data the device is usually called a smart device. Originally, in [152] Schmidt considers a smart device as *a device that is not ignorant about its environment*. When a smart device works in a distributed system, it is usually contextualized in a communications client/server paradigm. In [143], Salzmann defines a smart device as a device *where more intelligence is placed at the server side and little or no assumption is made about the client*. Some examples of smart devices are the Microsoft Kinect or the Asus xTion [56]. These kinds of devices provide images enriched with depth information directly from the server (smart device) to a client.

Currently, when a smart device offers more than the sensor information (or provides more than access to an actuator) in a distributed system, the smart device becomes a smart resource [104] and is contextualized in a publish/subscribe [59] paradigm. Examples include a sensor that provides specific environment elements such as tables or chairs in a home environment and clients that request the service: *let me know when the smart resource has found a chair*.

If a distributed smart resource provides smart information, the environment must be defined in the same smart terms. Consequently, the model used to observe the environment is at the core of a good MAS, and so there is a considerable literature defining what is a smart environment [34] or how to define the sensors that observe the smart environment [33].

3.2.6 Smart resources

Among the smart resources that operate in the real world, sensors read physical magnitudes and provide real information (from simple magnitudes such as temperature to complex information such as images acquired by an RGB camera). In the same way, actuators in the real world change physical magnitudes (or the resource characteristics): for example, by increasing the temperature or moving a resource. However, a smart resource provides access to this data and processes this data to produce richer information that is closer to the environment or client requirements. For example, a smart resource that offers the speed of detected objects in a bounded area, needs to detect the object and process data from various sensors by applying equations to calculate the speed. Smart resources must then publish the processing results in order for clients to receive the right information. Therefore, suitable and well known communication protocols must be used. Considering all the above, a distributed smart resource is structured in levels that convert it from a set of sensors or actuators to a resource with the capacity to process and distribute information. A distributed smart resource that is made up of three levels [103] is shown in Fig. 3.3.

The sensor/actuator level provides basic functions for access to acquisition/actuation hardware. The processing level adds intelligence to the device so that it can locally transform data into useful information for the client. For example, when locating specific objects (people) within a specified distance in an RGBD frame [30], a smart resource is used for a specific function for a specific client. Beyond the specific smart resource function, when the information processed by the smart resource can change or clients can dynamically change their requirements, it is necessary to offer the information as services by means an application program interface (API) and a protocol. In these last cases, the use of a smart resource concept is justified to discriminate between objects defined by clients [103].

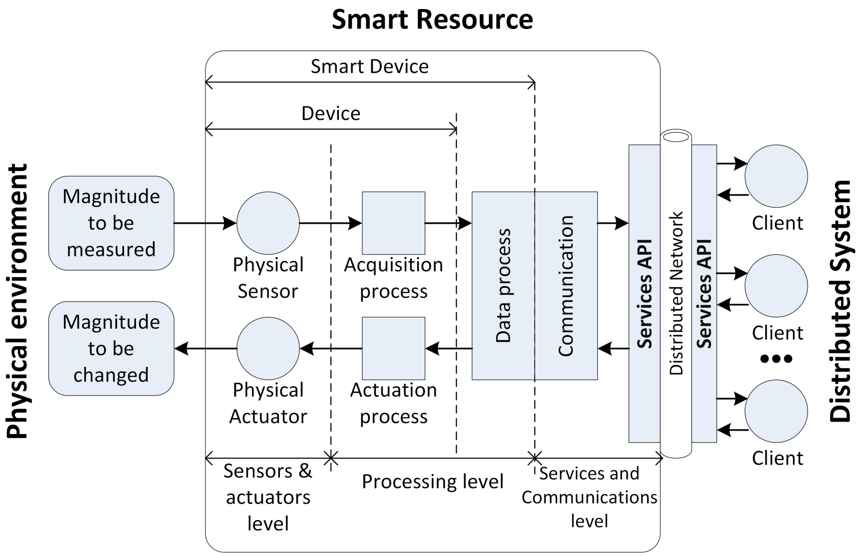


Figure 3.3: Components of a distributed smart resource.

3.3 Methods

In this section two different extensions incorporated into the *MAM5* meta-model and into the $JaCalIVE$ framework are presented. The first extension enables the *IVE* to connect with the real world, using physical artifacts to obtain information from the real world. By using these devices it is possible to build complex simulations, complement real sensorization with virtual information, or even enable agents to control different elements in the real world.

The second extension introduces the human in the loop, and enables humans to interact in both worlds (real and virtual) – meaning in the augmented or mirror world.

The rest of the section presents how these extensions were added to the $MAM5$ meta-model and to the $JaCalIVE$ framework.

3.3.1 Modified MAM5

The first approach of *MAM5*[14] was intended for an *IVE* designer who wants to design an *IVE* based on a multi-agent system. Therefore, the authors only took into account the separation between entities that have a virtual representation in the virtual environment (virtually physically situated) and the entities that do not have a physical representation in the *IVE*, that is, that are not situated (non virtually physically situated).

As commented previously, $MAM5$ has been extended for use in designing *IVEs* as well as augmented worlds, or even mirror worlds. For this reason, it has to initially include all the sensors and actuators that enable access to the real environment and so mix virtual and real environment data. Then, it is incorporated the human in the loop within this mix of virtual and real environments.

These elements enable us to increase the possibility of creating an interaction between the real world with the *IVE* and the human with the agents – thereby creating a meta-model that enables the modeling of augmented worlds (Fig. 3.4).

Human-immersed agents This kind of agent models the representation of a human in the system. It can be simply an interface of the human with the MAS, or it can have the capability of perceiving and recognizing the associated human [135]. This agent may even model the preferences of the associated human (e.g. the music preference [136]).

To sum up, the main goal is to model the human in the loop, taking into account the double immersion – meaning humans interacting with agents with natural interfaces and agents interacting with humans as with other agents.

Fig. 3.5 (right) shows the symbol used to model a human immersed agent.

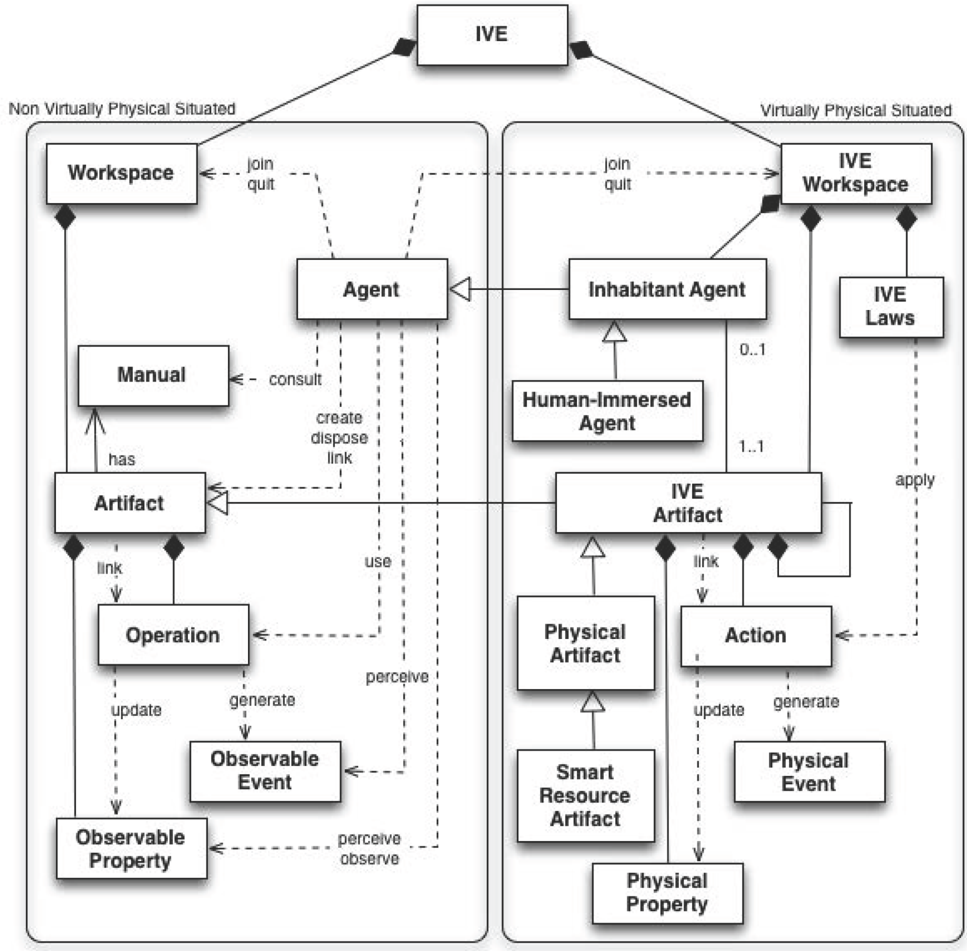


Figure 3.4: Modified MAM5.

Smart Resource Artifact A smart resource artifact (SRA) is an artifact that incorporates embedded sensors and actuators providing access to the environment in an MAS.

These artifacts must have advanced processing and communication capabilities to supply high-level information by abstracting agents from data acquisition processing

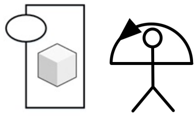


Figure 3.5: New symbols used in the $MAM5$ extension: smart resource artifact (left), and human immersed agent (right).

and recognition mechanisms.

In the proposed extended meta-model MAM5, an SRA is represented with a new symbol(see Fig. 3.5 left).

The functionality of an SRA can be provided by using a real device that works in the physical world. In this case, the SRA is a wrapper for the device. Additionally, this functionality can be simulated (Fig. 3.6). Agents will not distinguish this characteristic and will access both SRA through the same interface. In this way, agents can use an SRA without knowing the source of the data (real or simulated).

When the SRA represents a real device, this device is implemented according to the smart resource described in the introduction (see sub-section *Smart Resources*). In this case, the SRA can be considered as a client of the smart resource in a similar way to clients appearing in Fig. 3.3. The smart resource and its wrapper (smart resource artifact) communicate using a communications protocol based on services.

3.3.2 JaCalIVE Modified

In this section the extension of $JaCalIVE$ is presented to give support to the new modified $MAM5$ introduced in the previous section. These new elements enable the designer to connect the developed system to the real world and introduce humans in the same design. Using these new elements, it is possible to create a complex simulation, introducing measures such as temperature, pressure, biomedical signals, and other values. This measures will be used as perception of agents, and the agents can give an answer or change their behavior. Fig. 3.7 shows the general scheme of

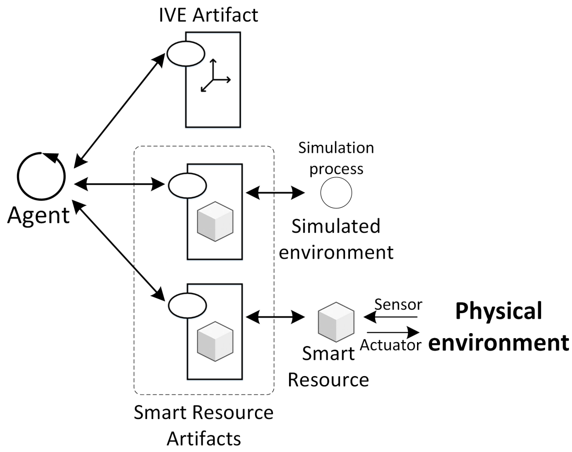


Figure 3.6: Interaction between agents and artifacts: IVE and SRA (simulated or a wrapper of a smart resource).

modified $JaCalIVE$. Applications are developed in the same way as the previous version of $JaCalIVE$. The main difference is that there is a template to model the peculiarities of human immersed agents, and a new kind of artifact enabling access to the real environment (SRA).

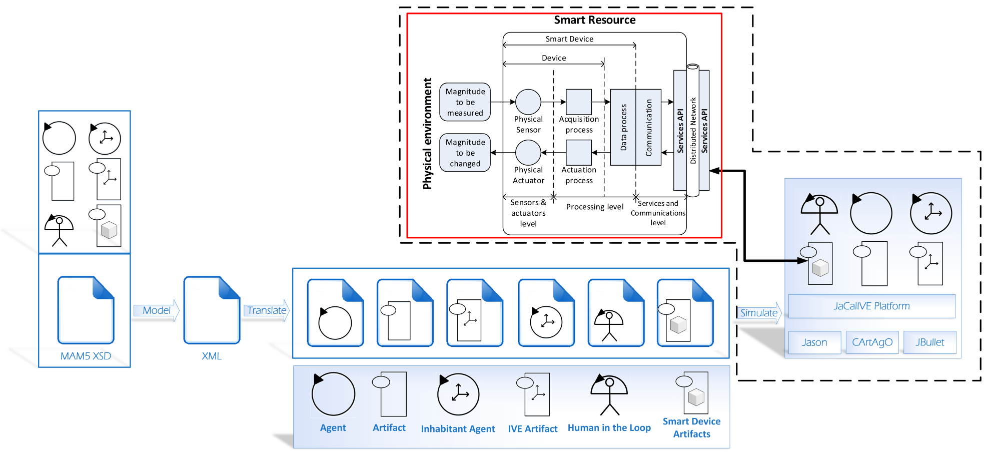


Figure 3.7: Modified $JaCalIVE$.

This access to the real world is possible when the SRA is connected to a smart resource (that is, the functionality of the SRA is not simulated). Communication between the SRA and the smart resource is based on access to services. The smart resource offers an interface to the services as used by its SRA wrapper in the system generated by the new *JaCalIVE* by following a proposed protocol (Fig. 3.8).

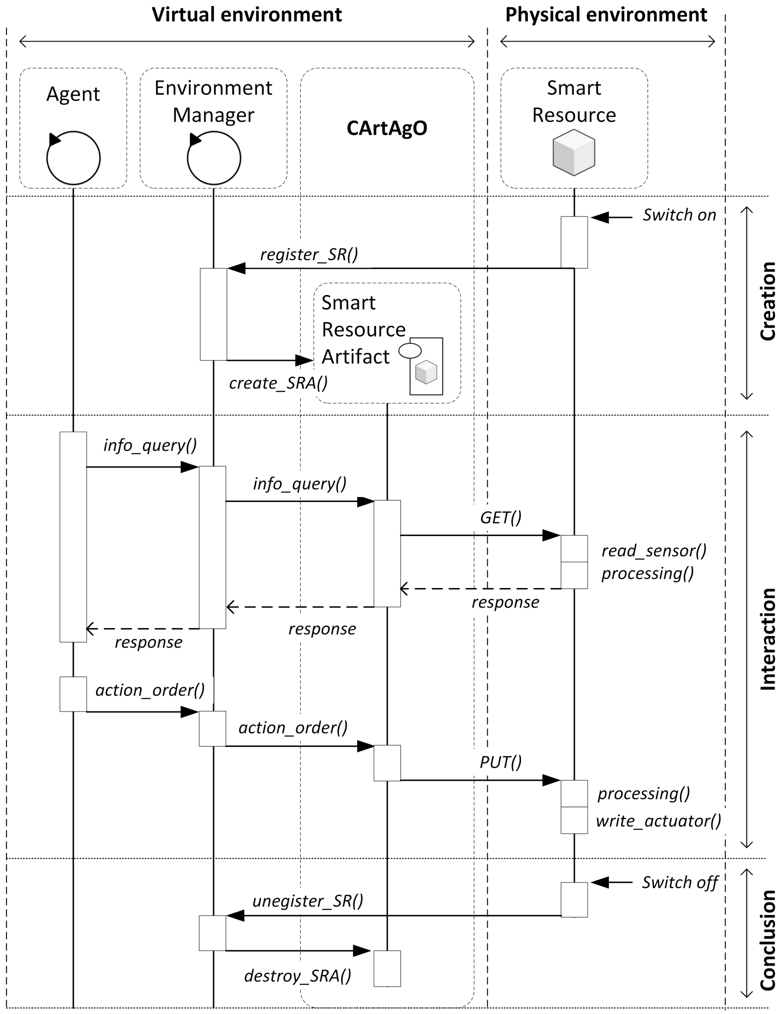


Figure 3.8: Protocol to work with a SRA based on a smart resource.

When a smart resource is switched on, it initializes sensors, actuators and communications and then connects to the environment manager in order to register itself in the virtual environment. For that, SRAs need only know the IP address of the environment manager that is public in the system. Once connected, the smart resource sends to the environment manager an XML message with content about its identification, location, and the resources or services it offers. The environment manager then creates a smart resource artifact associated with the smart resource.

With the structure of the XML message (Fig. 3.9), the environment manager knows how to access the resources/services offered by the registered smart resource. The *location* tag indicates the base URI (uniform resource identifier) of the smart resource and each *resource* tag identifies a different resource/service. A resource/service can be applied for by adding to the base URI its identifier. For example, the temperature could be obtained with: ”http://192.168.1.14/resources/temperature”. Every resource/service has its own URI in the same way that representational state transfer (REST) services work (the software architectural style of the World Wide Web) [129].

SRAs work with resources/services through standard HTTP operations such as GET (to obtain the value of a resource: for example, the current temperature from a sensor) or PUT (to send data to update a resource: for example, to switch on/off a light actuator). When an agent needs to access a resource, the agent makes a request to the environment manager that knows the associated SRA. The environment manager translates the request to the corresponding SRA and then this SRA uses the resource URI by sending a GET or PUT operation to the smart resource. In the case of a GET operation, the smart resource responds to the SRA with the content information of the resource by means of an XML message (Fig. 3.10) and, in the case of a PUT operation, the SRA sends to the smart resource an XML message (Fig. 3.11) with the information to be updated.

When the SRA is simulated (there is no smart resource associated) the protocol with agents (Fig. 3.12) is similar to that presented previously. In this case, the environment

```xml
<artifact>
 <type>physical smart resource</type>
 <name>speed sensor 01</name>
 <position>GPS coordenates</position>
 <location href="http://192.168.1.14:80/resources">Resources List</location>
 <resource>/speed
  <type>sensor</type>
 </resource>
 <resource>/temperature
  <type>sensor</type>
 </resource>
<!--
 ......
 ......
-->
  <type>actuator</type>
 </resource>
  <resource>/light
  <type>actuator</type>
 </resource>
 <resource>/light/intensity
  <type>actuator</type>
 </resource>
</artifact>
```

Figure 3.9: XML message to register a smart resource in the virtual world.

```xml
<resource>/temperature
 <type>sensor</type>
 <magnitude>Celsius</magnitude>
 <value>25</value>
</resource>
```

Figure 3.10: XML response message for GET operations.

```xml
<resource>/light
 <type>actuator</type>
 <magnitude>0_100</magnitude>
 <value>50</value>
</resource>
```

Figure 3.11: XML message for PUT operations.

manager creates the SRA depending on the system configuration. This SRA will simulate the smart resource functionality. To interact with the SRA, agents use the same protocol and interface as when a smart resource is associated with the SRA.

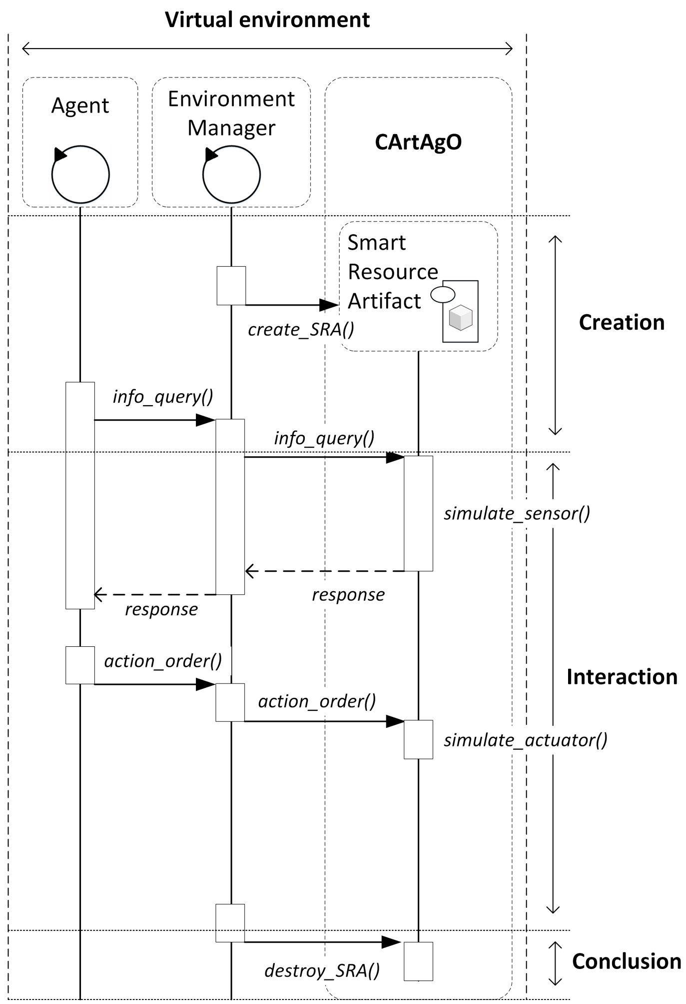


Figure 3.12: Protocol to work with a simulated SRA.

3.4 Results

To validate the proposal, the extended JaCalIVE framework has been tested in a scenario for managing streetlights. Streets are populated with a very large number of streetlights. Even if these streetlights are using LED lights, the total energy consumption may be very large. One solution is the automation of the on and off switch using detection sensors. The position of such sensors is usually based on the experience of the person making the installation. This leads to number of sensors being higher than necessary. Moreover, the installation of such sensors is progressive and very slow. The proposal presented here uses the extended JaCalIVE framework to enable a reduction in the number of sensors needed and the installation time (mixing real with virtual sensors). Using agents to control the streetlights on/off can help improve user comfort, as they may control how many streetlights are switched on so that the user always has the sensation of being in a fully illuminated environment.

More specifically, the proposed prototype is formed by a set of 18 streetlights distributed as can be seen in Fig. 3.13. The lighting power of these streetlights can be automatically adjusted by the system. The main goal of the proposed scenario is to provide the appropriate light to any car driving along the street while maintaining the highest possible number of streetlights off or with the lowest possible power. The state of the streetlights must be adapted as a consequence of car displacement. As a car is moving along the street, the system must adapt the street lighting without disturbing the drivers.

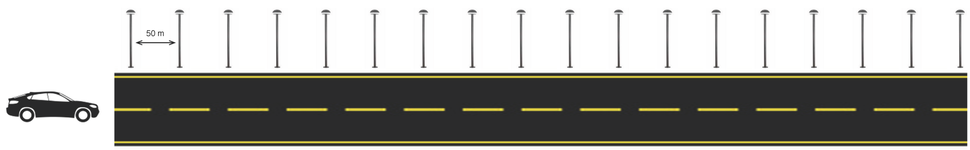


Figure 3.13: Prototype streetlights distribution

Variations of this scenario can be observed in Fig. 3.14. The first situation is the preferred one because it provides excellent lighting to the front of the car. The second and third situations can be considered adequate from the point of view of light consumption, but the driver does not have an adequate view of the street. Any other combination with more streetlights on is discarded due to its higher power consumption.

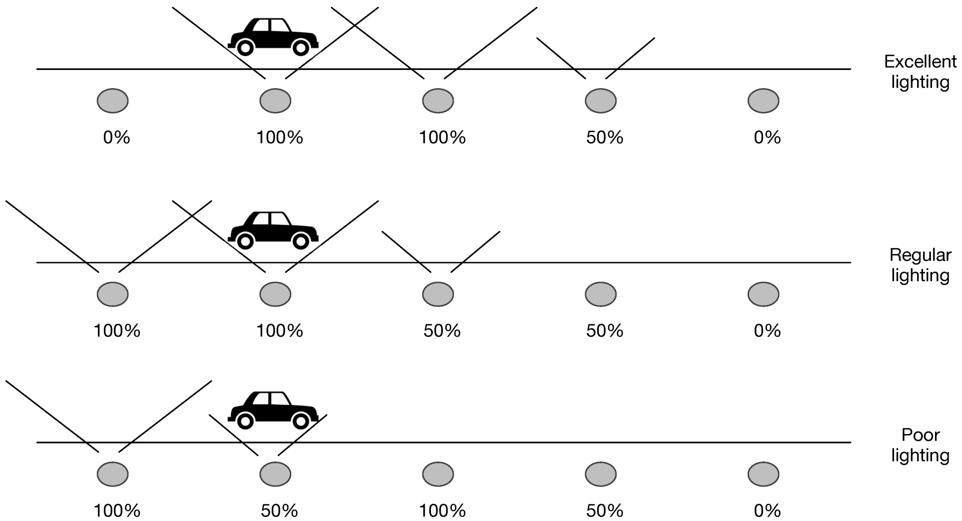


Figure 3.14: Prototype description view

3.4.1 System design

To develop the system described in the previous section, the following model using the extended *MAM5* meta-model has been devised (see Fig. 3.15):

- Each car is modelled as an *inhabitant agent* taking into account that it can decide to follow or not a constant velocity, increase or decrease its velocity, or even stop at any time.
- Each streetlight will be modelled as a *Smart Resource Artifact*, although there

are two types: some being only a wrapper for a real smart resource (with an actuator to switch on/off the streetlights and presence sensors to detect real cars and their velocity), and those in which the sensorization part is simulated (by means of an *IVE artifact*).

- A set of agents, each in charge of controlling n streetlights . This number, along with the number of real sensors, constitutes the two parameters that could be used to tune the system. In the prototype it has been used n = 3.

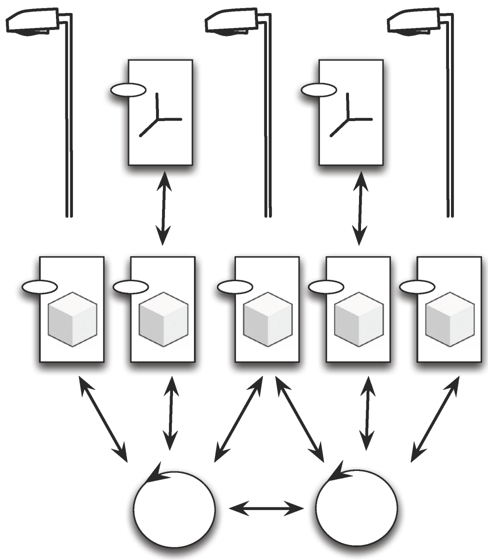


Figure 3.15: Model design for the streetlight system

It is important to underline that the topology of the street map will influence in the minimum number of real sensors needed to have a good enough system performance. For instance, if there is a streetlight close to a crosswalk, this streetlight would need a real smart resource, because it is necessary to detect if the car stops (due to people crossing).

Each *smart resource artifact* detecting a car would inform the agent (or agents) it is linked to that a car is passing and its speed. As the goal of the system is to obtain a lighting similar to the first situation in Fig. 3.14, agents must communicate that a car is approaching and at which speed. Following the example of the design in Fig. 3.15, if a car is detected by the sensor on the left streetlight in the figure (assuming that in streetlights the figure represents real sensors, and *IVE artifacts* represent simulated sensors) it communicates this information to its agent (on the left) along with the car speed. The agent sends the order to this same streetlight and to the next to light at full power, and to the third streetlight to light at half power. The car speed is used to simulate when it is going to arrive at the next streetlight (that does not have sensors). When this occurs, the agent orders the streetlight to light at full power (and to the previous streetlight to turn off) and the agent also sends to the third streetlight the order to change to full power and communicates with the right agent to send it the car position and speed so that this agent can turn on the corresponding streetlight to half power. This process will go on for each car passing along the street. The *Smart Resource Artifact* is in charge of each streetlight and responds to the order that provides the most light if various orders are received simultaneously.

Smart resource artifacts To control lighting in smart cities [80], current systems usually work with basic sensors that detect people or vehicles and send their values directly to an automatic control system that switches on/off the corresponding lights. To improve the data obtained from sensors, it is proposed to work with smart resources that provide processed information. A smart resource connected to various sensors sends the values of the sensors (such as the presence or not of an object) and processes these values to provide richer information. A smart resource could measure the speed of the object detected, its acceleration, size, color, and could even identify the object.

Furthermore, in the case of the detection of vehicles, a smart resource could also report on the number of vehicles detected by measuring the density of traffic. Therefore, light control agents will have more information to adapt the number of streetlights switched on, their duration, and intensity to the traffic circumstances.

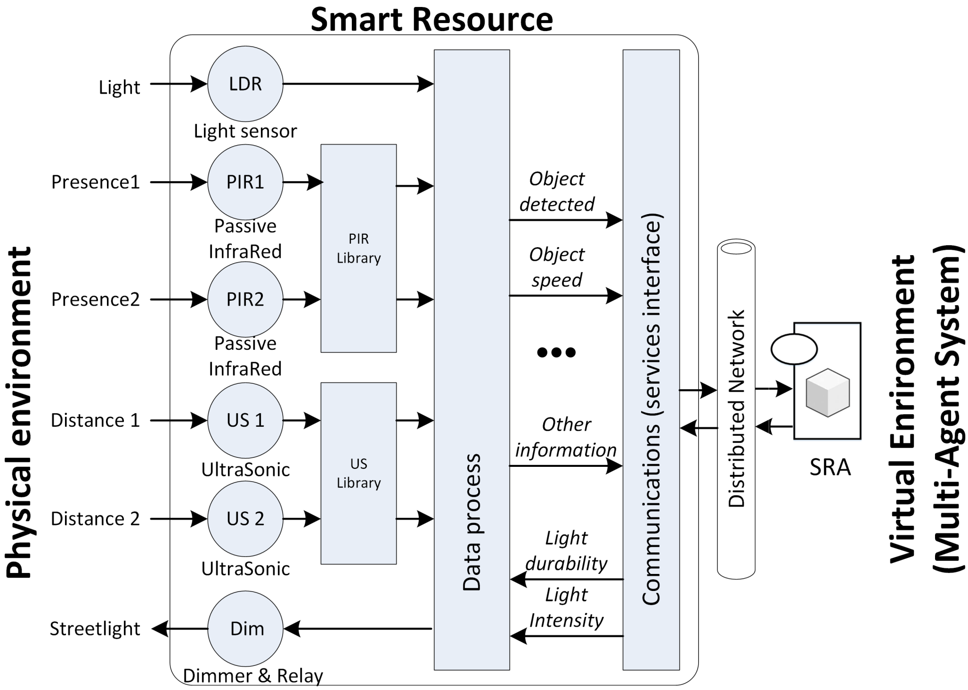


Figure 3.16: Internal components of the speed detector and light control smart resource.

In the experiments presented in this paper it is necessary to provide the speed of the detected objects and control the light intensity of the streetlights. Fig. 3.16 shows the design of the smart resource used.

To detect objects, the smart resource uses a passive infra-red (PIR) sensor. When the PIR detects an object, two ultra-sonic (US) sensors, placed in a row, start to measure the speed. The processing level detects the time when the distance obtained in every US sensor is reduced by the presence of the object. The object speed is then calculated by taking into account the temporal difference. Depending on the resources/services required by the agents, other information, such as object length, may be sent.

Agents control the streetlights connected with the smart resource. Consequently, the smart resource also offers the resources/services: light intensity and durability.

3.4.2 Implementation

Agents and artifacts developed In the developed prototype (see Fig. 3.13), a street with 18 streetlights was controlled. Each of these streetlights is controlled by a smart resource artifact. There are nine agents in the system, each in charge of three smart resource artifacts, taking into account that two agents may share a smart resource artifact as indicated in Fig. 3.15.

These agents and artifacts have been defined according to the Modified $MAM5$ implementation in the $JaCalIVE$ modified framework. The system is defined in an XML file that is compiled to the Jason agents and CARTAGO artifact templates that is filled with the specific code for the system.

To interact with the simulation, a Unity3D render engine have been developed as can be seen in Fig. 3.17.

Figure 3.17: Developed prototype.

Smart resources developed Smart resources are mounted on a waterproof case in order to validate them in external environments. All sensors, actuators, and control components, are placed inside the case (Fig. 3.18). The following components have been used:

- Sensors
  - PIR sensor GH-718, it is an infra-red sensor typically used in smart illumination systems [46].
  - US sensors HC-SR04, used in a wide variety of systems due to its accuracy [93] and speed [114].
  - The light sensors used are LDR (Light Dependent Resistor) with dual operational amplifier (LM358) to increase the accuracy of the signal and adapt it to be used in external environments [72].
- Actuators
  - Relay SRD-05VDC, a classical relay used in control systems [31].
  - Dimmer: a PWM-based LM3405, that was used previously in light control systems [113].
- Control and communications
  - Arduino UNO, one of the most used micro-controller board, based on the ATmega328P and very suitable to implement devices in Agents systems [168])
  - Wifi module, the recently appeared ESP8266 that uses IEEE 802.11 standard protocol, and it is becoming popular including in critical systems [157].

SRAs connect with smart resources through a client/server communications system based on TCP/IP network. This communications system uses wireless connections

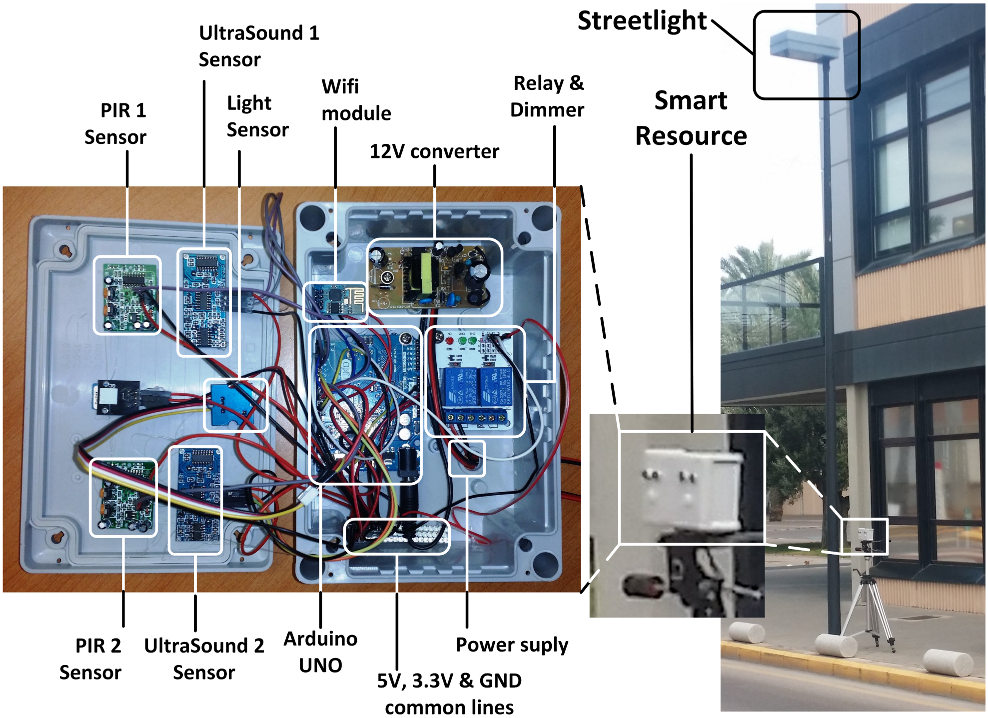


Figure 3.18: Prototype developed of the smart resource.

accomplishing the standard IEEE 802.11b in the 2.4 GHz band and 11 Mbit/s. $JaCalIVE$ Framework includes a TCP server that allows smart resources to be registered according to the previously defined protocol. Besides, every smart resource configures a TCP server that is executed in the Arduino micro-controller in order to attend SRAs requests based on the XML messages described in the protocol.

3.4.3 Execution tests

Different tests have been done in order to evaluate the proposed framework. Specifically, the aim of the tests is to validate the use of smart resource artifacts as a way to model smart resources in the $JaCalIVE$ framework. Moreover, the experiments allow studying the minimum set of real sensors that the proposed system needs to ensure a specific error margin.

In the proposed model, a real sensor in every streetlight won't be necessary if agents can simulate the movements of the cars correctly. For that, the multiagent system in the virtual environment will have to get the positions of the cars by using both virtual and real smart resource artifacts (that is, connected to a simulation or to a real smart resource). Logically, real measurements will come from the real smart resource artifacts, that are connected to available smart resources, and the virtual smart resource artifacts will have to estimate their measurements.

The next graphics show the effect produced in the simulation when the number of real versus virtual smart resource artifacts is increased. The results are obtained when the car moves with different speeds and speed-ups. Every graphic shows the error in the simulation by calculating the difference between the estimated position of the virtual car and the position of the real car when this goes through every streetlight.

To do this, the framework has been tested changing the number and the position of the real sensors.

- Experiment 1: There only exist two real smart resources, they are connected to streetlights number 1 and number 18. The rest of lampposts are controlled by virtual sensors.
- Experiment 2: Streetlights number 1, 9 and 18 are connected to real smart resources. Similar to the previous experiment, the rest of lampposts are controlled by virtual sensors.
- Experiment 3: Streetlights number 1, 6, 12 and 18 are connected to real smart resources. As before, the rest of lampposts are controlled by virtual sensors.

Moreover, each experiment has been evaluated with different behaviors of the drivers:

- Behavior A: the car speed is constant all along the experiment.

- Behavior B: the car initially accelerates and then it maintains its velocity constant.
- Behavior C: the car initially accelerates and then it decelerates its velocity.
- Behavior D: the velocity changes in a random pattern.

Fig. 3.19 shows the results of the Experiment 1 with the four different commented behaviors. As we can see, except from the situation where the driver maintains the same speed, results show a high error prediction done by the virtual sensors. This is logical because they only have information about the speed of the car supplied by the real sensor placed on lamppost 1.

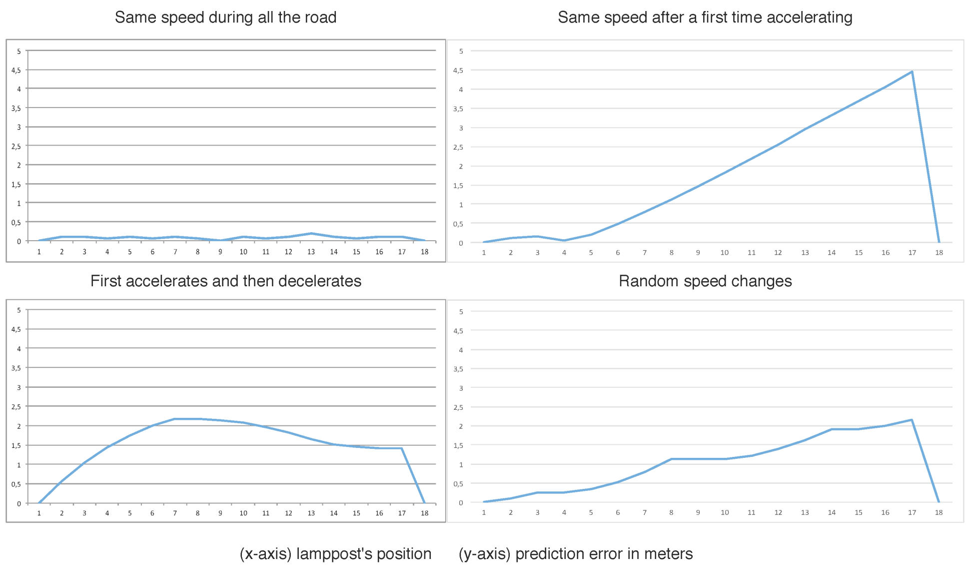


Figure 3.19: First experiment: Smart Resources only in 1st and 18th lampposts

Fig. 3.20 shows the results of the Experiment 2 with the four different commented behaviors. In this situation, the error prediction has been reduced in all the situations. Nevertheless, the error is still high in some situations where the car changes the speed for a longer time.

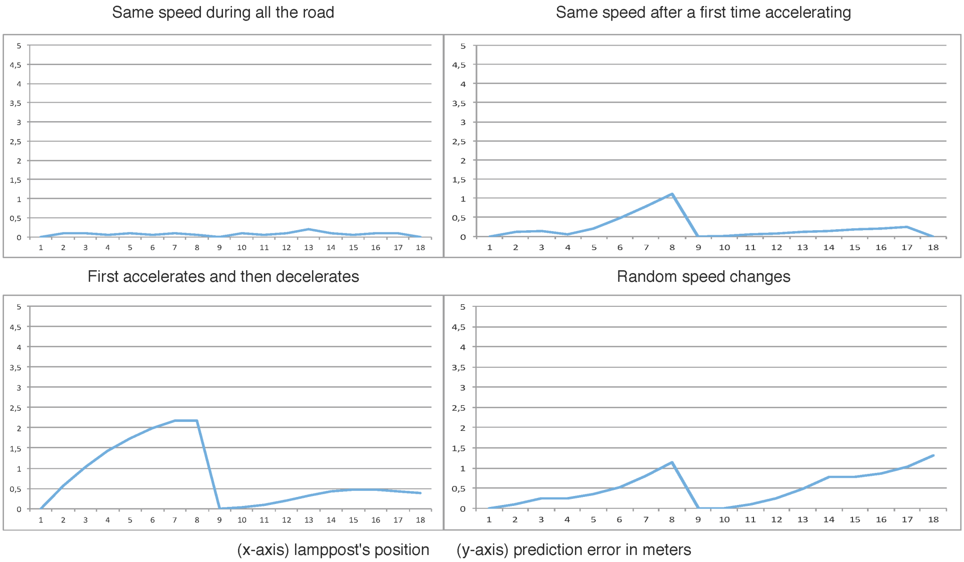


Figure 3.20: Second experiment: Smart Resources in 1st, 9th and 18th lampposts

Fig. 3.21 shows the results of the Experiment 3 with the four different commented behaviors. In this last experiment, we can see how the predictions done by virtual sensors have been improved in all the situations independently of the driver's behavior.

Logically, when the number of real sensors is increased, the error in the simulations is reduced. As it can be seen in Fig. 3.21, with only 4 out of 18 real smart resources the error obtained even in behaviours that are difficult to predict as the third one, the error obtained is admissible to be used as a solution for incremental installations. So, in this case, the automatic streetlight system could be working with an admissible error in 1/4 of the whole time needed for the installation of the whole real smart sensors. It's important to remark that minimizing the error in the prediction of virtual sensors, the system achieves a correct lighting of the road employing the minimum set of real sensors. This is a promising result that leads to go on working in this line of developing.

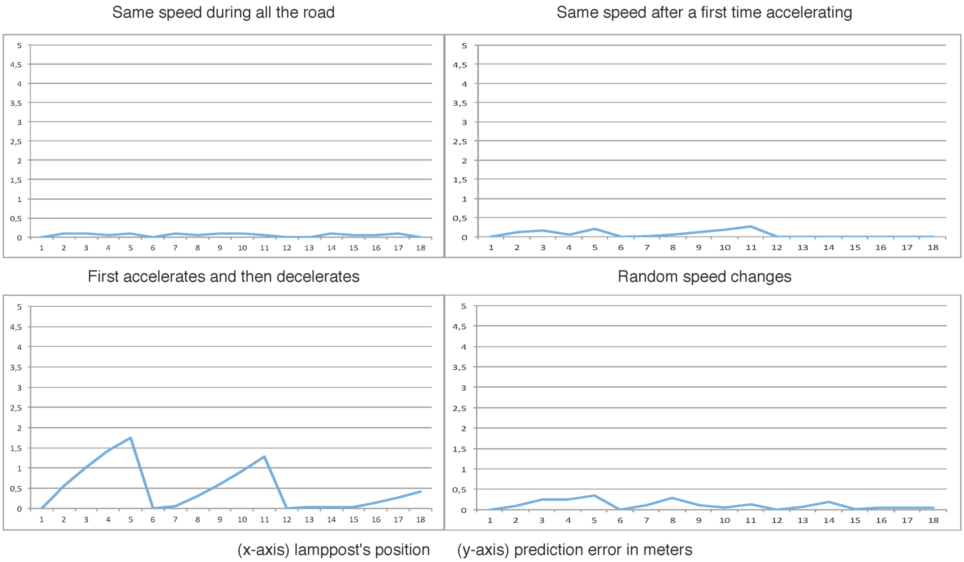


Figure 3.21: Third experiment: Smart Resources in 1st, 6th, 12th and 18th lampposts

Conclusions

A new approach for the design and implementation of mirror worlds has been presented in this paper. This new approach is an extension of the $MAM5$ meta-model and the $JaCalIVE$ framework. The goal of these new extensions is to include the human in the loop and connect (even in real-time conditions) the physical environment with the virtual world - and so enable the creation of augmented worlds. Specifically, the aim was the aggregation of new components such as *smart resources artifacts* and *human immersed agents* in the development process of these intelligent virtual worlds, and help designers create *mirror worlds*.

The integration of these elements represents a great advantage because it facilites the design of models that can capture the real world with a high level of abstraction, and so enable the designer to create a representation of the real world that facilitates the introduction into these models of components based on AI and MAS technology.

Apart from providing a new method to develop this type of intelligent virtual world, the proposal enables the execution of the system through the use of a supporting platform.

Finally, an application example of the proposed framework was presented. This example is related with the efficient and intelligent use of streetlights. The example has been used in a satisfactory way to obtain a suitable prototype using the proposed models and execution framework.

The method presented can be used to determinate the minimum set of real sensors that a system needs to ensure a determinate error margin.

As a future work, the aim is to continue developing more complex examples, integrating other sensors or actuators with new utilities that produce more information about the environment and enable the control of complex physical devices. This framework can be applied to many different domains. Some of them, that we would like to underline, include big systems simulations in ambient intelligence to predict the energy saving reduction in smart cities and smart mobility. Moreover, applications in the smart health domain where agents could be used to monitor bio-signals, falling detection, and so on. This monitorization could be done not only by software agents but also with mobile robots. In the robotics domain, this framework can be used to model and simulate the real environment using smart resources to improve the accuracy of the robot navigation.

CHAPTER

4

Influencing over People with a Social Emotional Model

AUTHORS:
J. A. RINCON, F. DE LA PRIETA, D. ZANARDINI, V. JULIAN AND C. CARRASCOSA
{jrincon, mgarcia, vinglada, carrasco}@dsic.upv.es
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN
UNIVERSIDAD POLITÉCNICA DE VALENCIA
CNO/ DE VERA SN
46022 VALENCIA, SPAIN
DEPARTMENT OF COMPUTER SCIENCE - UNIVERSITY OF SALAMANCA
FER@USAL.ES
DEPARTAMENTO DE INTELIGENCIA ARTIFICIAL, UNIVERSIDAD POLITECNICA DE MADRID, SPAIN
DAMIANO@FI.UPM.ES

4.1 Abstract

This paper presents an approach of a social emotional model, which allows to extract the social emotion of a group of intelligent entities. The emotional model *PAD* allows to represent the emotion of an intelligent entity in 3-D space, allowing the representation of different emotional states. The social emotional model presented in this paper uses individual emotions of each one of the entities, which are represented in the emotional space *PAD*. Using a social emotional model within intelligent entities allows the creation of more real simulations, in which emotional states can influence decision-making. The result of this social emotional mode is represented by a series of examples, which are intended to represent a number of situations in which the emotions of each individual modify the emotion of the group. Moreover, the paper introduces an example which employs the proposed model in order to learn and predict future actions trying to influence in the social emotion of a group of people.

4.2 Introduction

In the 80s decade, Human-Computer Interaction (HCI) appeared as a new field giving access to the new digital technologies and converting all the people in potential users without any knowledge about computers. During last decades, HCI has involved information interchange between people and computers using some kind of dialogue, like programming languages and information interchange platforms. These platforms have included from input devices such as keyboards and optical mouses to output devices as the own computer screens.

Lastly, cognitive psychology integration within the HCI field leads to adopt new forms of information processing and to better understanding how people communicate with the devices. Nevertheless, in spite of the accessibility solutions presented by HCIs, user interfaces were very limited. As a result, the discipline has adopted other research subjects focused in usability, ergonomics trying to build new interfaces and allowing a more natural interaction between humans and machines.

These research subjects have made appear new interaction paradigms created by the mobile computing, portable and ubiquitous. They have incorporated devices to communicate directly with the physical world such as movement and gestures capture through the *Kinect* [161] and even user biosignals capture through the *MYO* and *Emotiv* devices [24], [117]. The idea is that machines will not only receive orders from users but also they will perceive their emotional states or behaviors using all this information to execute the different actions [107], [97].

The information increase generated by the new ways of interaction has made appear the need of using other computational toolkits to analyse and process information to benefit users. Artificial Intelligence tools such as pattern recognition ones, machine learning, and multi-agent systems (MAS) allow the development of this kind of complex tasks, creating adaptive environments to human needs to improve his welfare and life quality.

Human beings manage themselves in different environments, either in the working place, at home or in public places. At each one of these places we perceive a wide range of stimuli, that interfere in our commodity levels modifying our emotional levels. For instance, the high levels of noise or the temperature conditions may produce stress situations. Before each one of these stimuli, humans answer varying our face gestures, body or bio-electrical ones. These variations in our emotional states could be used as information useful for machines. Nevertheless, it is needed that the machines will have the capability of interpreting or recognizing such variations. This is the reason for implementing emotional models to represent the different emotions.

Emotional models such as *OCC* [111] presented by *Ortony, Clore & Collins* and the *PAD* model [99] are the most used ones to detect or simulate emotional states. Nevertheless, these models don't allow the execution of intelligent decisions based on the emotional state perception. Between these toolkits, we can find MAS, which are able to modify their behavior based on the emotional state perception. This way, it is obtained that the agent being part of the MAS contains an emotional model able of interpreting and/or emulating different emotional states. To detect emotional states, it is needed to include pattern recognition algorithms, automatic learning contributing to the decision making to execute an action. For instance, if an agent detects that the user presents an emotional state of sadness, it is able to counter that emotional state by executing actions trying to modify it. This way a clean and transparent human-machine interaction is obtained. However, this situation is only valid for a lonely entity inside the environment. The incorporation of more entities inside the environment (multiple emotions) is not contemplated by current emotional models.

The goal of this work is to give an approach to a social emotional model including multiple emotions between humans and agents. Our model uses as base the *PAD* emotional model to represent the social emotion of a group. Moreover, the paper introduces a case study where the social emotion is used for predicting next actions in order to improve the emotional state of a group of people. Concretely, the case study has been developed simulating a bar, where there is a DJ in charge of playing music and a specific number of persons listening that music. The main goal of the DJ is to play music making that all the people within the bar are mostly as happy as possible. This proposed application engages *Ambient Intelligence (AmI)* and *Ubiquitous Computing (UC)* involving humans and helping to improve their living conditions. This kind of applications involves the interaction between humans and agents, being these responsible for a continuous monitoring of the different emotional states. This is mainly due to the influence that music can have on people's moods. This influence has already been studied [172] analyzing how different musical genres can influence people's emotions. Existing works take into account the social and cultural importance of music which influences positively or negatively on people behavior [124],[159], [38].

The rest of the paper is structured as follows: Section 2 introduces related work; Section 3 presents the social emotional model; Section 4 describes the case study introducing some illustrative situations and analysing the obtained results; finally, Section 5 includes some conclusions.

4.3 Previous approaches

This section presents an introduction to the emotional models *OCC* and *PAD*. The goal is to give a general view of both emotional models.

4.3.1 Ortony, Clore & Collins: OCC

The *OCC* model designed by Ortony, Clore & Collins is a model frequently used in applications where an emotional state can be detected or simulated. This has allowed to create applications to emulate emotions in virtual humans [17] and to create agents reacting to stress situations [67].

The *OCC* model specifies 22 emotional categories, which are divided into five processes: 1) the classification of the events, the action or the found object, 2) the quantification of the affected emotions intensity, 3) the interaction between the just generated emotion with the existing ones, 4) the cartography of the emotional state of one emotional expression and 5) is the one expressed by the emotional state [112]. In *OCC* model is observed. These processes define the whole system, where the emotional states represent the way of perceiving our environment (objects, persons, places) and, at the same time, influencing in our behaviour positively or negatively [4]. However, the *OCC* model utilization presents one complication due mainly to his high dimensionality.

4.3.2 PAD Model

The *PAD*[100] is a simplified model of the *OCC* model, since the *OCC* model represents the emotion using eleven dimensions whilst the *PAD* only uses three. This reduction of dimensionality allows to do mathematical operations faster and to represent the emotion as a vector in the $\mathbb{R}^3$ space. In this representation, the values that compose the emotion are usually normalized in the range $[-1, 1]$, and correspond to the three components conforming the emotional model (*Pleasure, Arousal, Dominance*). Each one of these components allow to influence over the emotional state of an individual in a positive or negative way. This influence evaluates the emotional predisposition of such individual, modifying in this way his emotional state. The Pleasure-Displeasure Scale measures how pleasant an emotion may be. For instance both anger and fear are unpleasant emotions, and score high on the displeasure scale. However joy is a pleasant emotion. This dimension is usually limited to 16 specific values. ([98], pp. 39–53). The Arousal-Nonarousal Scale measures the intensity of the emotion. For instance while both anger and rage are unpleasant emotions, rage has a higher intensity or a higher arousal state. However boredom, which is also an unpleasant state, has a low arousal value. This scale is usually restricted to 9 specific values([98], pp. 39–53).The Dominance-Submissiveness Scale represents the controlling and dominant nature of the emotion. For instance while both fear and anger are unpleasant emotions, anger is a dominant emotion, while fear is a submissive emotion. As have been presented above, the existing emotional models are thought to detect and/or simulate human emotions for a lonely entity. That is, it is not taken into account the possibility of having multiple emotions inside an heterogeneous group of entities, where each one of such entities have the capability of detecting and/or emulating one emotion. The need of detecting the emotion of an heterogeneous group of entities can be reflected in the different applications that could be obtained. With the appearance of the different smart devices, ubiquituous computation and ambient intelligent, emotional states turn into valuable information, allowing to develop applications that help to improve the human being life quality. Therefore, it is needed to create a new model that allows to detect the emotion of a group.

4.4 Social Emotional Model Based on the PAD model

This section proposes a model of social emotion based on the *PAD* emotional model. This model will represent the social emotion of a heterogeneous group of entities capable of expressing and/or communicate emotions. To define a model of social emotion, it is necessary first to define the representation of an emotional state of an agent on the *PAD* model. The emotion of an agent ag_i is defined as a vector in a space $\mathbb{R}^3$, represented by three components that make up the *PAD* emotional model: *P=Pleasure*, *A=Arousal* and *D=Dominance*. Each one of these elements allow to represent the emotion of each agent ag_i. The variation of each component allows to modify the emotional state of the agent (Equation 4.1).

$$\vec{E}(ag_i) = (P_i, A_i, D_i) \tag{4.1}$$

A first approach to a social emotion representation of a group of n agents $Ag = \{ag_1, ag_2, ..., ag_n\}$ is obtained by averaging their P, A, D values (Equation 4.2). This average will enable us to determine where the central emotion (*CE*) of this group of agents and be visualized in the *PAD* space.

$$\bar{P} = \frac{\sum_{i=1}^{n} P_i}{n}, \bar{A} = \frac{\sum_{i=1}^{n} A_i}{n}, \bar{D} = \frac{\sum_{i=1}^{n} D_i}{n} \tag{4.2}$$

The final result is a vector in the space $\mathbb{R}^3$ which is the core emotion or $CE(Ag)$ of a group of agents (Equation 4.3).

$$\vec{CE}(Ag) = \left[\bar{P}, \bar{A}, \bar{D}\right] \tag{4.3}$$

The $CE(Ag)$ by itself is not enough to represent the social emotion of a group of agents, since there may be different groups of agents with the same central emotion but in a very different emotional situation. Figures 4.1 and 4.2 [1], show two different

[1] This is a graphical representation of the emotional states of a group of agents according to the values

situations where the central emotion is the same. In Figure 4.1, the emotional states of a group of agents is observed with $CE(Ag) = [0.0, 0.22, 0.45]$. As can be observed, the individual emotional states of these agents are subdivided into two very different groups. In fact, they may represent, for instance, a group of people which are seeing a football match that has just finished, having supporters from the two teams. On the other hand, Figure 4.2 shows the individual emotional states of another group of agents with completely different emotions. Although it can also be observed two different groups, their emotions are not as oppossed as in the previous figure. Nevertheless, as before commented, the two examples represented in Figure 4.1 and Figure 4.2 generate the same central emotion.

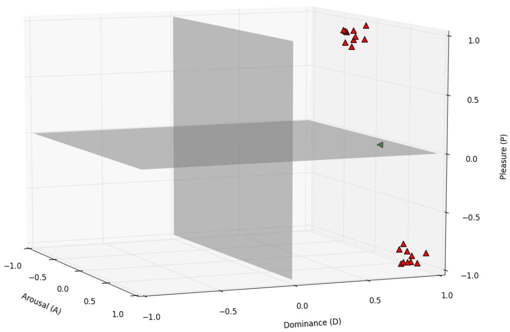


Figure 4.1: Representation of a group of agent's emotions with two subgroups completely opposite.

Clearly, the $CE(Ag)$ is not enough to represent the social emotion of an agent group. As it can be seen in the previous example the emotions of an agents group can be very different but have the same CE. This is why it is necessary to introduce some measurement about the distance of the agents with respect to the CE. To do this we include the definition of the maximum distances of agent emotions respect to $CE(Ag)$. In order to calculate the maximum distances, the Euclidean distance (Equation 4.4, 4.5, 6.10) is used as follows.

of the PAD model in a space $\mathbb{R}^3$. Each red triangle represents the emotional state of an individual agent and the green triangle represents the central emotion $CE(Ag)$ of the group of agents.

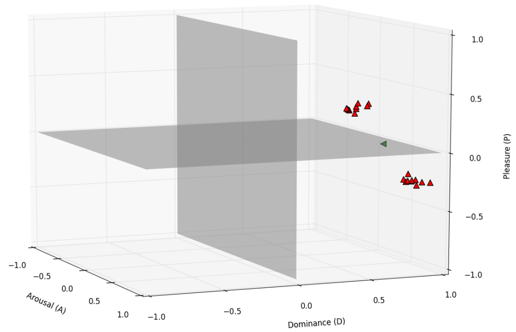


Figure 4.2: Representation of a group of agent's emotions with two subgroups with more nearest emotions.

$$m_P(Ag) = max\left(\sqrt{(P_i - \bar{P}(Ag))^2}\right), \forall ag_i \in Ag/E(ag_i) = [P_i, A_i, D_i] \tag{4.4}$$

$$m_A(Ag) = max\left(\sqrt{(A_i - \bar{A}(Ag))^2}\right), \forall ag_i \in Ag/E(ag_i) = [P_i, A_i, D_i] \tag{4.5}$$

$$m_D(Ag) = max\left(\sqrt{(D_i - \bar{D}(Ag))^2}\right), \forall ag_i \in Ag/E(ag_i) = [P_i, A_i, D_i] \tag{4.6}$$

The results of these equations can be represented as a tor of maximum distances (Equation 4.7).

$$m(Ag) = [m_P(Ag), m_A(Ag), m_D(Ag)] \tag{4.7}$$

The $m(Ag)$ can indicate if there exist agents having their emotional state far away from the central emotion. From a graphical perspective it is also possible to use these maximum distances to plot an enveloping which encapsulates all emotions, allowing the limit of all the agents to be defined. To represent this enveloping shape of emotions an ellipsoid as a geometric figure was used. This ellipsoid (Figure 4.3)[2] can be adapted to represent different emotional states, which allows a dynamical way for displaying the social emotion of a group.

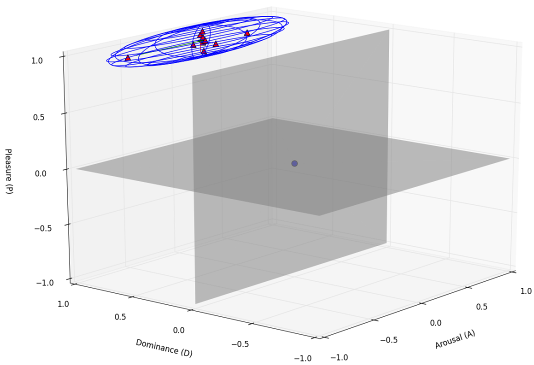


Figure 4.3: Representation of an ellipsoid enveloping the emotions of a group of agents.

Furthermore, considering $m(Ag)$ as a part of the definition of the social emotion of a group of agents, there may be situations in which $m(Ag)$ is not enough. In Figure 4.4 and 4.5 a group of agents is shown with similar $CE(Ag)$ and $m(Ag)$, but with completely different emotional situations. In order to solve this problem the notion of standard deviation (SD) is introduced. This SD allows the calculation of the level of emotional dispersion of this group of agents around the central emotion $CE(Ag)$ for each component of the *PAD*(Equation 6.14).

[2]This figure is an snapshot of the emotion of a group of agents in an specific time

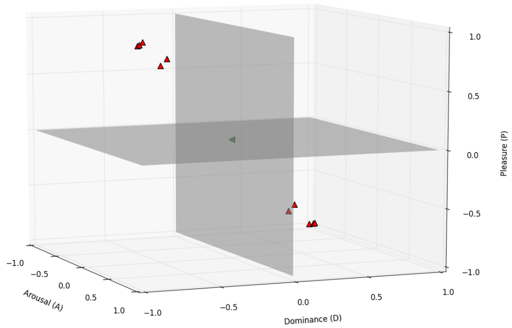


Figure 4.4: Representation of a group of agent's emotions with two very different subgroups .

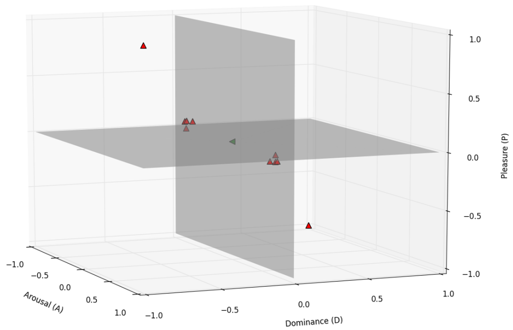


Figure 4.5: Representation of a group of agent's emotions with two rather different subgroups.

$$
\begin{aligned}
\sigma_P(Ag) &= \sqrt{\frac{\sum_{i=1}^{n}(P_i - \bar{P}(Ag))^2}{n}}, \forall ag_i \in Ag/E(ag_i) = [P_i, A_i, D_i] \\
\sigma_A(Ag) &= \sqrt{\frac{\sum_{i=1}^{n}(A_i - \bar{A}(Ag))^2}{n}}, \forall ag_i \in Ag/E(ag_i) = [P_i, A_i, D_i] \\
\sigma_D(Ag) &= \sqrt{\frac{\sum_{i=1}^{n}(D_i - \bar{D}(Ag))^2}{n}}, \forall ag_i \in Ag/E(ag_i) = [P_i, A_i, D_i]
\end{aligned}
\tag{4.8}
$$

The result of each of the above equations can be represented as a tor (Equation 4.9), which allows to determine the level of emotional dispersion.

$$\sigma(Ag) = [\sigma_P(Ag), \sigma_A(Ag), \sigma_D(Ag)] \tag{4.9}$$

The maximum value that can be represented in each one of the components of the tor $\sigma(Ag)$ is 2. Therefore, in our case, we consider that the emotional dispersion is high when the $|\sigma(Ag)|$ is greater than 2, from this definition it can be deduced that:

1. **if** $|\sigma(Ag)| >> 2$, the group has a high emotional dispersion, i.e. the members of the group have different emotional states.
2. **if** $|\sigma(Ag)| \cong 0$, the group has a low emotional dispersion, this means that individuals have similar emotional states.

Adding the emotional dispersion in the definition of the social emotion of a group of agents, the social emotion of a group of agents $Ag = ag_1, ag_2, ..., ag_n$ can be defined by the following triplet (Equation 4.10).

$$SE(Ag) = (CE(Ag), m(Ag), \sigma(Ag)) \tag{4.10}$$

Where $CE(Ag)$ is the central emotion, $m(Ag)$ represents the maximum distances and $\sigma(Ag)$ represents the emotion dispersion of an agent group.

Based on this model it is possible to determine the emotional distance among different groups of agents or between the same group in different instants of time. This will allow to measure the emotional distance between the current social emotional group and a possible emotional target. This approach can be used as a feedback in the decision making process in order to take actions to try to move the social emotion to a particular area of the *PAD* space or to allow that the emotional state of a group of agents can be approached or moved away from other groups of agents. From an emotional point, of view these movements or actions are domain-dependent and are out of the scope of this model. In Equation 4.11 the profile of the emotional distance function is defined as the distance of the social emotions of two groups of agents.

$$\Delta_{SE} : SE(Ag^i), SE(Ag^j) \rightarrow [0,1] \tag{4.11}$$

According to this profile, Equation 6.17 shows how we calculate this emotional variation. The equation calculates three distances corresponding to the three components of the *SE*. Given two groups of agents Ag_i, Ag_j with social emotions $SE(Ag^i), SE(Ag^j)$ respectively, the emotional distance between these two groups is calculated as:

$$\begin{aligned}\Delta_{SE}(SE(Ag^i), SE(Ag^j)) = \frac{1}{2}\Big(&\omega_c\Delta(CE(Ag^i), CE(Ag^j))\\ &+\omega_d\Delta(m(Ag^i), m(Ag^j))\\ &+\omega_v\Delta(\sigma(Ag^i), \sigma(Ag^j))\Big)\end{aligned} \tag{4.12}$$

$$where \quad \omega_c + \omega_d + \omega_v = 1; \quad \omega_c, \omega_d, \omega_v \in [0,1] \tag{4.13}$$

and Δ calculates the torial distance between two tors. As every dimension of the *PAD* space is bounded between $[-1,1]$, each Δ will give values between $[0,2]$. Therefore, Δ_{SE} will have a range between $[0,1]$.

Calculating the distance among social emotions allows the study of the behavior of emotional-based agents, either minimizing or maximizing the $\Delta_{SE}(SE(Ag^i), SE(Ag^j))$ function. This way, it can be achieved that an agent group approaches or move away of an specific emotional state. To do this it is necessary to modify through stimuli the individual emotions from each agent and therefore changing the social emotion. Nevertheless, how to maximize or minimize the emotional distance is domain-dependent and it is out of the scope of this paper.

4.5 Case Study

A practical application which uses the previously proposed model is presented in this section. This application example is based on how music can influence in a positive or negative way over emotional states [172], [166], [150].

The application example is developed in a bar, where there is a DJ agent in charge of playing music and a specific number of individuals listening the music. The main goal of the DJ is to play music making that all individuals within the bar are mostly as happy as possible. Each of the individuals will be represented by an agent, which has an emotional response according to its musical taste. That is, depending on the musical genre of the song, agents will respond varying their emotional state. Moreover, varying emotions of each agent will modify the social emotion of the group.

In such a way, the proposed application seeks to identify the different emotional states using them as a tool of communication between humans and agents. To perform this detection we need to use pattern recognition algorithms and image and audio processing techniques in order to detect and classify the different emotional states of humans. The application has been developed as a virtual multi-agent system using the $JaCalIVE$ framework [134] where there will be different entities. Each one of these entities may represent not only real or simulated human beings, and the DJ agent, but also the furniture or the speakers with their location. The application is composed of two types of entities, a DJ agent which is the responsible of playing the music and different agents that represent humans who are inside the bar (see Figure 5.3). The main tasks of the different type of agents are:

- DJ agent: the main goal of this agent is to achieve a emotional state of happiness for all of the people which are in the bar. When the DJ agent plays a song, it must analyze the emotional state of people. According to this analysis it will select the most appropriated songs in order to improve, if possible, the current emotional state of the audience.

- Human-immersed agent: it is in charge of detecting and calculating the emotional state of an individual which is in the bar, sending this information to the DJ agent. In order to accomplish its tasks, this agent must have access to a variety of input/output information devices as cameras, microphones, ...

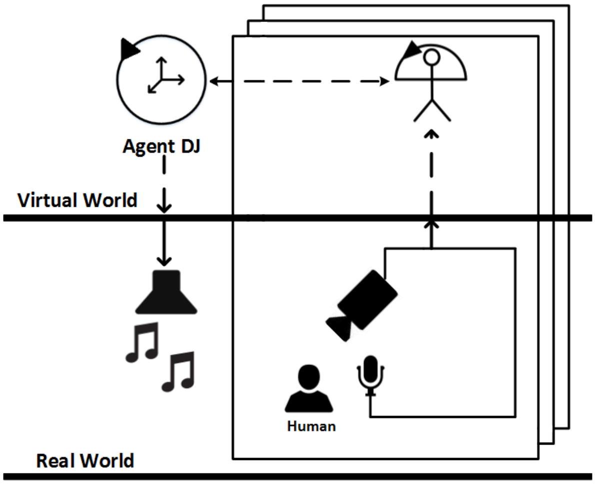


Figure 4.6: Simple scheme of the application

4.5.1 Analysis of possible scenarios

Two different scenarios have been designed in order to illustrate how the social emotion can facilitate the decision making of the DJ. In each scenario the DJ agent plays a song. Once the song has ended, the DJ evaluates the social emotion of the group of listeners that are within the bar. In this way, the DJ agent can evaluate the effect that the song has had the song over the audience. This will help the DJ to decide whether to continue with the same musical genre or not in order to improve the emotional state of the group.

4.5.1.1 Scenario 1: Group of agents with a low emotional dispersion

The first case analyzed is one in which the emotional states of the agents are close. This emotional difference may be due mainly because the agents have little differences in their musical tastes. The social emotion in this scenario has a $EC(Ag)$ very close to all the values of the agents and the $m(Ag)$ and $\sigma(Ag)$ values will be very small and in many cases close to zero. This provokes that the DJ will try to play songs of similar generes trying to maintain this situation, which is not the ideal situation but it can be considered as a very good situation. A graphical representation of this example can be seen in Figure 4.7 while Table 4.1 shows the different emotional states of each of the agents in this group.

Table 4.1: Individual emotion of each agent and its magnitude in the PAD space

Agents	P	A	D	Emotional State
ag_0	0.90	0.0	0.90	Happy
ag_1	0.70	0.0	0.91	Happy
ag_2	0.80	0.0	0.95	Happy
ag_3	0.85	0.0	0.99	Happy
ag_4	0.91	0.0	0.89	Happy
ag_5	0.93	0.0	0.86	Happy
ag_6	0.89	0.0	0.83	Happy
ag_7	0.79	0.0	0.81	Happy
ag_8	0.92	0.0	0.89	Happy
ag_9	0.81	0.0	1.0	Happy

As it can be see in the Figure 4.7 all the represented emotions in this group are around the emotion *Happy*, achieving a social emotion with these values of $SE(Ag) = ([0.85, 0.0, 0.9], [0.85, 0.0, 0.9], [0.07, 0.0, 0.06])$.

4.5.1.2 Scenario 2: Group of agents with high emotional dispersion

In this second case it is represented the existence of a group of agents emotionally dispersed in the bar. These agents have completely different emotions distributed along the *PAD* space. The emotional values of each of the agents can be seen in Table 4.2. This high dispersion is reflected in the calculated values of the social

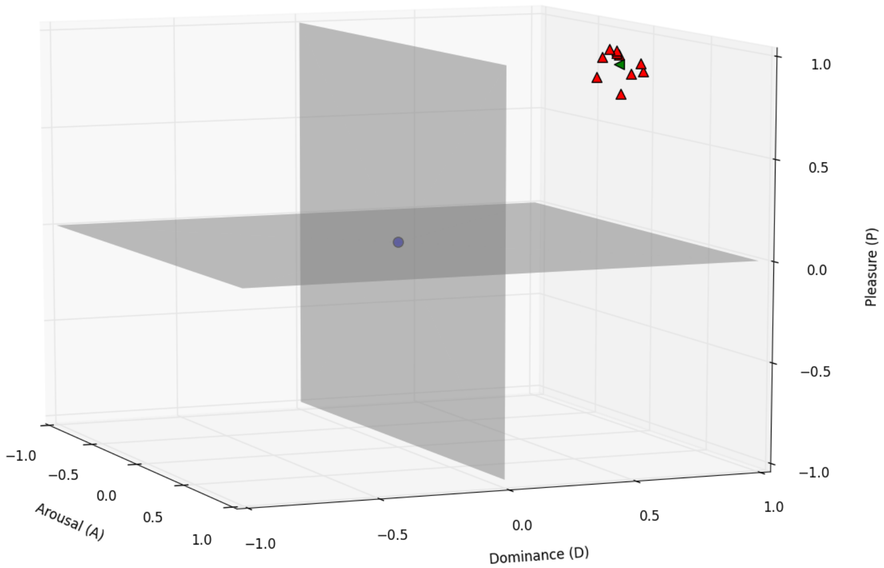


Figure 4.7: Scenario 1: Representation of the emotional states of a group of agents with a low emotional dispersion.

emotion:

$$SE(Ag) = ([0.14, 0.42, 0.1], [1.14, 1.32, 1.0], [0.87, 0.67, 0.79])$$

In this case, the social emotion is very different and more complicated to manage than in the previous case. The central emotion is very far from the emotional states of each agent and the maximum distances and dispersion values are high too. So, from the perspective of the DJ this scenario is very chaotic and unwished because it is difficult to choose which kind of music is the most appropriated. In this case the DJ should try to move the central emotion to a state close to ”happy” testing different musical styles and analyzing carefully the effect of each song in the social emotion of the group.

Table 4.2: Individual emotion of each agent and its magnitude in the PAD space

Agents	P	A	D	Emotional State
ag_0	-0.9	-0.9	-0.9	Remorse
ag_1	-0.7	0.6	0.0	Anguish
ag_2	0.9	0.9	0.0	Joy
ag_3	0.9	-0.5	0.9	Satisfaction
ag_4	-0.7	0.8	-0.9	Hurt
ag_5	0.9	0.9	0.9	Admiration
ag_6	0.9	0.0	0.9	Happy
ag_7	-1.0	1.0	0.9	Anger
ag_8	1.0	1.0	-0.9	Love

4.5.2 Improving by learning of the DJ agent

The above commented application is formed by two different types of agents: the DJ Agent and the Human-Immersed Agent. Regarding the DJ Agent, this agent uses the information sent by the Human-Immersed Agent to analyze the group's emotional state, using it to decide which is the following song to play. Obviously, the goal of the DJ Agent is that all the humans feel as happy as possible. To achieve this, each of the agents representing humans should communicate previously their emotional state. For doing this, it is necessary to provide each Human-Immersed Agent with a serie of sensors which generate enough information to be able to know the emotional state of each human. After this, the DJ agent calculates the social emotion and decides the next song. As this decision will affect in the emotional state of people, it is necessary to include some kind of mechanism to facilitate this decision.

The mechanics selected to give the *DJ* agent the capability of learning and predicting the new social emotion was a neural network (*ANN*). Since it not only has the capability of classifying but it also has the ability to predict[108]. This ability is used to predict the social emotion caused by an specific song played in the bar. The main process is defined as follows: 1) the social emotion of the group is calculated using the information provided by the Human-Immersed Agents; 2) An ANN is used to predict the future social emotion of people taking into account the previously calculated current social emotion and a set of songs, which are similar to the previously played song (this is done searching by genre in the database of songs);

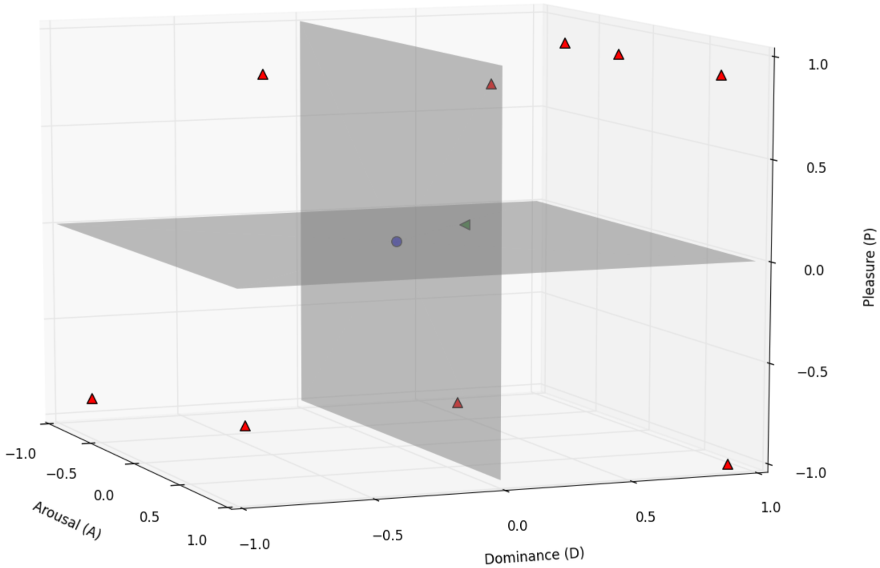


Figure 4.8: Scenario 2: Representation of the emotional states of a group of agents with a high emotional dispersion.

3) the song which minimizes the distance between the target emotion (happiness) and the predicted social emotion is selected; and 4) the selected song is played in the bar and the process begins again. A graphical view of the application process is shown in Figure 4.9.

The *ANN*'s architecture used is a *Background Propagation* as it is the most used and gives the best results for making predictions [105]. Figure 4.10 shows the architecture of the proposed *ANN*, which is composed of: *10 input neurons, 5 neurons in the middle layer and 9 output neurons*. The ten inputs of the neural network are the current social emotion and a musical genre. The output layer is the social emotion predicted by the *ANN* for the group of agents.

Different sessions were created in order to train our *ANN* with different personalities

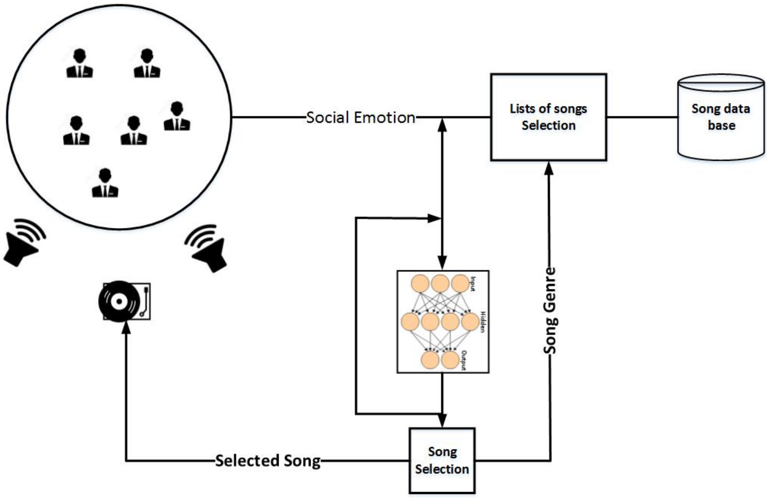


Figure 4.9: Predictive System Scenario

and musical preferences of the audience. The results of these simulations were stored in order to create a dataset. For each iteration, the previous social emotion, the played song, the musical genre and the social emotion calculated after the song were stored. Once the simulations were completed, the generated data was used for the training process of the proposed *ANN*.

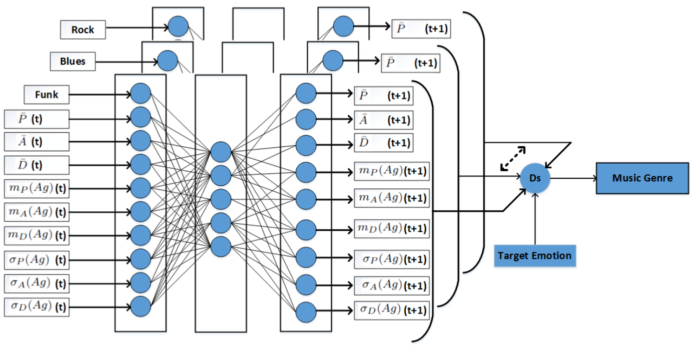


Figure 4.10: Neural Network Architecture

The *ANN* was trained using a supervised trained methodology, since the objective of the network is to predict the future value of the social emotion after a specific song is played. Concretely, the training process employed a dataset composed by *3000* entries. This information was extracted from 5 simulated sessions. In each one of the sessions, the parameters of each agent were modified changing their personality values (defined according to the OCEAN Model [47]) and their musical preferences. From the obtained dataset, the 25% was used to test and the 75% was used to train. Figure 4.11 shows the generated main square error in each iteration of the training process.

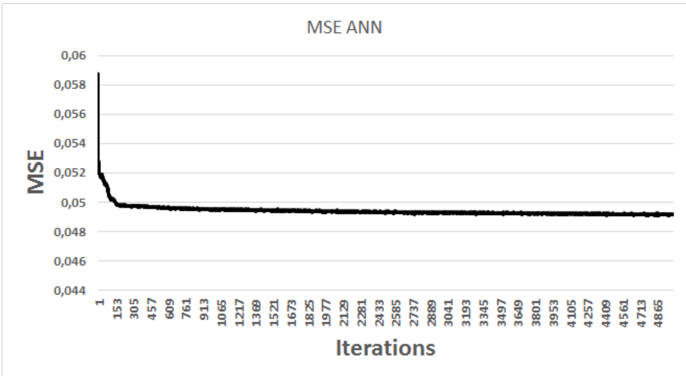


Figure 4.11: Evolution of the Mean Square Error (MSE) of the Neural Network

Moreover, the results generated by the proposed ANN have been compared with two simple methods. The first one considers that the predicted social emotion is the same than the previous one. The second one predicts as the new social emotion the average of the last ten social emotions calculated. Table 4.3 summarizes the obtained results for each method, showing the *MSE* calculated individually for each PAD value. These *MSE* values calculated in the three experiment allow us to compare which of the three experiment gives the best results.

As can be seen, the ANN achieves better results with lower *MSE* values for each component of the *PAD* vector.

Experiment	MSE P	MSE A	MSE D
ANN	0.006337	0.0082748	0.01309546
Simple 1	0.0503843	0.0086129	0.5056521
Simple 2	0.0463723	0.0084158	0.5057954

Table 4.3: Mean Square Error obtained for each PAD component

In order to analyze the effect of the ANN in the proposed application, different experiments have been developed. Specifically, the aim of these tests is to validate the use of the ANN as a way to predict the social emotion of a group of agents and, thus, improve the decision making process. To do this, the implemented prototype has been tested changing the music played in the bar (around 50 songs are played in an iterative way) and each test has been repeated 100 times. All the tests are composed by agents representing people which is in the bar. These agents have assigned random initial emotions at the beginning of the execution of each test.

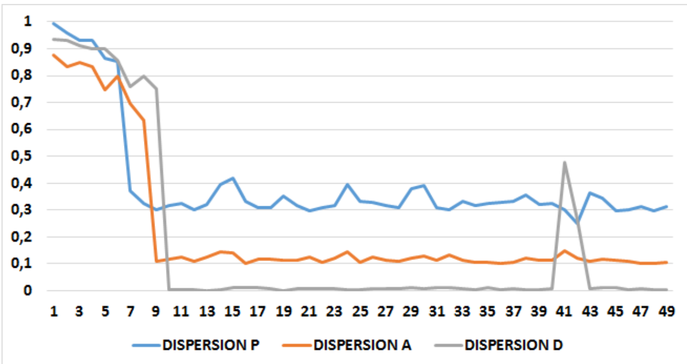


Figure 4.12: Evaluation of the maximum dispersion of the social emotion for each PAD component (without the ANN)

Figures 4.12 and 4.13 represent the evolution of the dispersion around the group of agents' social emotion in two different configurations of the system. In the first one, the songs played are randomly selected among the set of songs of the same genre. In the second one, the application selects the songs to be played using the previously exposed process including the proposed ANN. The dispersions showed in

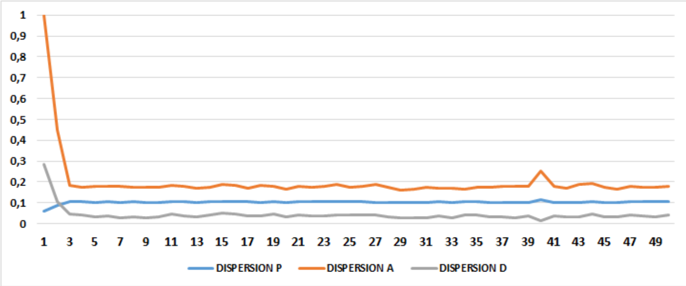


Figure 4.13: Evaluation of the maximum dispersion of the social emotion for each PAD component (with the ANN)

the Figures are calculated separately for each component of the PAD vector. As it can be observed, as time passes the dispersion is lower in the case of the experiment which includes the ANN as a prediction tool. This fact is especially remarkable in the case of the Pleasure (P) component, because it is the most related component regarding happiness. The achieved dispersion in the Pleasure component is extremely lower when we employ the ANN.

Finally, Figure 4.14 shows the results obtained regarding the distance between the social emotion of the group and the target emotion (Happy). As we can see, the system quickly reduces the distance of the social emotion with regard to the target emotion. After few iterations, the distance remains constant around 0.5, which can be considered a very good and close approximation.

4.6 Conclusions and future work

A new model for representing social emotions has been presented in this paper. The goal of this model is to give a first approach for the detection and simulation of social emotions in a group of intelligent entities. This social emotion model builds on the *PAD* emotional model, which allows the representation of individual emotions in intelligent entities. The proposed model of social emotion uses the individual

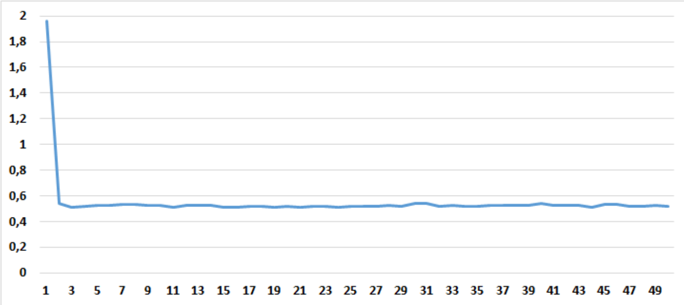


Figure 4.14: Evaluation of the distance between the social emotion against happiness

emotions of each entity of a group, allowing us to represent the emotion of that group as a triplet consisting of three tors $(EC(Ag), m(Ag)\ and\ \sigma(Ag))$. This definition allows us to represent the emotional state of a group of entities that are placed in a specific environment. Moreover, the model adds the mechanisms to compare the social emotional state of two groups of agents or the social emotion of a group in different time instants. The social emotion of a group of agents not only allows a global view of the emotional situation of the group, moreover it can be used as a feedback in order to change the emotional state of the group or only of a part of the agents.

The paper also presents an application able to extract (in a non-invasive way) and to analyze the social emotion of a group of persons and it takes decisions according to that emotional state. The application includes a learning module which will allow to predict future emotional states of the people. According to the obtained results, this module improves the decision making of system comparing the social emotional values that are obtained during the execution of the system.

The proposed model can be also applied to many different domains. Some of them, that we would like to underline, include big systems simulations in ambient intelligence in order to predict human behavior, even in industrial environments, where it can monitor and simulate the human conditions inside a factory. Moreover, in the robotics domain, this model can be used to model and simulate the emotions of the individuals that interact with the robots allowing an improvement of the accuracy of the robot's decision-making.

As future work we want to introduce the human within the model, adding their emotional state through the analysis of body gestures or through the face. To do this, specialized hardware must be used in order to obtain this information, helping to create environments in which humans interact in a transparent way with intelligent entities employing their emotional states.

4.7 Acknowledgements

This work is partially supported by the MINECO/FEDER TIN2015-65515-C4-1-R and the FPI grant AP2013-01276 awarded to Jaime-Andres Rincon.

CHAPTER

5

Developing an Emotional-based Application for Human-Agent Societies

AUTHORS:
J. A. RINCON, V. JULIAN AND C. CARRASCOSA
{jrincon, mgarcia, vinglada, carrasco}@dsic.upv.es
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN
UNIVERSIDAD POLITÉCNICA DE VALENCIA
CNO/ DE VERA SN
46022 VALENCIA, SPAIN

5.1 Abstract

The purpose of this paper is to present an emotional-based application for human-agent societies. This kind of applications are those where virtual agents and humans coexist and interact transparently into a fully integrated environment. Specifically, the paper presents an application where humans are immersed into a system that extracts and analyzes the emotional states of a human group trying to maximize the welfare of those humans by playing the most appropriate music in every moment. This system can be used not only online, calculating the emotional reaction of people in a bar to a new song, but also in simulation, to predict the people's reaction to changes in music or in the bar layout.

5.2 Introduction

Ubiquitous Computing and Ambient Intelligence (AmI) [147], [91] have changed the concept of smart homes, introducing devices that improve people's quality of life. Among other applications, we can highlight smart devices that learn our tastes, smart homes that help reducing energy consumption [61], or safer homes for elderly [65]. To achieve this, developers may use different Artificial Intelligence (AI) tools, sensor networks, and new and sophisticated embedded devices.

The development of applications which employ this new technology, as well as the introduction of multi-agent systems in many of these applications, has allowed the creation of systems in which the human is completely involved in the system. Currently there exist applications in which intelligent agents offer different kinds of complex services to humans in an environment of whole integration. This kind of applications are what we call a *Human-Agent Society* [20], which can be defined as a new computing paradigm where the traditional notion of application disappears. This new paradigm is based on an immersion of the users in a complex environment that enables computation.

The main challenge to achieve real human-agent societies lies in the design and construction of intelligent environments, in which humans interact with autonomous, intelligent entities, through different input and output devices. This means that there are two layers in which humans interact within the environmental and ubiquitous computing intelligence. The first layer is the real world where the human being interacts with other humans and with real objects. The second layer is a virtual layer in which humans interact with virtual entities and objects. This latter layer will be inhabited by intelligent entities (agents), which must be able to perform the different human orders. The virtual environments where agents are involved, are known in the literature as intelligent virtual environments or *IVE*. An *IVE* [60] is a 3D space that provides the user with a collaboration, simulation and interaction with software entities, so the user can experience a high immersion level. This immersion is achieved through detailed graphics, realistic physics, AI techniques and a set of devices that obtain information from the real world. As an example, the $JaCalIVE$ framework [134] enables the design, programming and deployment of systems of this kind.

On the other hand, the development of applications in which people's emotions are taken into account has become very popular over the last few years, but it is a hard and complex process. Humans use their emotions in their decision making. Human beings manage themselves in different environments, either in the working place, at home or in public places. At each one of these places people perceive a wide range of stimuli, that interfere in our commodity levels by modifying our emotional levels. These emotional levels can be expressed in many different ways. The most common is the use of the face to express emotions. The expressed emotion can be detected using an algorithm which analyzes the facial features [155]. The information extracted by this algorithm allows us to identify the detected persons and estimate their emotional state. This emotional information has been very useful in the development of applications of AmI [81] and Robotics[27]. Nevertheless, it is needed for the machines to have the capability of interpreting or recognizing emotional variations. This is the reason for the design of emotional models that interpret or represent the different emotions. These emotional models can be embedded into intelligent agents, which have the capability of perceiving the world, acting and communicating between them. Therefore, humans could be represented in a virtual environment as intelligent agents, which can detect and analyze the emotional state of a specific person and act based on that emotional analysis.

This paper presents a new way of interacting between humans and intelligent autonomous entities living within intelligent virtual environments using emotions as a form of communication. To do this a number of AI tools were used, thus creating a hybrid application which is composed of a multi-agent system, machine learning and statistical classification mechanisms. All this integration of different AI tools is tested in an emotional-based application which is presented in this article. This application controls the music played in a bar with the goal of making all the individuals inside it as happy as possible. In order to achieve this, we use different algorithms for capturing the environment, for detecting faces and music, for converting inputs into emotional values and for analyzing the decision making of the entities. The application has been developed using the $JaCalIVE$ framework [134], which is a framework for the design and simulation of intelligent virtual environments. This framework differs from other works in the sense that it integrates the concepts of agents, humans, artifacts and a physical simulation. The main reason to employ $JaCalIVE$ that it allows an easy integration of human beings into the system.

The rest of the paper is organized as follows: Section 2 presents the related work; Section 3 describes the proposed problem; the application design is presented in Section 4, and Section 5 explains implementation details. Finally, some conclusions are presented in Section 6.

5.3 Related Work

Over the last few years different approaches have been proposed in order to define the emotional state of an entity. Among them, the best known are the *OCC* [112] and

PAD [99] emotional models. The *OCC* model designed by Ortony, Clore & Collins is a model frequently used in applications where an emotional state must be simulated. This has allowed to create applications which emulate emotions in virtual humans [17] or applications which incorporate agents reacting to stress situations [67]. The *OCC* model specifies 22 emotional categories, which are divided into five processes. These processes define the whole system, where the emotional states represent the way of perceiving our environment (objects, persons, places) and, at the same time, influencing in our behavior positively or negatively [4]. However, the *OCC* model utilization presents one important complication due mainly to its high dimensionality which greatly difficult computation.

On the other hand, the *PAD* is a simplified view of the *OCC* model. This model allows to represent the different emotional states using only three values. These three values are usually normalized in $[-1, 1]$, and correspond to the three components conforming the emotional model (*Pleasure, Arousal, Dominance*). These components can be represented in a $\mathbb{R}^3$ space. Each one of the components conforming the *PAD* model allows to influence the emotional state of an individual in a positive or negative way. This influence evaluates the emotional predisposition of such individual, modifying in this way his/her emotional state. For instance, if the *Pleasure* parameter is modified so that its value is positive (the same can be achieve modifying its opposite *Displeasure* with negative values), then such an individual would tend to prefer pleasant stimuli rather than unpleasant ones. In the same way, if the *Arousal* parameter is modified so that its value is positive (or *Calm* is modified to have negative values) then the excitation levels and the duration of it will change. Last, the *Dominance* parameter can be modified (or its opposite *Submissiveness*) to indicate the usual inclination to feel in control facing different situations, and the relation existing with feeling controlled in different circumstances.

The emotional model is only a tool for an intelligent agent in order to understand and represent the human emotion. But, the detection of human emotions by the agent implies the use of different recognition techniques. For the recognition of human emotions, biosignals [29] and body gestures [118] are typically employed. This last feature is maybe the most common technique for the classification of the human emotions. To do this, it is necessary the employ different machine learning techniques. In the literature can be found different applications in which authors use neural networks [66], support vector machines [170] and anthropometric measurements of the face [74]. All these techniques require a preliminary step that previously extracts the face information. This information is a series of points called Facial Feature Points (Figure 5.1).

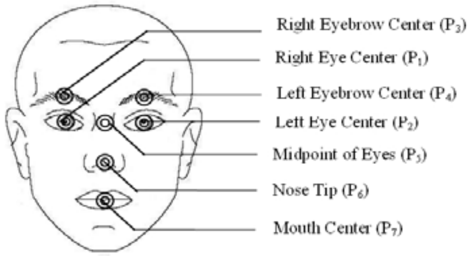


Figure 5.1: Facial Feature Points

Based on these points, the distance between them is calculated (see Figure5.2). The next step consists of using these distances in order to train a neural network, a support vector machine or another machine learning technique for emotion recognition.

The existing emotional models and methods for detecting human emotions are usually employed in applications where multi-agent systems (MAS) are involved, allowing the simulation and the recognition of human emotions for a lonely entity. Emotional states turn into valuable information, allowing to develop applications that help to improve the human being life quality.

On the other hand, over the last few years, there have been different approaches for using MAS as a paradigm for modeling and engineering IVEs and how to introduce the human into this IVE. The use of MAS is mainly due to the characteristics that agents have, such as autonomy, proactivity, reactivity and sociability. Different

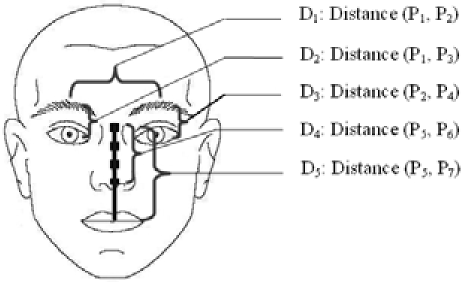


Figure 5.2: Distances between Facial Feature Points

approaches have been devised in order to develop MAS. All approaches for creating multi-agent systems that exist to date should be studied according to two principal categories: those that simply support the creation and interaction of agents (JADE, April, Jason, JIAC), and those that permit the creation of virtual organizations (VO) with such key concepts as norms and roles (MadKIT, Jack, S-MOISE+, AMELI, Janus, Magentix2, Pangea). Until now, these platforms can create agents (some with different models), follow their life cycle and manage communication and services. And in the case of VO support, the platforms take into account the normative and organizational aspects that the platform itself should provide.

However, when modeling an IVE as a MAS it is necessary to take into account that not all the entities are agents. In this case, the main goal is to immerse human beings into IVEs. When a human being is completely immersed into a system of this kind, he/she can interact with the system in a natural way. Moreover, agents immersed in the system can learn about human actions adapting its behaviors and taking decisions about future situations. Examples of these systems can be domotic scenarios, production lines in an industry, entertainment industry, ... This kind of applications is, as it has been commented in the introduction, a Human-Agent Society. Rather than developing software applications that accomplish computational tasks for specific purposes, this paradigm is based on an immersion of the users in a complex environment that enables computation [21]. Regarding the immersion of agents in dynamic physical, virtual or mixed environments, one of the main problems is to define what is the interface between the agents and their environment. One of the ways to solve this problem is to manage this interface as the agent body, i.e. a component that is attached to each agent to manage its interface with the environment [149]. Other works have introduced the use of mediators between agents and the environment [57][158], which can be viewed as functionally similar to bodies.

In order to achieve this kind of immersion, an IVE development process must employ a middleware specially designed for the execution and adaptation of Intelligent Virtual Environments, allowing an easy integration of human beings in the MAS. As far as we know, existing approaches do not allow an easy way to design systems of this kind. The tool that we propose is called JaCalIVE [133]. JaCalIVE specifically defines a type of entity which is a virtual representation of a human. This entity can have the capability of detecting the human, associate itself to it and, moreover, associate many other characteristic that are considered for a better human-agent interaction into the IVE. Moreover, JaCalIVE also facilitates the interface between the agents and their environment allowing the definition of architectural building elements and patterns for agent-environment interactions

5.4 Problem description

The application example is developed in a bar, where there is a DJ agent in charge of automatically playing music and, also, there is a specific number of people listening to the music. The idea behind is that music may change the emotional state of a person or a group, as it influences in the human mood. But not any kind of music, but music that matches people interests. According to this, the main goal of the DJ agent is to play music making that all individuals within the bar are mostly *happy* as possible. Every time that a song is played, each one of the persons placed in the bar will have an emotional response according to his/her musical taste. That is, depending on the musical genre of the song, people will respond varying their emotional states. Moreover, varying emotions of each person will modify the social emotion of the group of people. If the DJ agent could have a way to evaluate the emotional state of the people which are in the bar, he could know the effect that the songs have over the audience. This will help the DJ agent to decide whether to continue with the same musical genre or not in order to improve the emotional state of the group. In some sense, this approach can be considered as a group recommendation technique, where the system tries to recommend the best song for the group of people [92]. Nevertheless, this proposal differs from typical group recommenders where typically the main issue arises how to combine individual user models in order to decide what would be optimal for the group. In this proposal it is not known what is good for each individual user, all the decisions are based on emotion recognition techniques.

This idea differs from other systems such as MUSICFX[94], where the users vote for the played songs, and they could even cheat to only play their favourite songs, in the sense that the users choose the songs unintentionally, avoiding the possibility of cheating for choosing only their favourite songs.

In the proposed application, each one of the individuals will be represented in the system as an intelligent agent, which will have an emotional response according to the person that represents. Moreover, these agents will try to recognize the real emotional states of people through the detection of changes in their facial expressions. With all of this, depending on the song that is being played in the bar, agents representing humans will be able to vary their emotional states in accordance with humans.

Specifically, the operating mode of the proposed application will be the following: in a specific moment the DJ agent will play a song in the bar. In such a way, the proposed application will seek to identify the different emotional states using them as a tool of communication between humans and agents. To do this, agents that represent humans need to recognize the human associated to it, and moreover, recognize the musical genre of the song and, also, recognize the emotion. To perform these detection processes, we need to use pattern recognition algorithms and image and audio processing techniques in order to detect and classify the different emotional states of humans and try to modify the environment in which the humans are. Concretely, the proposed emotional detection uses image processing techniques as a way to capture information about changes in the facial expressions of the persons that are listening to the music into the bar. Moreover, the system can also capture the ambient sound in order to classify the music that is being played. Next subsections describe how the application has been designed and the description of the different employed techniques.

5.5 Application Design

Over the last few years, there have been different approaches for using MAS as a paradigm for modeling and engineering IVEs, but they have some open issues like low generality and then reusability, and weak support for handling full open and dynamic environments where objects are dynamically created and destroyed. As a way to tackle these open issues, and based on the MAM5 meta-model [13], the $JaCalIVE$ framework was developed [134]. This framework provides a method to develop this kind of applications along with a supporting platform to execute them.

The presented work has used an extension of both the MAM5 meta-model along with the $JaCalIVE$ framework to develop Human-Agent Societies, that is, to include humans in the loop. MAM5 allows to design an IVE as a set of entities that can be virtually situated or not. These entities are grouped inside Workspaces (*IVE Workspaces* in the case of virtually situated). Entities are divided into Artifacts (*IVE Artifacts* if situated) and Agents (*Inhabitant Agents* if situated). One new type of situated agents, that is, of *Inhabitant Agents* are *Human-Immersed Agents*, that model the human inside the system.

The application has been developed as a virtual multi-agent system using the $JaCalIVE$ framework where there are will be different entities. Each one of these entities may represent not only real or simulated human beings, as the DJ agent, but also the furniture or the speakers with their location (Figure 5.3). The different types of agents and their main tasks are:

- DJ agent: the main goal of this agent is to achieve an emotional state of happiness for all of the people which are in the bar. When the DJ agent plays a song, it must analyze the emotional state of people. According to this analysis it will select the most appropriate songs in order to improve, if possible, the current emotional state of the audience.

- *Human-Immersed Agent*: it is in charge of detecting and calculating the emotional state of an individual which is in the bar, sending this information to the DJ agent. In order to accomplish its tasks, this agent must have access to a variety of input/output information devices such as cameras, microphones, ...

In order to facilitate the access to this kind of devices, they have been modeled as artifacts (as can be seen in Figure 5.3)[1]. Concretely, there has been designed an artifact for managing each camera which allow the face detection; each microphone is managed by an artifact which captures the ambient sound in order to classify the musical genre; the music DB has been designed as an artifact employed by the DJ agent (it stores around 1.000 songs classified by genres) and there is an artifact for controlling the multimedia player and the amplifiers for playing songs in the bar with the appropriated volume.

Each one of these entities has been designed using $JaCalIVE$ through an XML file describing all its different properties (including physical ones).The XML file allows to describe if you need some kind of sensor to capture information, some type of actuator or simply an agent that does not need real-world information. It is also in this XML where humans are associated with each agent. These XML files are automatically translated into code templates using the $JaCalIVE$ framework.

[1] In this Figure and in the following ones, we are using data from a simple example with only 3 Human-Immersed Agents

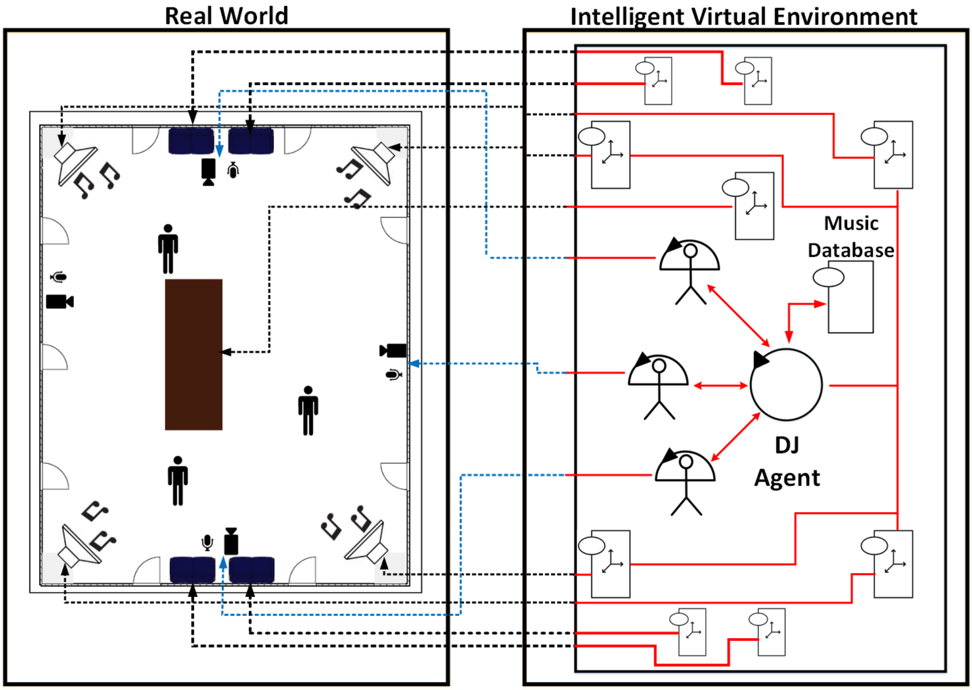


Figure 5.3: General view of the different entities involved in the proposed application

5.6 Implementation

As above commented, the proposed application includes two different types of agents: the DJ Agent and the Human-Immersed Agent. Next subsections explain the functionalities of these two agents.

5.6.1 DJ Agent

Regarding the DJ Agent, this agent uses the information sent by the Human-Immersed Agent to analyze the group's emotional state, using it to decide which is the next song to play. The goal of the DJ Agent is that all the humans feel as happy as possible. To analyze the current or to detect which emotion of the group, the DJ agent uses the social emotional model presented in [137]. This model employs the individual emotions sent by each *Human-Immersed Agent* and gives a way to calculate an aggregated emotional state for a group of agents. This social emotion allows the DJ agent to make a decision about next song to be played.

The process that follows the DJ agent is the following: in each cycle the DJ agent plays a song. Once the song has ended, the DJ evaluates the social emotion of the group of listeners that are within the bar (according to the individual emotional states received from the *Human-Immersed Agent*). In this way, the DJ agent can evaluate the effect that the song has had over the audience. This will help the DJ agent to decide whether to continue with the same musical genre or not in order to improve the emotional state of the group.

To do this, in our current implementation, the DJ agent has a connection with Spotify[2]. In Spotify we have the songs classified into different musical genres (i.e. Rock, Pop, Classical, Country, Jazz and Techno). When the DJ agent must play the next song, it randomly chooses a song of the most appropriated musical genre. The DJ maintains a list of the last played songs (with a number of songs being configurable) to avoid repeating them. So, if the next song is inside this list, the DJ selects again other song. The chosen song will share the same musical genre of the previous song if the emotional state of the group tends to *happiness* or another musical genre if the emotional state moves away from *happiness*. It is very important to emphasize that the DJ does not have absolute knowledge of how the different human beings emotions may be affected by external factors, it only knows how the musical genre can make them happy.

[2]https://www.spotify.com/

5.6.2 Human-Immersed Agent

The Human-Immersed agent is an entity that is associated with a specific human and it is the representation of this human into the virtual environment. This agent allows the data acquisition from the real world that are related with the human directly associated with the agent. This data will be used by the agent in order to estimate the emotional state of the human. After this estimation, the agent will communicate the estimated emotional state to the DJ Agent. These estimations and shippings of the emotional states are repeated in each execution cycle of the agent.

To do this, it is necessary to provide each Human-Immmersed agent with a set of tools so that the DJ agent would be able to know the emotional state of each human. Those tools will help Human-Immersed agents to perceive the real environment, to interpret human it has associated and be able to classify the different emotional states this human expresses. So, each one of these agents contains both audio processing and image recognition mechanisms. In order to achieve the detection of the emotional states, each one of the Human-Immersed agents needs to perform a serie of four main processes in order to recognize the human and his/her emotional state. The first step is the data acquisition process which is divided into two subprocesses, a physical acquisition of the needed data and a face detection mechanism. The second process is in charge of the image processing in order to try to detect the emotional state of the human from his/her facial expressions. This process will be launched only if the face detection mechanism has properly ended. The third process is in charge of the audio processing. This process identifies the music that is being played and estimates the emotional state of the human according to his/her musical tastes. Finally, the last process is in charge of communicating the estimated emotional state to the DJ agent. It's important to remark that the image processing block and the audio processing block are two different ways to estimate the emotional state of a human. The agent will launch the image processing process only if the data acquisition process has been capable of detecting the human face, otherwise the agent will launch only the audio processing mechanism. Figure 5.4 shows all the processes and their relationships.

These processes are explained in detail in the following subsections.

5.6.2.1 Data Acquisition

This first process to be executed by the agent corresponds with the acquisition of the needed information from the real world. Specifically, two physical devices are used for acquiring that information, a *WebCam* and a microphone. The information acquired by these two devices will be used to detect the emotional state of each human. Using these devices the *Human-Immersed agent* has the capability of listening and seeing the environment. The images captured by the webcam are processed using an identification mechanism which is based on the Viola and Jones algorithm [169]. This mechanism allows the face detection of the human that is associated with the agent. This process can fail in some occasions depending on the light conditions and the human movement. According to this, if the agent has been capable to identify the human face, the next step will try to estimate the human mood from the analysis of the facial gestures. On the contrary, if the agent has not identified the face, it will launch the audio processing block. Regarding the audio acquisition, the Human-Immersed agent captures the song that is being played using a microphone. This microphone captures the environment sound which is stored as a *.wav* file. The recorded sound will be analyzed in a subsequent process.

5.6.2.2 Image Processing Block

In this block the *Human-Immersed Agent* divides the process in two sub-processes, *Face Classification* and *Emotion Recognition*. In the face classification process, the *Human-Immersed Agent* uses feature face points [171] to recognize the human associated to the agent. The API *Dlib*[3] was used to extract these feature points of each one of the faces. These points were used to recognize the persons and to classify the emotions by calculating different distances between the points. This process is

[3]http://dlib.net/

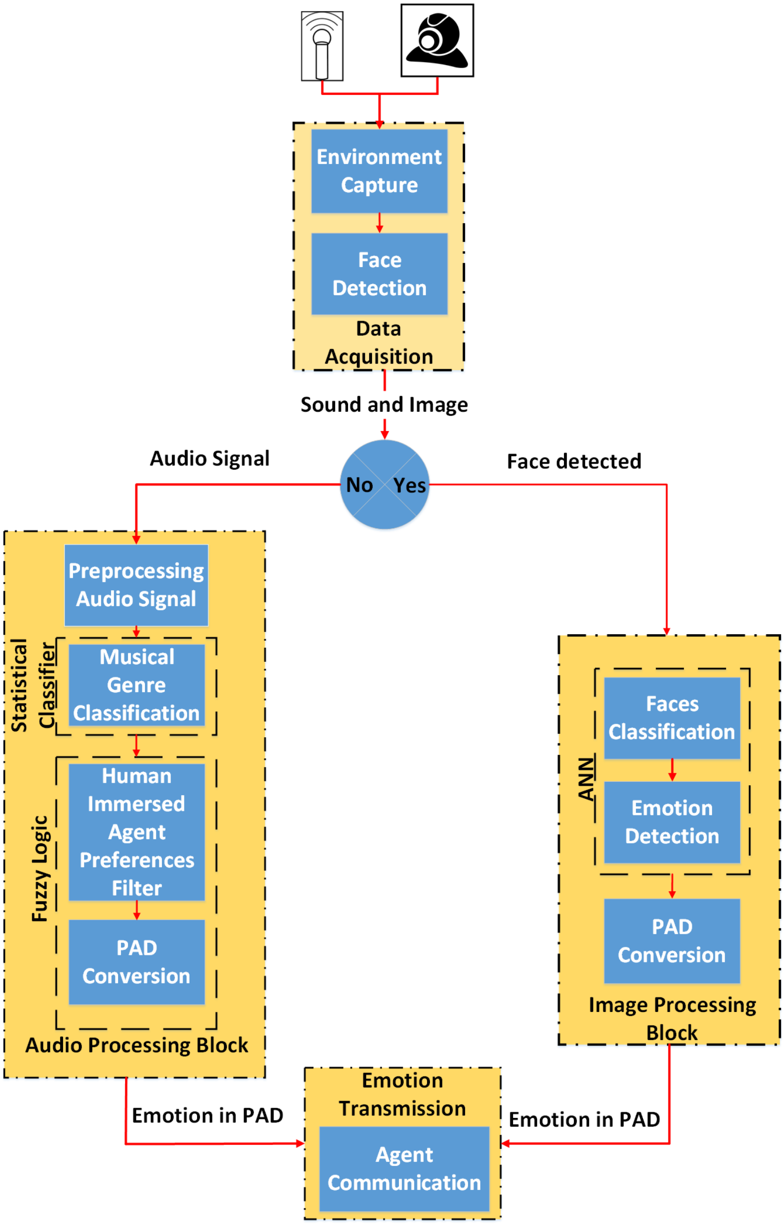


Figure 5.4: Process sequence in the Human-Immersed agent

very important because is in which the *Human-Immersed Agent* learns how is the human that the agent has associated and it only captures information from that human. The face classification process uses feature faces points [171]. In our case twenty two points were used, and based on these points eighteen Euclidean distances were calculated. With these distances a vector for each person was created. This process was repeated in order to obtain seventy-four vectors. These vectors were grouped to create a matrix of 18x74, where 18 corresponding to eighteen distances between face points and 74 corresponding to the number of samples, that is, the number of photographs being used for a person.

This matrix is the face information for each human associated to a *Human-Immersed Agent*. In our experiments we have employed three different classes (persons) creating a matrix of $18x222$. One of the problems of this matrix is its size, because adding a new person implies to add 74 new columns to the matrix. For this reason, it is very important to try to reduce the dimensionality of this matrix and to find a linear separation between classes. Thus, we have used Linear Discriminant Analysis (LDA) [41] to find linear separation between different classes. Figure 5.5 shows the distances between points before the use of *LDA* and Figure 5.6 shows the points after using LDA.

For the classification process, we have used a logistic regression. Figure 5.7 shows the classification map, with the delimitations for each class. To classify the images, the agent calculates the distance between points and do the test. An example of the obtained results after the test can be seen in Figure 5.8.

Once the face has been classified, the next step is to detect the emotion. To classify the emotion we use eight face feature points [170]. Based on these points, we calculate the displacements distance (Euclidean distances) of the feature points between a neutral facial expression and the six particular emotive expressions (Afraid, Angry, Disgusted, Happy, Sad and Surprised) (as it can be seen in Figure 5.9). It can establish a characteristic motion pattern for each expressed emotion. These characteristics were used to train a machine learning model. The training method used the database

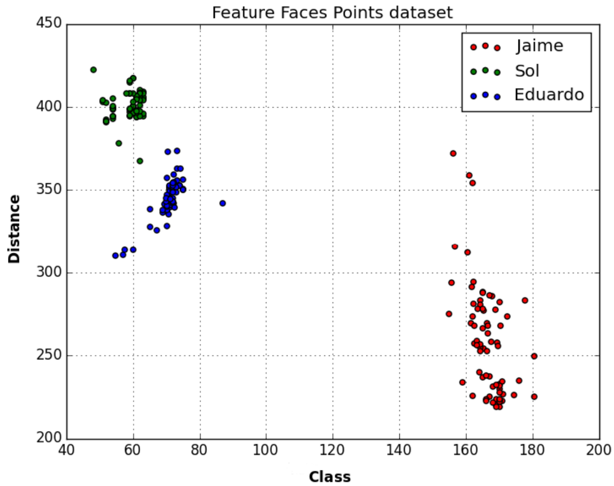


Figure 5.5: Distance Feature Face Points without LDA

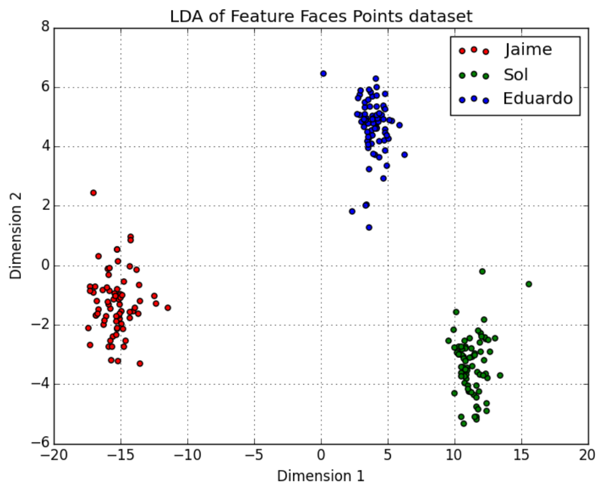


Figure 5.6: Distance Feature Face Points with LDA

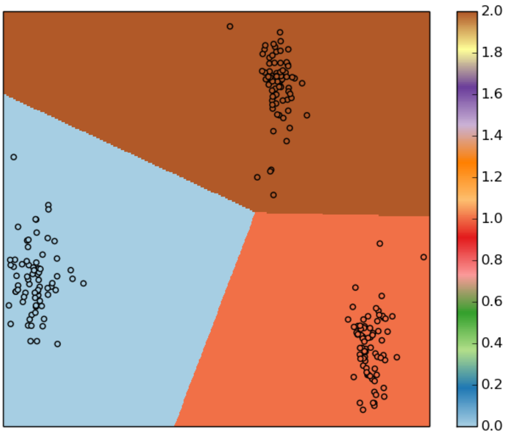


Figure 5.7: Example of the classification map

of *KDEFEmotional Lab at Karolinska Institute* [54] which is divided into 4900 pictures of 562 x 762 pixels and it specifies the related emotion (Afraid, Angry, Disgusted, Happy, Neutral, Sad or Surprised). In this way, in order to detect the emotions, it is necessary to extract some specific attributes from the images. These attributes are the distance between some specific points of the face. Figure 5.10 specifies the eight points with which the displacement vector can be calculated.

In the training process, the ninety percent of the images from the database were used and the rest of the images were dedicated to the test stage. The machine learning algorithm used to recognize human emotions is an artificial neural network *(ANN)*. The architecture of the neural network is shown in Figure5.11. The inputs are the eighth distances calculated using the database and the output are the seven emotions to recognize (in fact, the ANN only classifies 6 emotions, the neutral emotion is obtained when the ANN does not return any value). After this process, the agent has an estimation of the emotional state of the human through an analysis of the facial gestures. Following the equivalences shown in Table 5.1 (the values were extracted

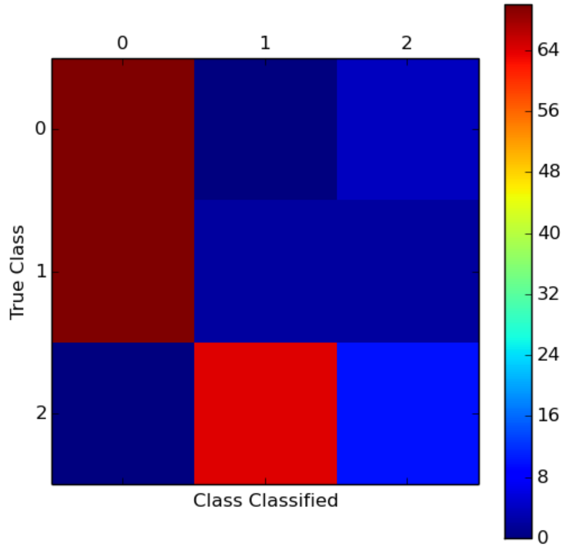


Figure 5.8: Sample output of the Face Classification

from [162]) the agent transforms the estimated emotion to a PAD value. For instance, the emotion *Happy* corresponds with the PAD values of $[0.55, 0.24, 0.28]$). This emotional state expressed as a PAD value will be transmitted to the DJ agent in a subsequent process.

Moreover, some experiments have been made in order to evaluate how the light conditions into the bar can potentially affect the ML models. In this way we have test the proposed ANN in different situations, changing the luminosity (100%, 60%, 40%) and using different cameras such as RGB or infra-red (IR). Table 5.2 shows the obtained results, as can be seen, the classification rate gets worse when reducing luminosity. On the other hand, the RGB camera has been tested with normal luminosity levels without improving the results obtained with a normal camera.

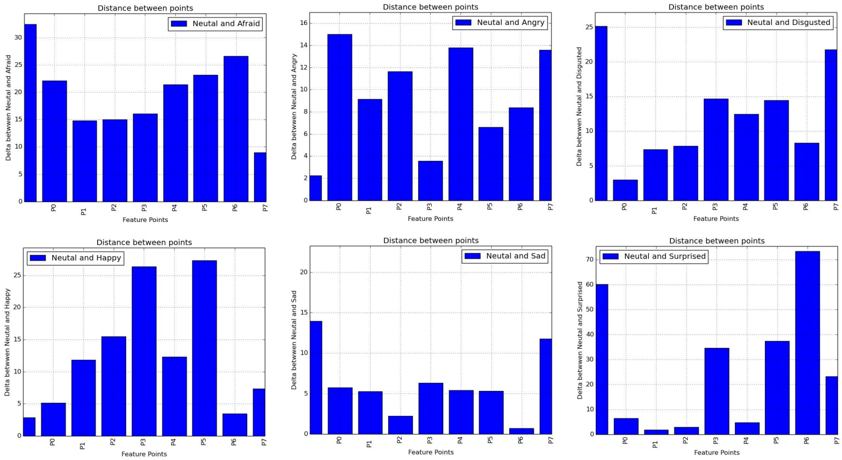


Figure 5.9: Distance between Neutral Face Points and Face Points of Afraid, Angry, Disgusted, Happy, Sad and Surprised emotions

	P	A	D
Afraid	-0.08	0.18	-0.39
Angry	-0.51	0.59	0.25
Disgusted	-0.36	0.08	0.13
Happy	0.55	0.24	0.28
Sad	-0.18	0.03	-0.14
Surprised	0.34	0.34	0.04

Table 5.1: Mapping between emotions and PAD values

Nevertheless, the IR camera notably improves results obtained with low luminosity. According to these results, it's obvious that the best results are obtained with high luminosity levels, which can be complicated in some kind of bars. In these cases, IR camera can be used but with lower probability of success.

Figure 5.10: Example of the Feature Face Points detection

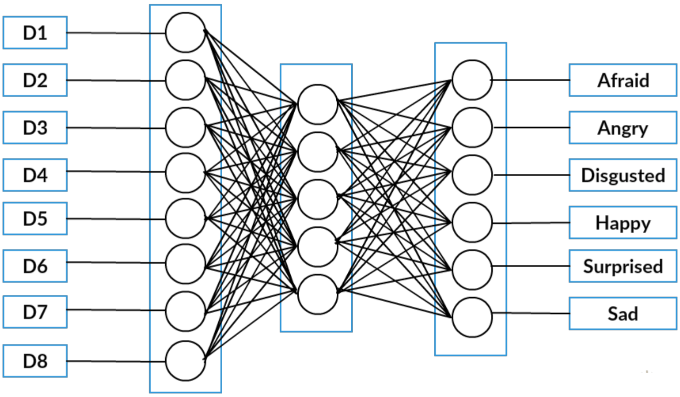


Figure 5.11: Graphical view of the ANN Architecture used for Emotion Recognition

	Classification Rate	MSE
Light 100%	93.5	0.063
Light 60%	74.1	0.112
Light 50%	40.5	0.450
RGB Camera%	93.1	0.062
IR Camera%	84.5	0.091

Table 5.2: Experiment results with different luminosity levels and cameras.

5.6.2.3 Audio Processing Block

This process will be executed if the agent is not able to detect the human face. In this case, the agent tries to detect the musical genre from the previously recorded sound.

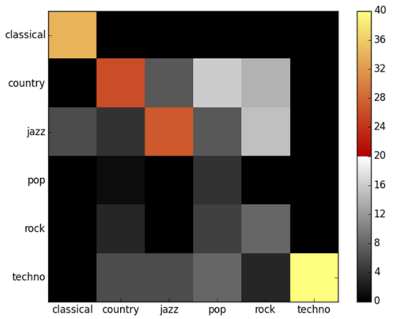
This block is divided in two phases, the first one is an offline training the second one is the execution of the machine learning model into each *Human-Immersed Agent*, along with the transformation to an emotion expressed as a PAD value:

- *Phase 1: Off-line training:* The offline process consists of an statistical model training [130][64][163] for the recognition of six musical genres (classical music, jazz, country, pop, rock and techno music). This model is the same for all the agents, that will use it to identify the musical genre.
- *Phase 2: On-line execution:* This phase consists of the following steps:
    1. *Musical genre classification:* Each *Human-Immersed Agent* executes its musical genre classificator, to identify the song being played by the DJ.
    2. *Human Immersed Agent preferences filter:* Once the agent has obtained the musical genre, it is necessary to extract the emotion associated to this genre. As each agent is associated to a human, it knows the musical preference of its associated human. This knowledge is a list of musical preferences that was given for its associated human to him. Based on this information, it was used a fuzzy-logic algorithm to get the *PAD* value corresponding to the emotion produced by the musical genre in

the human associated to the agent. To get the best membership function several experiments were performed by modifying the functions of fuzzy logic, using triangular functions, Gaussian and trapezoidal. Modifying each of these functions in the model, the answer could vary the input *PAD* values. In our experiment we used the triangular membership.

It is important to highlight that the ambient noise is a serious problem. For this reason, it is important to have a series of filters in the *Human-Immersed Agent* as *Butterworth* with different configurations [154], [71].

The variation of the membership function depends on the corresponding human musical preferences, e.g. a human can respond favorably to pop music but not to the blues music. This is the reason why it is needed that each human configure its Human-Immersed Agent before using the system, varying the membership function of the fuzzy logic module.



Classical	**34**	0	0	0	0	0
Country	0	**26**	7	16	14	0
Jazz	6	4	**27**	7	15	0
Pop	0	1	0	**4**	0	0
Rock	0	3	0	5	**8**	0
Techno	0	6	6	8	3	**40**
	Classical	**Country**	**Jazz**	**Pop**	**Rock**	**Techno**

Figure 5.12: Example of a Numerical and Graphical Confusion Matrix.

5.6.2.4 Emotion Transmission

Finally, once the emotion has been estimated, it is necessary to send this emotional value to the *DJ Agent* in order to calculate the social emotion [137] of all of the people which are in the bar. This social emotion model employs a representation of the emotional value according to the *PAD* model where the emotion is represented as a vector in the $\mathbb{R}^3$ space and each component have a scale range from $[-1.0, 1.0]$. These PAD values are sent to the *DJ agent* which receives all the emotional values from all the Human-Immersed agents that are active.

5.7 Conclusions and future work

Multi-agent systems allow the design and implementation of applications where the main components can be humans and software agents. These agents must be able to interact and communicate with humans in order to help them in their daily activities. In this sense, this paper presents an ambient intelligence application where humans and agents must coexist in a framework of maximum integration. The application has been developed over the $JaCalIVE$ framework allowing an easy integration of the human in the multi-agent system and a visualization of the system in a virtual environment. Thus allowing the agents to learn and detect the human emotion and recognize the human associated to each *Human-Immersed Agent*. The proposed system is able to extract (in a non-invasive way) and to analyze the social emotion of a group of persons and it can take decisions according to that emotional state. Moreover, it can be used in an online fashion, where the system is reflecting what is happening in the real world at the same time, or it can be used to simulate - predict what would happen in the conditions that are used in the simulation, where this conditions can be about the number of agents and their preferences and also about the layout of the environment. As a future work we are applying this system to other application domains, as it can be extracted out of the music domain, and carried out to any other ambient intelligence domain, even an industrial one, where it can monitor and simulate the conditions inside a factory.

On the other hand, another aspect to be considered as future work is the improvement of users' satisfaction including other elements in the decision-making such as the opinion of the user. The idea is to include automated negotiation techniques, like employed in [48], in order to select a subset of songs or genres that everyone likes in some degree. This will reduce the complexity of the song selection process made by the DJ agent. Moreover, groups of people with very similar musical tastes could be detected into the bar. These groups can play as negotiation teams in this process, as in done in [145] and [144]. We are also planning the deployment of our proposal in the MEDERI[4] living lab in our university. This living lab is a multidisciplinary environment, mainly focused on health technology, that will give us the needed tools for a real involvement of users in order to improve the experience and robustness of the proposed techniques.

5.8 Compliance with Ethical Standards

Funding: This work is partially supported by the MINECO/FEDER TIN2015-65515-C4-1-R and the FPI grant AP2013-01276 awarded to Jaime-Andres Rincon.

Conflict of Interest: Jaime Rincon declares that he has no conflict of interest. Vicente Julian declares that he has no conflict of interest. Carlos Carrascosa declares that he has no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors. All the data used in the experiments were obtained from public datasets.

[4] http://mederi.ai2.upv.es/

CHAPTER

6

Introducing Dynamism in Emotional Agent Societies

AUTHORS:
J. A. RINCON[1], A. COSTA[2], G. VILLARUBIA[3], V. JULIAN[1*], C. CARRASCOSA[1]
{jrincon, vinglada, carrasco}@dsic.upv.es
[1]DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN (DSIC),
UNIVERSITAT POLITÈCNICA DE VALÈNCIA
CNO/ DE VERA SN
46022 VALENCIA, SPAIN
[2]ISLAB / ALGORITMI, ESCOLA DE ENGENHARIA, UNIVERSIDADE DO MINHO, GUIMARÃES
[3]BISITE, DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF SALAMANCA

6.1 Abstract

This paper presents the development of a dynamic emotional model to be employed in agent societies. The proposed model is based on the *PAD* emotional model and allows the representation of the emotional contagion phenomena of a heterogeneous group of agents that are capable of express emotions. The model is mainly based on three elements: personality, empathy and affinity. These elements allow the characterization of each individual, causing them susceptible to vary in some degree the emotions of other individuals. Additionally, the model allows defining of the social emotion of this group of agents.

6.2 Introduction

To attain a person's intentions it is essential to grasp the psychological and the physical aspects. Disregarding one of these aspects may lead to unreliable results [68]. The physiological representation of decisions is a very powerful way to determine if a person is being honest or not. For instance, a person can be actively lying but the physical response may tell otherwise. Although there are ways to overcome this situation where the person is able to physically control the display of emotions most of the people does not [116].

Most of the human-computer interaction is solely based on text or clicks, which looses several features present in normal human communication. For instance, the lack of knowledge about one's emotional status constricts the information about the actual disposition (in terms of acceptance/refusal). Furthermore, there are several other factors that influence the emotional status and, therefore, the possible response to a suggestion. Therefore, to enhance the machine decisions it is critical that the complete information about the human/agent in each interaction.

One of the most important influencers of the emotional status in the surrounding environment. Changes in the environment can have an positive or negative outcome (improve the mood or tone down the excitement). From a computational perspective, emotions have been employed as a way to improve social simulation processes which require human interactions, but very little work has been done on representing collective emotions and emotion's dynamicity [50].

Several models have been developed to address emotions and personality so that they are able to recognize and simulate emotions, and they are: *OCEAN* model [95], *OCC* emotional models [112], *Plutchiks* theory [119] and the *PAD* model [99]. With these models we are able to produce agent systems that are capable of interacting with humans and express emotions. When interacting with other agents they can also perceive the environment. Furthermore, if the agents are equipped with environment sensors (such as cameras[28], speech analysis[153], bio-sensors[24], etc.) they are able to extract the emotional information of humans (although they are still at an early stage).

The issue with the available proposals is that they are static and impervious to change. Normal interactions between the entities (humans or agents) may result in an environment change, like a burst of emotion contagion (here the emotional change of an entity influences other entities). Emotion contagion may be possible when the model takes into account elements such as situation, affinity, empathy, etc.. Until now, only a few works have tried to model the emotional contagion in computational entities: Saunier and Jones[148] modelled the emotional contagion suggesting that each agent is the body and the mind separated; Bosse et al. [25] proposed the spiral model that gives a solution to the emotional propagation by distinguishing the different factors that influence the emotional contagion.

Our proposal, which is an extension of a previous work [131], is to introduce a model that is dynamic, which learns from the entities actions and the environment changes. We have based our model on the *PAD* emotional model with the additional ability of representing the emotional contagion of a heterogeneous group of entities capable of express and/or communicate emotions. Moreover, with this advance it is possible to define the social emotion of a group of agents. To define the model we employ concepts like empathy, affinity and personality of each entity. The aim of this proposal is to attain the complete information about the current emotional status so it can be used on other systems so that they are able to improve their operation.

One of the projects that will benefit from this advance is the iGenda platform [36, 37]. The iGenda helps managing everyday activities of elderly or disable people. The main features are the scheduling system, the social network and the medical status monitoring. The platform also implements active aging efforts by automatically scheduling activities on the users' free time. One of the issues with this approach is that automatized systems are constructed having one or a few profiles serving as the base options for all the community. Therefore, the people that do not fit on those profiles do not really benefit from the platform features. Our proposal will improve the execution procedures of the iGenda by using virtual actors that have responses similar to human responses.

This paper is structured as follows: section 2 presents the related work and the robotic advances that relate with our concept; section 3 presents the dynamic emotional model and its logical structure; section 4 presents the validation tests of the model presented in section 3; section 5 presents the validation of the model in a real life scenario through the use of a mobile robot; finally, section 6 presents the conclusions and future work.

6.3 Related Work

The emotional states are defined as the way to express emotion by human beings in a period of time. These emotions are not static and can be propagated through the environment, begin widely used in crowd simulation. It is essential to these applications too have the ability of emulating emotion as they are used to the decision making process. In crowd simulation the most common emotional state is fear, which allows the creation of emergency evacuation simulations [2], [8]. Nevertheless, these simulations try to predict the behavior of humans in distress. These simulations have helped to design buildings, evacuation routes and simulate how the police, firefighters and ambulance may optimally respond to a disaster situation [63]. However human being have a whole range of emotions that can be propagated to other agents, such as: happiness, sadness and anger, among others. To propagate these emotions the *Newtonian Emotion System (NES)* [87] was designed for multi-agent systems, establishing the three laws of motion presented by *Newton*. In the *Newton* dynamic the aim is the study of movements of objects and the origin of these movements, where each object is represented by a particles system. Each one of these particles have internal properties which makes them different to the other particles properties as the mass, length, with and height, among others this provide to the object a different behaviour when external forces are acting on it. The application of these forces on a particle can changes your direction and velocity or knows if this particle is attracted to another. The authors based on their model in the *Newton* laws and apply some of the concepts presented by *Newton*, concepts as *force*, *mass*, *acceleration* and *velocity*. Using this concepts the author defined two laws of emotion dynamics, this two law is based on the laws of dynamic of *Newton*.

Other works have tried to introduce the contagion effect that humans can feel in multiple situations. One of these works is the emotional contagion spiral model [26]. This model tries to give a solution to the emotional propagation, distinguishing among different factors that influence in the emotional contagion. This model is based on a emotional model that was proposed by *Barsade* [15], which includes six hypotheses about how is produced the propagation of emotions. This work is applied in an evacuation simulation scenario, taking into account how human behaviours are affected by the dynamicity and propagation of emotions. Nevertheless, the complexity of these analysis forces these approaches to be limited to one emotion, in this case fear. So, behaviours of simulated agents are also affected by only one emotion.

We aim towards a harmonious environment that is just the opposite of what is presented in these works. But that gives us an advantage that is the knowledge about the efficacy of those solutions, thus it is foreseeable that with opposite stimulus there is an opposite response. That is confirmed by current developments in the home robotic assistance area.

6.3.1 Robotics Application

Currently there are some efforts to implement robot systems in the home environment with the aim of helping people on their daily tasks, track them on their home (and report their health status) and be a sentient companion. Studies show that although current human-robot interactions are far from optimal, future developments would be vital for supporting people [121]. There are several projects directed to assist users and interact with them in an effort to change their mood and influence their emotions. The most active and similar projects are the following.

The Hobbit project [49] is an service robot whose aim is to provide assistance in performing certain tasks. It also interacts with the users, asking them for help to perform tasks that it is unable to do, forcing interaction. It has limited interaction features and does not establishes a communicative environment.

The Cosero robot [156] is a humanoid-like attempt that focus more on visual interpretation and grasping. Its aim is to aid to perform home tasks, like cleaning and serving beverages, being able to receive vocal commands or typed messages. It does not posses any type of communicative features, being at its core a service robot.

Lastly, there is the SERGIO robot [86] that is a humanoid robot, that like the previous one, it is a service robot, aiming to perform basic tasks like grasping and object identifying. It is able to communicate using a natural language processor, thus it is capable of basic human communication and keeping a very simple conversation. It is able to receive structured voice command and navigate in an home environment.

The issue with these robots is that they are unable to assimilate the current social and emotional condition of the environment, i.e. they are only capable of performing their tasks whether the user wants it or not. Their strict operation patterns create a distance between them and the humans, being considered as toy by not meeting the user's expectations.

This work tries to overcome this issue by giving an approximation of a dynamic emotional model that allows the representation of the emotional contagion of a heterogeneous group of entities capable of express and/or communicate emotions. The next section explains in detail the proposed model.

6.4 Dynamic Emotional Model

This section proposes a dynamic emotional model based on the PAD emotional model. This model represents the emotional contagion of a heterogeneous group of entities capable of expressing and/or communicating emotions.

Before defining the dynamic emotional model, it is necessary to define the representation of an emotional state of an agent on the PAD model (Pleasure, Arousal and Dominance). The emotion of an agent ag_i in an instant t ($\vec{E}_t(ag_i)$) is defined as a vector in $\mathbb{R}^3$, represented by the components that make up the *PAD* emotional model. The variation of each component allows to modify the emotional state of the agent (Equation 6.1).

$$\vec{E}_t(ag_i) = [P_t(ag_i), A_t(ag_i), D_t(ag_i)] \tag{6.1}$$

This representation in $\mathbb{R}^3$ allows us to see emotions as a system of particles. They attract or repel depending on the internal properties of each one of them. These particles have the ability to move around the space because these particles have internal properties like *Mass*. The mass in a particle is a measure of the amount of matter that has a body, and one of the properties related to it is that it is proportional to the resistance to be attracted by others.

The attraction carried out in the PAD space reflects the emotional contagion between entities. An entity will be more easily suffer from contagion of other emotions according to different factors. The main factor, depending on the own entity is called *Empathy*. The empathy is a psychological motivator for helping others in distress [96]. The empathy could be defined as the ability to feel what other people feel. The empathy denotes a deep emotional understanding of another's feelings or problems, while sympathy is more general and can apply to small annoyances or setbacks. Our dynamical model uses this psychological concept, allowing agents to have an empathy level. The *Empathy Level* of an agent ag_i, denoted $\varepsilon(ag_i)$, represents a value in the range [0, 1] indicating the ability of agent ag_i to perceive what another agent may feel. In the PAD space, the mass of agent ag_i ($m(ag_i)$) is defined as the inverse of empathy (Equation 6.2) as an indicator of the difficult to be attracted by others, that is to be contagied by other emotions as $m(ag_i)$ increases.

$$m(ag_i) = \frac{1}{|\varepsilon(ag_i)|} \tag{6.2}$$

Another important factor in the emotional contagion is the relationship between the emotion source and the emotion receiver, that is, the *Affinity* existing between them. It is not the same to perceive the emotions of a close acquaintance than a stranger. The Affinity Level between two agents ag_i and ag_j at instant t ($Af_t(ag_i, ag_j)$) is a value between $[-1, 1]$ that describes the level of affinity between agents ag_i and ag_j, being -1 the value dedicated to sworn enemies, 0 to perfect strangers and 1 to best of friends. The last factor to take into account in the emotion dynamics is the physical distance between the emotion source and the emotion receiver ($D_t(ag_i, ag_j)$ to denote the physical distance between entities ag_i and ag_j at instant t) .

The emotional dynamics described is based on the *Newton* universal attraction law. Newton's law of universal gravitation states that any two bodies in the Universe attract each other with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. Based on this theory, we define the force that an agent ag_j makes over an agent ag_i at instant

t ($\vec{F}_t(ag_i, ag_j)$) to attract or repulse it in the PAD space, that is, this force will control the emotion contagion between all the agents. The emotional force is a vector in $\mathbb{R}^3$ space. This vector measures the emotional change in the PAD space (Equation 6.3).

$$\vec{F}_t(ag_i, ag_j) = \frac{\varepsilon(ag_i) \cdot Af_t(ag_i, ag_j)}{2^{D_t(ag_i, ag_j)}} \cdot ||\vec{E}_t(ag_i) - \vec{E}_t(ag_j)|| \tag{6.3}$$

$\vec{F}_t(ag_i, ag_j)$ represents the force vector, which help us to know if the emotion of the agent ag_i is attracted by the agent ag_j. $\varepsilon(ag_i)$ represents the emphatic level of entity ag_i, and $Af_t(ag_i, ag_j)$ represents the affinity level between ag_i and ag_j at instant t. $D_t(ag_i, ag_j)$ is the physical distance between ag_i and ag_j at instant t and $\vec{E}_t(ag_i)$ represents the emotion of the ag_i at instant t and $\vec{E}_t(ag_j)$ represents the emotion of the ag_j at instant t. According to this, we define the *Emotional Attraction Force* of agent ag_i at instant t ($\vec{EAF}_t(ag_i)$) as the combination of all the attraction forces over agent ag_i at instant t (Equation 6.4).

$$\vec{EAF}_t(ag_i) = \sum_{\forall ag_j \neq ag_i} \vec{F}_t(ag_i, ag_j) \tag{6.4}$$

To calculate the new emotion of agent ag_i at instant $t+1$ and assuming that there is no external stimuli that may change agent ag_i emotion out of the rest of entities in the system, it will be calculated according to movement in the PAD space. To get this new emotion it is necessary to use the second law of *Newton's* or the fundamental principle of dynamics. Based on this law, the $\vec{EAF}_t(ag_i)$ is used to calculate the emotional acceleration of agent ag_i at instant t ($\vec{a}_t(ag_i)$). This acceleration is the emotional variation per time unit of agent ag_i emotion (Equation 6.5).

$$\vec{EAF}_t(ag_i) = m(ag_i) \cdot \vec{a}_t(ag_i) \tag{6.5}$$

Once the emotional acceleration $\vec{a}_t(ag_i)$ is calculated, the emotional velocity of entity ag_i at instant t can be obtained ($\vec{v}_t(ag_i)$). This is a measure of the emotional propagation velocity within the PAD space (Equation 6.6).

$$\vec{v}_t(ag_i) = \vec{a}_0(ag_i) + (\vec{a}_t(ag_i) \cdot t) \tag{6.6}$$

Finally, it is necessary to calculate the new PAD emotion for entity ag_i at instant $t+1$ ($\vec{E}_{t+1}(ag_j)$) (Equation 6.7).

$$\vec{E}_{t+1}(ag_j) = \vec{E}_t(ag_j) + (\vec{v}_t(ag_i) \cdot t) \tag{6.7}$$

It is important to consider that emotions within the PAD space do not present any opposition by the environment, e.g., there is no friction causing a reduction of speed. There is no inercia affecting the emotions within the PAD space thus, there are no oscillations. This swing up was eliminated by adding this restriction to the model **if** $E\vec{A}F_t(ag_i) = 0$ **then** $\vec{v}_t(ag_i) = 0$.

The proposed dynamic model allows us to model and represent the emotional contagion phenomena among different intelligent agents. Nevertheless, these entities typically are not alone in the environment but are part of a group of agents. Our proposal is to model not only how an agent is influenced by other agents but also how the group of agents as a whole can be emotionally affected by its components. To do this, we need to define a social emotional model, which allows to calculate and represent the social emotion of a group of intelligent entities. Next subsection presents the proposed model for representing social emotions.

6.4.1 Social Emotional Model

The aim of the social emotional model is to obtain the social emotion of a group of heterogeneous agents in an specific instant. This model is composed by a triplet that allows us to define the social emotion (*SE*) [138] for a group of n agents $Ag = \{ag_1, ag_2, ..., ag_n\}$ at instant t (Equation 6.8).

$$SE_t(Ag) = (\vec{CE}_t(Ag), \vec{m}_t(Ag), \vec{\sigma}_t(Ag)) \tag{6.8}$$

Where $\vec{CE}_t(Ag)$ is a vector in the PAD space, where each one of its components is calculated averaging the P, A, and D values, respectively of the n agents forming the set Ag (Equation 6.9). These averages will enable us to determine where the central emotion (*CE*) of this group of agents is and to visualize it in the *PAD* space.

$$\bar{P}_t(Ag) = \frac{\sum_{i=1}^{n} P_t(ag_i)}{n}, \bar{A}_t(Ag) = \frac{\sum_{i=1}^{n} A_t(ag_i)}{n}, \bar{D}_t(Ag) = \frac{\sum_{i=1}^{n} D_t(ag_i)}{n},$$
$$\vec{CE}_t(Ag) = [\bar{P}_t(Ag), \bar{A}_t(Ag), \bar{D}_t(Ag)] \tag{6.9}$$

The $\vec{m}_t(Ag)$ can indicate if there exist agents having their emotional state far away from the central emotion. The Euclidean distance is used to calculate the maximum distances between the emotion of each agent respect to the $\vec{CE}$ (Equation 6.10, 6.11, 6.12, 6.13) as follows.

$$mP_t(Ag) = max\left(\sqrt{(P_t(ag_i) - \bar{P}_t(Ag))^2}\right), \forall ag_i \in Ag \tag{6.10}$$

$$mA_t(Ag) = max\left(\sqrt{(A_t(ag_i) - \bar{A}_t(Ag))^2}\right), \forall ag_i \in Ag \tag{6.11}$$

$$mD_t(Ag) = max\left(\sqrt{(D_t(ag_i) - \bar{D}_t(Ag))^2}\right), \forall ag_i \in Ag \tag{6.12}$$

$$\vec{m}_t(Ag) = [mP_t(Ag), mA_t(Ag), mD_t(Ag)] \tag{6.13}$$

The $\vec{\sigma}(Ag)$ or standard deviation (SD) allows the calculation of the level of emotional dispersion of this group of agents around the central emotion $\vec{CE}(Ag)$ for each component of the *PAD*(Equation 6.14).

$$\sigma P_t(Ag) = \sqrt{\frac{\sum_{i=1}^{n}(P_t(ag_i) - \bar{P}_t(Ag))^2}{n}}, \forall ag_i \in Ag$$
$$\sigma A_t(Ag) = \sqrt{\frac{\sum_{i=1}^{n}(A_t(ag_i) - \bar{A}_t(Ag))^2}{n}}, \forall ag_i \in Ag \quad (6.14)$$
$$\sigma D_t(Ag) = \sqrt{\frac{\sum_{i=1}^{n}(D_t(ag_i) - \bar{D}_t(Ag))^2}{n}}, \forall ag_i \in Ag$$

The result of each of the above equations can be represented as a vector (Equation 6.15), which allow to determine the level of emotional dispersion.

$$\vec{\sigma}_t(Ag) = [\sigma P_t(Ag), \sigma A_t(Ag), \sigma D_t(Ag)] \quad (6.15)$$

From this definition, it can be deduced that:

1. **if** $\vec{\sigma}_t(Ag)$ >> **[0,0,0]**, the group has a high emotional dispersion, i.e. the members of the group have different emotional states.
2. **if** $\vec{\sigma}_t(Ag) \cong$ **[0,0,0]**, the group has a low emotional dispersion, this means that individuals have similar emotional states.

This model takes into account that at some stage you may have two or more agent groups and each group has its own social emotion or have a single group which wants to move to a target emotion. This will allow to measure the emotional distance between the current social emotional group and a possible emotional target. This approach can be used as a feedback in the decision making process in order to take actions to try to move the social emotion to a particular area of the *PAD* space or to allow that the emotional state of a group of agents can be approached or moved away from other groups of agents (Equation 6.16).

$$\Delta_{SE} : SE_t(Ag^i), SE_{t'}(Ag^j) \rightarrow [0,1] \quad (6.16)$$

According to this profile, Equation 6.17 shows how we calculate this emotional variation. The equation calculates three distances corresponding to the three components of the *SE*.

$$\Delta_{SE}(SE_t(Ag^i), SE_{t'}(Ag^j)) = \frac{1}{2}\Big(\omega_c \Delta(\vec{CE}_t(Ag^i), \vec{CE}_{t'}(Ag^j)) + \omega_d \Delta(\vec{m}_t(Ag^i), \vec{m}_{t'}(Ag^j)) + \omega_v \Delta(\vec{\sigma}_t(Ag^i), \vec{\sigma}_{t'}(Ag^j))\Big) \tag{6.17}$$

$$where \quad \omega_c + \omega_d + \omega_v = 1; \quad \omega_c, \omega_d, \omega_v \in [0,1] \tag{6.18}$$

and Δ calculates the distance between two vectors. As every dimension of the *PAD* space is bounded between $[-1, 1]$, each Δ will give values between $[0, 2]$. Therefore, Δ_{SE} will have a range between $[0, 1]$. Calculating the distance among social emotions allows the study of the behaviour of emotional-based agents, either minimizing or maximizing the $\Delta_{SE}(SE_t(Ag^i), SE_{t'}(Ag^j))$ function. This way, it can be extrapolated the knowledge about if an agent group approaches or moves away from a specific emotional state. To achieve this, it is necessary to modify through stimuli the individual emotions of each agent and therefore changing the social emotion.

Using this model is possible to determine the emotional distance among different groups of agents or between the same group in different instants of time. This will allow us to measure the emotional distance between the current social emotional group and a possible emotional target. Moreover, the combination of the presented models allows us to model and represent the emotional contagion of a heterogeneous group of agents and observe how it influences the social emotion of that group of agents.

6.5 Validation tests

Different tests have been done in order to validate the proposed model. Concretely, a simulation prototype was implemented in Python (using a *jupyter*[1] notebook with *numpy* and *matplotlib* libraries). The simulation experiments were conducted to evaluate different aspects and to try to show the correct behavior of the proposed model. Visualization of results has been done using three different kind of images:

- PAD space representation: a 3D representation of the emotional states in the PAD space. In each graphic, current emotional states of each agent and the social emotion of the existing groups are represented.
- Physical space position representation: a 2D representation of the different agents, similar to a graph where each agent is a node situated in its physical coordinates (x,y). The size of the agent is inversely proportional to its empathy and if there is any affinity between agents, it will be represented by a link joining them. Finally, a sequence of colors (see Figure 6.1) is defined as a way for representing the current emotion of each agent.
- Social emotional evolution: a 2D representation of the evolution of the different values composing the Social Emotion ($SE_t(Ag)$):
  - $\vec{CE}_t(Ag) = [\bar{P}_t(Ag), \bar{A}_t(Ag), \bar{D}_t(Ag)]$, represented in the figure as *CE P*, *CE A* and *CE D*, respectively.
  - $\vec{m}_t(Ag) = [mP_t(Ag), mA_t(Ag), mD_t(Ag)]$, represented in the figure as *maxDistP*, *maxDistA* and *maxDistD*, respectively.
  - $\vec{\sigma}_t(Ag) = [\sigma P_t(Ag), \sigma A_t(Ag), \sigma D_t(Ag)]$, represented in the figure as *stdP*, *stdA* and *stdD*, respectively.

The experiments have been grouped into three situations changing the characteristics of the agents' groups. Moreover, each experiment includes different cases changing

[1] http://jupyter.org

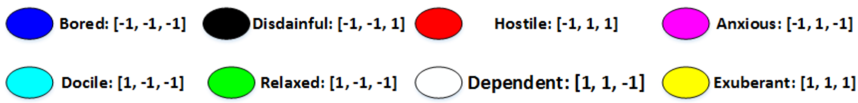


Figure 6.1: Color representation for the different emotions

Table 6.1: Summary of proposed experiments

Experiment	*# agents*		*Empathy*	*Affinity*	*Physical distance*	
					Case 1	*Case 2*
1st	1 group of 10 agents	a) b) c) d)	0 0 1 1	0 1 0 1	All agents have distance 0	All agents have random distances between 0 and 20
2nd	1 group of 10 agents (one agent with Empathy and Affinity = 0)	a) b) c) d)	0 0 1 1	0 1 0 1	All agents have distance 0	All agents have random distances between 0 and 20
3rd	1 group of 5 agents and 1 group of 10 agents	a) b) c) d)	0 0 1 1	0 1 0 1	All agents have distance 0	All agents have random distances between 0 and 20

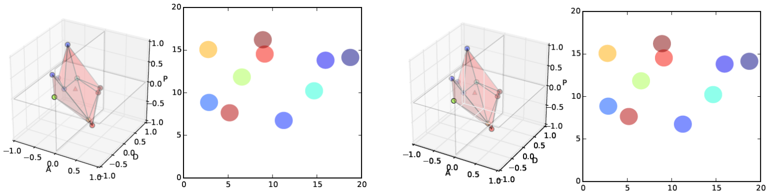
the affinity and empathy levels of the agents and also the physical distance among agents. The different proposed experiments are listed in Table 6.1.

6.5.1 First Experiment

The first experiment tried to evaluate how a group of heterogeneous agents evolve in the emotional space according to the dynamic model. To do this, we implemented a set of 10 agents with a randomized initial emotional state. In order to evaluate the emotional behavior in the agent group, different situations have been defined changing the empathy and affinity values of each agent. Moreover the physical distance has also changed from a minimum distance of 0 meters up to a maximum distance of 20 meters. For reasons of brevity only two of the combinations are described.

First one is the corresponding to all the empathies and affinities between agents to

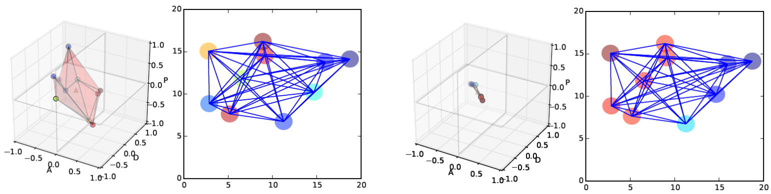
Figure 6.2: One group of 10 agents (with Empathy=0, Affinity=0 and distance between agents >0)



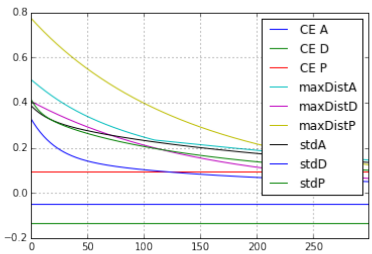
(a) Initial (b) Final

0, that is, a set of agents that has not any relationship between them and that are not moved by the emotions they feel around them. In this situation, the model works as expected, as the agents do not change their emotions. Figure 6.2 shows one execution of this first situation of this example by a PAD space representation and a Physical space position representation for the initial and final stages of the execution.

Figure 6.3: One group of 10 agents (with Empathy=1, Affinity=1 and all the agents with distance >0)



(a) Initial (b) Final



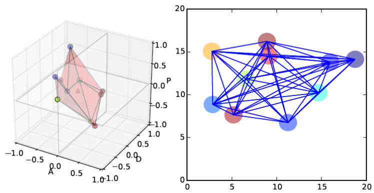
(c) Social emotional evolution

Alternatively, Figure 6.3 represents a situation where agents have a maximum value of the empathy and affinity levels. As we can see, the initial stage for the PAD values of the agents is the same of the previous situation (as can be observed in the corresponding PAD space representation). As this situation has different affinities and empathies, there exists links connecting agents in the Physical space position representation. This situation represents a group of agents that can be considered good friends and very sensitive to their friends emotions. As they are close enough (in a range of [0, 20] meters), their emotions are contaged tending to collapse in the PAD space (as is observed in the Figure 6.4(b) - left). This evolution can be observed, at individual level, in the evolution of the PAD space representation, and in the evolution of the colors of the agents in the PAD space representation and in the Physical space position representation. On the other hand, Figure 6.4(c) shows how fast is the convergence of the social emotional values during the experiment. The relevance of these experiments is the validation that all the situations have the expected behavior according to the proposed model.

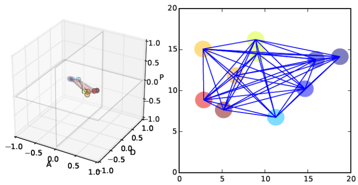
6.5.2 Second Experiment

The second experiment is trying to observe how the emotional state of the group is disturbed by an odd agent without empathy and affinity with any agent. Scenarios proposed in this experiment are affected in the emotional states of the group due to the emotional response generated by the odd agent. As an example we can see the scenario proposed in Figure 6.5(a) where all the agents of the group have the maximum value of the empathy and affinity levels except the odd agent (an initial situation similar to the one used in the Figure 6.4(a)). As we can see, the final situation shows a non perfect grouping of all the agents due to the distorsion caused by the odd agent. This can be observed too in the temporal evolution of the social emotional values, if compared with Figure 6.3.

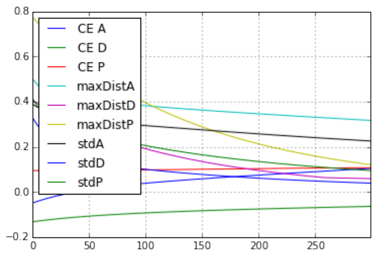
Figure 6.4: One group of 10 agents with an odd agent (with Empathy=1, Affinity=1 and all the agents with distance >0)



(a) Initial



(b) Final



(c) Social emotional evolution

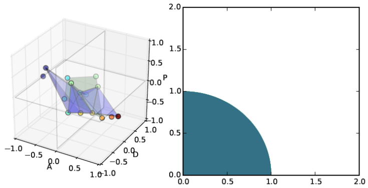
6.5.3 Third Experiment

Finally, the third experiment was centered in analyzing how two disimilar groups of agents change their emotional states following the proposed model.

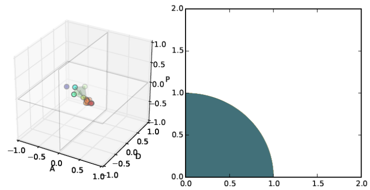
Figure 6.5 represents a scenario where there exists one group of ten agents and another group of five agents with the maximum level of empathy and affinity inside the group and the minimum distance between them. In this case, agents of each group are close to each other as can be expected. Regarding the temporal evolution, it is more evident in the case of the smallest group, where the social emotional parameters are more homogeneous than in the largest group.

To prove the applicability of this model we propose the implementation of it in a mobile robot and proceeded to test different scenarios either in simulated or real environments. The architecture and experiment results are presented in the next

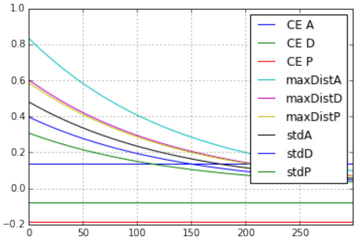
Figure 6.5: One group of 10 agents and another group of 5 agents (with Empathy=1, Affinity=1 and all the agents with distance =0)



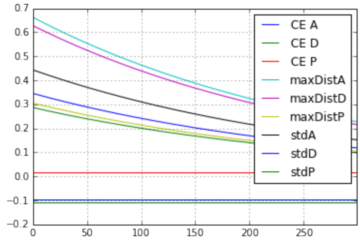
(a) Initial



(b) Final



(c) Social emotional evolution of 5 agents



(d) Social emotional evolution of 10 agents

section.

6.6 A robot guided by emotions.

The model implementation and the proof of concept was done through the use of a real environment, where there is a *NAO* robot [2] in charge of interacting with humans in a room. The main goal of this development is the automatic recognition of the emotional states of a group of individuals in order to enhance the wellbeing of these individuals. To achieve this, the robot moves around the room and tries to interact with any detected person. The robot calculates the emotional states of the identified individuals' group and, according to the proposed model, estimates possible emotional contagions among individuals. In order to make this process

[2]https://www.aldebaran.com/en

it uses different tools to communicate with its environment and to obtain the information that surrounds it:

- Speech recognition, the robot communicates with people to try to change their emotional states. Moreover, if the robot does not know the person, it estimates his personality using a dialogue game that follows the OCEAN test [95].
- Movement, the robot is continually moving around the room trying to interact and stimulate any individual presented in the room.
- Image processing, it is used to detect the emotional state of people around the room. To detect the emotional states, the robot employs a machine learning model explained in the next section.

Figure 6.6 shows a simulated environment of the proposed application where the NAO robot interacts with a group of three individuals.

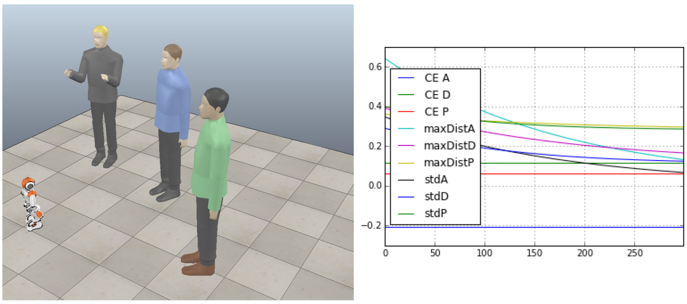


Figure 6.6: Simulation of the proposed application

With this information, the robot tries to stimulate people in the room. These stimulation actions are performed when the robot detects emotional changes that lead the social emotion away from a target emotion (happiness, usually). This continuous sensorization of the environment enables the estimation of the emotional dynamics of the group and the robot is able to react performing different actions like telling a joke, asking what is wrong with them or making a funny movement.

One of the possible applications of this system is in nursing homes, where they have to perform playful tasks. The robot would be responsible for carrying out these tasks while analysing emotions and modifying its actions according to the emotion of the group [106]. As aforementioned, another use is in the iGenda framework. The robot can inform the iGenda of the current emotional status of the environment, thus the iGenda is able to schedule event that please the group.

Due to the complexity of the proposed application, this paper only covers the emotion identification and contagion analysis phases which are the most important phases to the validation of the proposed dynamic emotional model. In the following sections we will explain the process followed for the emotion identification, and then we will show some experiments where we test the real evolution of the emotions dynamic against the simulation of such dynamics from the real emotions perceived.

6.6.1 Emotion identification

In this section we will focus on the design and implementation of the emotion identification of each person which is in the room.

The emotion state detection represents the knowledge about human feelings perceived by the machine *(Robot, Mobile Phones, etc)*. The detection of this emotion is done through a different algorithm, that gives machines this skill and allows them to recognize and classify emotional states. To recognize the emotional state, we can find different algorithms and techniques that extract facial information. Among them, we highlight the *Histogram Oriented Gradient (HOG)* [44], and the *Face Landmark Estimation* [42]. The first one is used to encode these facial components and concatenate them into a single feature vector; the other is the technique we have used, and it extracts a list of points as shown in the Figure 6.7. These points represent the most important characteristics of our faces represented in a 2D plane. Using this information, it is possible to create a set of feature vectors that can be used to train the machine learning models *(ML)*. To determine the best feature vector for the classification of emotions, three experiments were performed. The experiments used two databases, that had the same kind of emotions. The emotions contained in these databases are: *Afraid, Angry, Disgusted, Happy, Neutral, Sad and Surprised*. The first one, called *Karolinska Directed Emotional Faces database (KDEF)*[1], is used to train our *ML* and it is composed of 980 images and the second database, *The Radboud Faces Database (RaFD)* [79], is used to do the test and it is composed of 536 images.

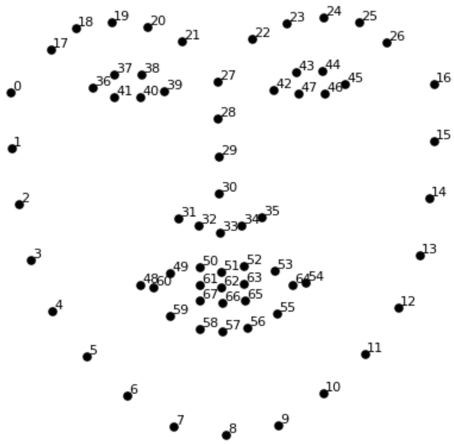


Figure 6.7: List of extracted points, using Face Landmark Estimation

Table 6.2 shows the database distribution and the percentage used to train and to test. These databases are composed by different faces that represent human emotions. To do a good classification, it is necessary to detect a series of face characteristics. These points are showed in Figure 6.8. Based on these points, we calculate the Euclidean distance between each of them. In our experiments, seventeen distances were used, that represent the input used to train the *ML*.

We use agents to represent the human emotions, being able to fully interact with humans. To be able to do this, first, the agent has to possess the ability of classify each emotion, thus possessing a knowledge base with information of a classifier.

Table 6.2: Database distribution.

Database Name	Total Images	Train	Test
KDEF	980	80	20
RafD	469	80	20
KDEF and RafD	1449	80	20

Figure 6.8: Points used to calculate the euclidean distance.

Table 6.3: Description of values used in the machine learning models.

Name	Description
C	Penalty parameter C of the error term.
Gamma	Radius of influence of samples selected by the model as support vectors.
Max Depth	The maximum depth of the tree.
n Estimators	The number of trees in the forest.
Max Features	The number of features to consider when looking for the best split.
Penalty	Used to specify the norm used in the penalization
Degree	Degree of the polynomial kernel function.

Table 6.4: Machine Learning Models and their configuration values.

Name Classifier	C	Gamma	Max Depth	n Estimators	Max Features	Penalty	Degree
SVC Linear Kernel	1.0	0.1	-	-	-	-	-
LinearSVC	1.0	-	-	-	-	-	-
SVC (RBF kernel)	1.0	0.1	-	-	-	-	-
SVC Polynomial (degree 3)	1.0	0.1	-	-	-	-	3
SVC Kernel Sigmoid	1.0	0.1	-	-	-	-	-
Logistic Regression	-	-	-	-	-	l2	-
Logistic Regression 1 My caption	-	-	-	-	-	l2	
Gaussian NB	-	-	-	-	-	-	-
Random Forest Classifier	-	-	14	10	7	-	-

To properly choose from the available methods (neural networks [164], support vector machines [102], etc.), we have performed a comparison between them and chose the one with better results.

Table 6.3 describes the different configuration values used in the machine learning models. Table 6.4 shows the different values of the configuration parameters used in the machine learning models.

The result of the classification can be seen in Table 6.5. The first column corresponds to the *KDEF* database, the second column corresponds to the *RaFD* database and the last one are the two joined databases. In each one of these experiments, the relation between the amount of samples used to train and to test was 80% to train and 20% to test. Other experiments were performed using *Artificial Neural Networks (ANN)*, with the same database and partition (80% for training, 20% for test) as the other ones. The topology used in our *ANN* was defined as: *input layer seventeen neurons*, *in the middle layer one hundred* and *output layer seven neurons*. Using this configuration it was achieved 97% of correct human emotions classification. It is

Table 6.5: Results of the classifiers for each of the databases

Classifier Name	KDEF	RafD	KDEF and RafD
SVC Linear Kernel	63	29	95
LinearSVC	57	30	79
SVC (RBF kernel)	57	30	79
SVC Polynomial	57	29	79
SVC Kernel Sigmoid	59	30	92
Logistic Regression	60	26	86
Logistic Regression 1	65	34	96
Random Forest Classifier	71	35	98
Background Propagation	**75.2**	**65.5**	**98.5**

very important to take into account that all the training process has been made offline, and once the best configuration has been obtained, it is embedded in the agent (in this case, the robot).

Table 6.5 shows that the best results were obtained joining the two databases.

Nevertheless, it is possible to use all the classifiers presented combined as a group of experts. This group of experts classify the emotions and count the number of times each emotion appears. Using this information the agents can determine which is the recognised emotion and the emotional state is added to its knowledge base. Once this emotion has been added, the agents may carry out the necessary actions to interact with the person.

6.6.2 Emotion dynamics experimentation

To validate our dynamic emotional model we have defined several experiments to compare the results obtained in simulation with reality. To make this comparison, it is necessary to have an initial information about the participants. This information will be used as an input parameter in the simulation. To obtain this information, the participant answers a series of questions from a personality test (the *OCEAN*, as previously mentioned). Using this test, it is possible to determine the level of empathy of each participant (using some studies [101] that associates *Agreeableness* component to empathy) and a list of the affinity levels between the participants. The

Table 6.6: Group 1: Empathy levels of each agent.

	Agent 0	Agent 1	Agent 2	Agent 3
Empathy	0.3	0.9	0.6	0.5

Table 6.7: Group 1: Level of affinity between agents.

	Agent 0	Agent 1	Agent 2	Agent 3
Agent 0	0.0	0.8	0.75	0.95
Agent 1	0.79	0.0	0.89	0.85
Agent 2	0.86	0.76	0.0	0.79
Agent 3	0.8	0.8	0.95	0.0

name *Agent* will be used in the experiments to refer to virtual or real participants. The list of affinity levels is normalized between 0 and 1. We have made experiments with two groups of 4 participants (with their affinity levels and empathies), making two experiments (with different initial emotions) with each group. Each experiment was divided in two parts, the first one is a real world execution, where the initial emotions of the participants and their evolution are detected using the above mentioned method. The second part of each experiment is a simulation from the initial emotions detected in the real execution at the beginning of the experiment first part, applying the emotion dynamics model presented in the paper.

6.6.2.1 Group 1

The empathy of the first group or participants are shown in Table 6.6. The friendship level between them is represented in the *affinity matrix* shown in Table 6.7.

In the first experiment, the initial emotion of each participant was detected to be as showed in the Table 6.8. In this table, the different agents' emotions and their corresponding values in *PAD* can be seen.

Figure 6.9-*b* shows the representation of the social emotion dynamics according to the real world execution. These emotions were detected by using the machine learning algorithm presented in previous sections. The delay between captures was of 2 minutes and this process is repeated for one hour. Each emotion detected with

Table 6.8: Group 1 - Experiment 1: Initial Emotion and PAD values of each agent.

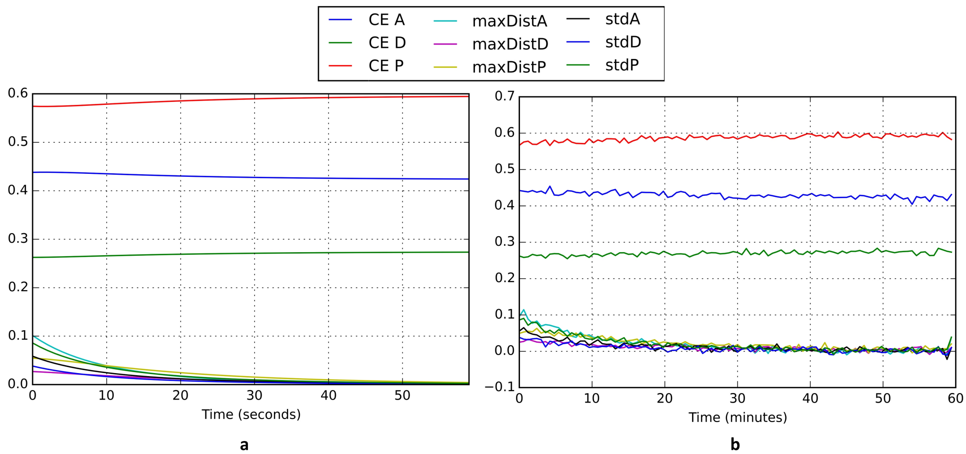
	P	A	D	Emotion
Agent 0	0.63	0.40	0.29	Happy
Agent 1	0.63	0.40	0.28	Happy
Agent 2	0.41	0.55	0.19	Surprise
Agent 3	0.63	0.40	0.29	Happy

our algorithm was transformed in PAD values.

The aim of this simulation is to check how the emotional dynamics work in a group. The Figure 6.9-*a* shows the emotional dynamics for 30 seconds. It can be seen how the agents attract each other. This can be observed in the tendency to zero of the *dispersion (standard deviation)* and the *max distance* values. This tendency indicates that the agents are grouped around a central emotion represented by *CE P, CE A, CE D*, that in this case is happiness.

If we compare the two graphics we can observe that the emotional dynamics in the two examples have had the same behaviour.

Figure 6.9: Group 1 - Experiment 1: High Affinity levels and positive emotions.



The second experiment with this group have the initial emotions situation perceived thatimg can be seen in Table 6.9. This second experiment deals about negative

Table 6.9: Group 1 - Experiment 2: Emotion and PAD values of each agent.

	P	A	D	Emotion
Agent 0	-0.59	0.08	0.47	Angry
Agent 1	-0.59	-0.01	0.40	Disgusted
Agent 2	-0.08	0.18	-0.39	Fearful
Agent 3	-0.28	-0.12	-0.37	Sad

emotions' evolution.

In a similar way as in the first experiment, this one was first executed in the real world, and the emotional evolution perceived in Figure6.10-*b*.

In the Figure 6.10-*a* we can see the simulation evolution of the emotional dynamics using as initial values of the agents' emotions the ones in Table 6.9.

As this group of participants possess a big empathy and affinity level between them, in both experiments, the group tends to be emotionally cohesive. This can be seen in the figures as the *dispersion (standard deviation)* and the *maximum distance* tend to zero. In this second experiment, the *central emotion* is around *Disgusted* emotion.

Figure 6.10: Group 1 - Experiment 2: High Affinity levels and negative emotions.

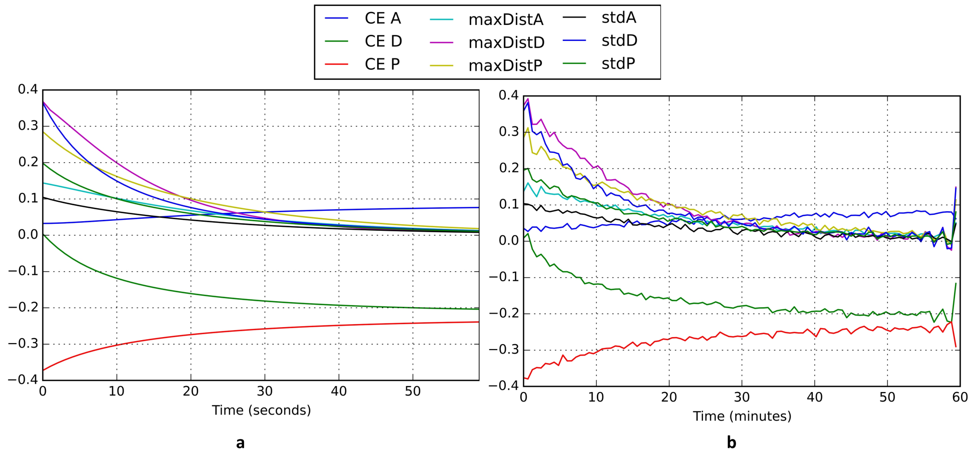


Table 6.10: Group 2: Empathy levels to each agent.

	Agent 0	Agent 1	Agent 2	Agent 3
Emphaty	0.1	0.5	0.8	0.95

Table 6.11: Group 2: Level of affinity between agents.

	Agent 0	Agent 1	Agent 2	Agent 3
Agent 0	0.0	0.1	0.01	0.05
Agent 1	0.09	0.0	0.04	0.06
Agent 2	0.01	0.06	0.0	0.07
Agent 3	0.08	0.02	0.03	0.0

6.6.2.2 Group 2

This second group of participants is formed by a group of 4 people that are practically unknown between them (this can be seen in Table 6.11 where the affinity matrix of this group is shown). Table 6.10 shows the empathies of such people.

The first experiment with this second group have the initial emotions perceived that are seen in Table 6.12.

Figure 6.11-*b* shows the social emotion dynamics in the real world execution of this firs experiment with the second group of participants. The data acquisition was similar to the one made in the first group experiments.

Figure 6.11-*a* shows the results of the social emotion dynamics' simulation using as initial emotion values the ones perceived in the real scenario.

In this first experiment the group maintains its emotional state very close to happiness. In the same way, the dispersion values and maximum distances are very close to zero, indicating that the group is responding positively. This behaviour is

Table 6.12: Experiment 2A: Initial Emotion and PAD values of each agent.

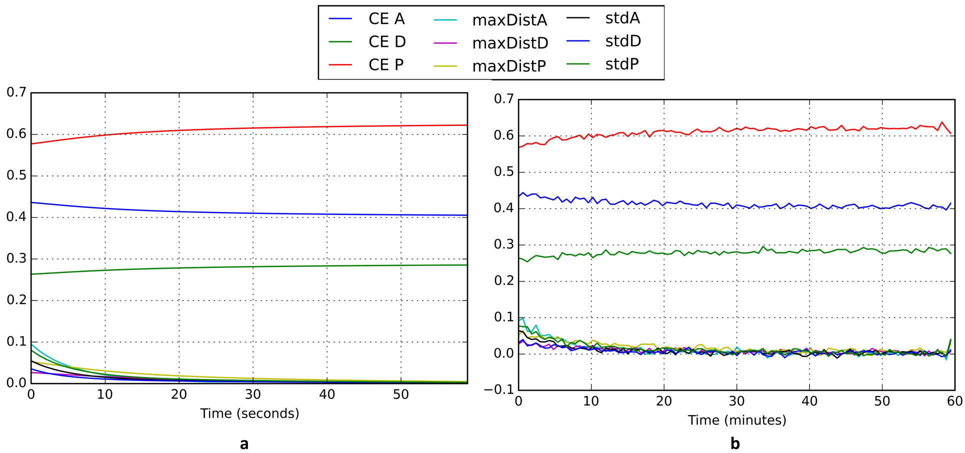
	P	A	D	Emotion
Agent 0	0.63	0.40	0.29	Happy
Agent 1	0.63	0.40	0.28	Happy
Agent 2	0.41	0.55	0.19	Surprise
Agent 3	0.63	0.40	0.29	Happy

Table 6.13: Group 2 - Experiment 2: Initial Emotion and PAD values of each agent.

	P	A	D	Emotion
Agent 0	-0.08	0.18	-0.39	Fear
Agent 1	-0.59	0.08	0.47	Angry
Agent 2	-0.59	-0.01	0.40	Disgusted
Agent 3	-0.28	-0.12	-0.37	Sad

not produced by the affinity levels, since in this experiment they are very low. This behaviour is caused by the individuals empathy levels, as two of them have high empathy values.

Figure 6.11: Group 2 - Experiment 1: Low Affinity levels and positive emotions.

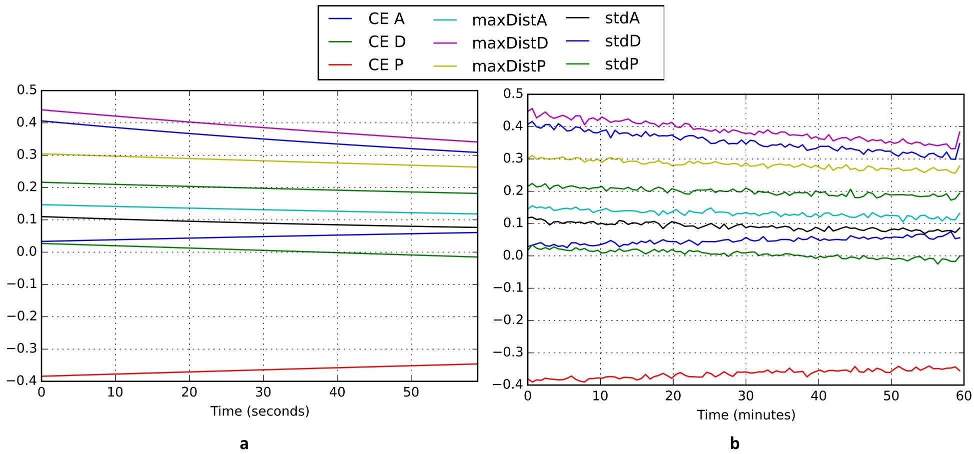


For the second experiment of the second group of participants, the initial emotions perceived can be seen in the Table 6.13.

The results of this experiment in the real world can be seen in the Figure6.12-*b*, and the simulation of the social emotion dynamics in the Figure6.12-*a*.

In this second experiment of the second group, we can see that the emotional dispersion and the maximum distance are high. This happens because the levels of affinity between the agents are low. This behaviour occurs both in simulation and in the real world.

Figure 6.12: Group 2 - Experiment 2: Low Affinity levels and negative emotions.



In the end, we may observe that the social emotion dynamics model seems to work quite properly, as it is able to predict the emotion dynamics, using the affinity levels, empathies and the emotions detected by the cameras. This could be used to anticipate the different actions that can be done so that the group may move towards an emotion (or avoid to reach one).

6.7 Conclusions and future work

A new model for the calculation of dynamic emotions has been presented in this paper, showing a first approach for the emotional contagion and simulation of dynamic social emotions into a group of intelligent entities. The proposed model uses the personality of each entity and the affinity level between entities in order to calculate and represent the emotional dynamic of a group. The dynamic emotional model of a group of agents not only allows a global view of the emotional dynamic of the group, also can improve the decision making based on the attraction level between entities.

Specifically, the proposed model uses the dynamic *Newton Law* and universal gravitation law, to calculate the attraction level ($\vec{EAF}_t(ag_i)$) and the new emotion of each agent ($\vec{E}_{t+1}(ag_j) = \vec{E}_t(ag_j) + (\vec{v}_t(ag_i) \cdot t)$). These definitions allow to calculate the emotional attraction between entities or groups. Moreover, it is possible to obtain the resulting emotion of the attraction in a $(t+1)$, as well as the emotional propagation velocity ($\vec{v}_t(ag_i)$) time. Considering these elements it is possible to know how is the emotional distribution among the agent group and to use this information to reason about future decisions.

The model evaluation was done through the use of a mobile robot application. Specifically, the proposed application consists of a NAO mobile robot that tries to interact with a group of people in a room. Results show that if the robot implements our proposed model, it is able to estimate the dynamic behaviour of people from an emotional point of view. Using these estimations, the robot may be able to enhance its decision-making process.

As future work, we want to introduce emotion recognition using physiological signals. Using this information we predict that it is possible of enhancing the emotion detection and improving the detection time of the human emotion.

6.8 Acknowledgements

This work is partially supported by the MINECO/FEDER TIN2015-65515-C4-1-R and the FPI grant AP2013-01276 awarded to Jaime-Andres Rincon. This work is supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT - Fundacao para a Ciencia e Tecnologia within the projects UID/CEC/00319/2013 and Post-Doc scholarship SFRH/BPD/102696/2014 (A. Costa).

Part III

Discussion

CHAPTER

7

General Discussion of the Results

The objective of this chapter is to summarize the main contributions of this PhD thesis presented in Part II. This chapter is organized as follows: section 7.1 summarizes the main features of JaCalIVE framework; section 7.2 presents the contributions related to the proposed social emotional model based on a static emotional model; section 7.3 shows the improvement of previous emotional model by the introduction of a dynamic emotional model; and, finally, section 7.4 presents the integration of proposed emotional models into the JaCalIVE framework to create Emotional Intelligent Virtual Environments (EIVE).

7.1 JaCalIVE framework

One of the first steps of this research work was to define and implement an appropriated framework for the development of MAS-based IVE. This framework has been used as development platform for testing several case studies proposed in the selected papers.

Chapter 2 presents the design of a MAS-based IVE using JaCalIVE framework based on the MAM5 meta-model [14]. MAM5 defines that an IVE must be constituted by several entities including those that are defined as artifact, meaning they have no intelligence in the system (inanimate objects such as a wall or a table) and can be used by agents as tools to interact within the IVE. MAM5 goes one step further in the design of IVEs dividing the IVE into two sections: non-virtual physical simulated and virtual physical simulated. In each one of these sections different types of entities can be found. In the non-virtual section, the developer can define entities that do not have a virtual representation. In the virtual section, physical laws and the representation of entities that have a 3D body can be defined. In this sense, an IVE is composed of three development modules: artifacts, agents and a physical simulation. In our proposal, artifacts are modeled as physical elements through which the environment is accessible and agents are defined as the intelligent entities becoming part of it.

The JaCalIVE framework was tested in a scenario for the simulation of modular robots controlled by agents. The proposed application was used to probe the versatility of JaCalIVE. The modular robots simulation composed of ten modular robot models act as a system that behave as a genetic algorithm, where the chromosomes will be the different robots. The system can be seen as an evolutive algorithm which allows the automatic adaptation of modular robots. At the end of the process, the system was able to set a configuration based on the ability of robots to adapt their shape depending on changing environment conditions.

Another contribution of the JaCalIVE framework development is the inclusion of the human in the loop and the connection of the physical environment with the virtual world, and so enabling the creation of augmented worlds. Chapter 3 describes the improvement of the proposed framework by integrating smart resources artifacts and human immersed agents to achieve a better design of the physical world in IVEs. This new elements facilitate the design of models that can capture the real world with a high level of abstraction, and even allows agents to communicate more easily with the outside world. Moreover, the proposed extension of JaCalIVE enables modeling of human so they can interact with other entities in the IVE.

7.2 Social emotional model

One of the main results of this PhD thesis was to find a way to integrate affective components to the proposed framework. In this sense, Chapter 4 gives a first approximation of an emotional model for the representation of social emotions and their integration within JaCalIVE. This model has mainly two aspects to highlight. First, it allows the calculation of the social emotion of a human-agent society, and secondly, it allows the representation of the emotional distance that exists within two or more groups of agents. Moreover, the model adds the mechanisms to compare the social emotion of a group in different instants in time. Therefore, the social emotion can be used as a feedback in the own system in order to influence in the emotional state of the group or only of a part of the agents.

On the other hand, Chapter 5 shows an application of the proposed model. The main goal was to show how agents learn and detect human emotions and also are able to recognize humans. In this way, the proposed system was able to obtain and analyze the social emotion of a group of persons who are listening music. Moreover, the system can take decisions according to that emotional state. The application was developed over the JaCalIVE framework allowing an easy integration of the human in the multi-agent system and a visualization of the system in a virtual environment. One of the highlights of the obtained system was that it can be used in an online fashion, reflecting what is happening in the real world or it can be used to simulate/predict what would happen if the conditions, that are used in the simulation, change in some way.

7.3 Dynamic emotional model

In order to improve social simulation processes in agent societies, it is necessary to represent collective emotions and emotion's dynamicity. Therefore, a particular target of this thesis was to introduce a dynamic emotional model that tried to learn from the entities actions and the environment changes.

Emotions are not static, and they can be propagated affecting other people. This phenomenon is known as emotional contagion. In order to try to replicate this behavior, three key elements had to be taken into account. The first one is the empathy, defined as a feeling of affective participation of one person when another is affected. The second one is the level of affinity of the individual, with respect to the other members of the group. And, finally, the third key is the physical distance between individuals. Taking into account these elements, in chapter 6 a dynamic emotional model was presented, which can be considered as a first approach for the representation of the emotional contagion and the simulation of the social emotions of a group of agents that dynamically change their emotional states. From our point of view, the proposed dynamic emotional not only allows a global view of the emotional dynamic, but also improves the decision making in the system based on the level of attraction among entities. This dynamic model was tested in a mobile robot application. The main goal of the proposed application was the identification and management of the emotional states of a group of individuals by a mobile robot (specifically, a NAO robot) in order to enhance the wellbeing of the group. During the experiment, the robot calculated the emotional states of the individuals estimating a possible emotional contagion among them. This information was used by the robot to decide possible new actions to improve (in some sense) the estimated emotional states.

7.4 Emotional Intelligent Virtual Environments

Finally, it is necessary to emphasize the integration of the different emotional models into the JaCalIVE framework in order to give support for the development of the proposed applications. Specifically, JaCalIVE was extended with well-known emotional models, such as PAD or Circumplex-Model, and the BIG-Five personality model. Moreover, the framework was also extended with the emotional models presented in this thesis. These extensions allowed the modeling and simulation of emotional intelligent virtual environments (EIVE). Within an EIVE it is possible to simulate artificial intelligent entities expressing emotions and, at the same time, being able to detect human emotions using devices such as video cameras or wearable devices. For this purpose, the JaCalIVE framework is equipped with the necessary elements to create complex simulations taking into account the user emotions.

Part IV

Conclusion

CHAPTER

8

Conclusions and future work

8.1 Contributions

This PhD contributes with a framework for the development of MAS-based IVE and a social emotional model for extracting the emotion from a group of agents, as well as, inducing emotional contagion among them to achieve more realistic applications involving human-agent interfaces. The contents of this document present the chronological achievements and successive refinements of JaCalIVE framework and the inclusion of the final version of the social emotional model witch notably improves human-computer interaction in emotional-based applications.

We highlight several main strengths of this research work:

- We reviewed the state of the art about emotions in multi-agent systems. This review was made with specially emphasis on the use of emotions from a social perspective, and their propagation process among different entities. Previous works revealed to us that there were some limitations in current approaches when it came to determine the social emotion and emotional propagation in a human-agent society. For example, in the area of crowd simulations, we could only find small advances related to emotional propagation, which focus only on basic emotions such as fear (emotion often generated in cases of fire or catastrophes). At the same time, we realized that there was no appropriate framework for MAS that incorporated emotional models to identify, classify and/or simulate human emotions.
- We proposed a new development framework for the design and simulation of applications in which intelligent entities can recognize and/or simulate human emotions. This framework, called JaCalIVE, allows to design and implement IVEs and it differs from other works in the sense that it integrates the concept of agents, artifacts and physical simulation. Furthermore, IVEs developed with JaCalIVE can be easily modified thanks to the XML modeling and the automatic code generation. JaCalIVE follows the MAM5 perspective, witch means that modules used to interact with the developed IVEs are uncoupled

from the rest of the system, enabling easy integration of several modules as needed. JaCalIVE framework was tested on a practical robotics application which demonstrated its versatility, flexibility and adaptability to be used in any application context.

- We presented a JaCalIVE framework extension to include the human in the loop and connect the physical environment with the virtual world. This new approach included an extension of the MAM5 meta-model. Further elements were introduced such as smart resources artifacts and human immersed agents with the idea of helping designers to create augmented worlds and improve the development of IVEs.

- We defined a social emotional model taking into account the emotions expressed by a set of humans and/or agents. The model presents a first approach for the detection and simulation of social emotions in a group of intelligent entities. The model was built based on the PAD emotional model. Moreover, It allows to compare the emotional state between two groups of agents, or the social emotion of a single group at different moments of time. The model was tested in a simple experiment of social interaction between agents. The results obtained reflected the improvement of decision-making process in the system by using the learning module to predict the emotional state of the people.

- We proposed a dynamic emotional model to represent the emotional contagion in a MAS. Specifically, the model allows to determine emotional propagation within a group of agents and humans. Our proposed model include the personality of each entity and the affinity level between entities in order to calculate and represent the emotional dynamics of a group. This model improves the decision-making process based on the attraction level between entities, as well as the emotional propagation velocity. This way, it is possible to get the emotional distribution among entities and use it to reason about possible future actions.

These contributions have been tested using several case of study based on emotional-based applications. For example, with the AmI application it was possible to extract and analyze the social emotion of a group of persons and take decisions according to it. This case of study incorporated an agent capable of classifying music by genres perceiving its surroundings. Once the musician genre was qualified, the actor was able to determine the social emotion of the group using the preferences of the users, and further modify the music that sounded. following the same idea, another study case was proposed for detecting emotion using image processing. Each agent was associated with a human employing facial identification, so that each agent became a virtual representation of the user. The last case of study included a SH application where there was a NAO robot in charge of interacting with humans in a real environment. The robot was able to estimate the dynamic behavior of people from an emotional point of view by using the proposed dynamic emotional model.

At the end, the proposed emotional-based applications showed that the incorporation of an social emotional model improved the interaction between humans and agents living within IVEs.

8.2 Future Work

Due to the novelty of the topic, the use of emotions into artificial intelligent entities opens new areas of research and development, being ambient intelligence (AmI) and robotics some of the most important. Typically, these areas use emotions as an input to improve the interaction processes with the humans. These improved interactions allow to reduce the distance between humans and machines. According to these, we describe some of the future lines that we consider as potentially interesting for future research:

- Wristband. The way that humans perceive the world influences their emotional state that has a repercussion on the physical level. While most can hide facial expressions and body movements (like hiding a state of surprise) there are low-level signals that the human body sends inadvertently like skin/muscle tensioning, pupil dilatation and micro-movements. Most of these signals have a corresponding bio-electrical impulse, thus they can be captured by sensors. We propose the usage of wearable devices in form of wristbands, that feed information for the detection of the human's emotions. The wristband captures the different bio-signals, that will be used to detect the emotion of a human being; and can predict the emotional state of the individual from the processed biosignals employing classifier algorithms.

- Robot. This thesis opens a new research field in robotics, especially in assistant robotics. This topic is maybe one of the most studied right now, since, in recent year we can see the introduction of robots in homes. These robots have the capability of navigate in our house, detect persons and identify the different persons that live in the house. However, the use of emotions in this kind of robot is relatively new. In the literature we can find few robots that introduce the emotions recognition, but only as a way to interact with one person, nonetheless, these robots interact with more than one person and current models don't take into account this aspect. That's why we propose as a future work to introduce the proposed social emotional model and the dynamic model in this type of robots.

  This robot needs to have some special features to facilitate interaction with humans such as face identification, emotions recognition, locomotion, input devices and output information (screens, headphones, microphones, etc.). However, it is possible to add different sensors such as CO_2, gas, air quality, among others. Also, it is important that the robot has as much as possible a pleasant and anthropomorphic face, as recent studies show that they are more accepted by people.

The first prototype of the robot commented was presented in the 15th International Conference on Practical Applications of Agents and Multi-Agent Systems, winning the first prize for best practical agent based applications. This prize along with the work on emotions and robotics presented in this thesis got a media repercussion that can be summarized as:

- 96 articles in newspapers.
- 6 radio interviews.
- 2 national TV interviews.

Bibliography

[1] (2008). The Karolinska Directed Emotional Faces: A validation study. *Cognition and Emotion*, **22**, 1094–1118.

[2] Adam, C., Canal, R., Gaudou, B., Vinh, H. T., Taillandier, P., and Others (2012). Simulation of the Emotion Dynamics in a Group of Agents in an Evacuation Situation . In *Principles and Practice of Multi - Agent Systems*, pages 604–619. Springer.

[3] Al-Zinati, M., Araujo, F., Kuiper, D., Valente, J., and Wenkstern, R. Z. (2013). Divas 4.0: A multi-agent based simulation framework. In *Proceedings of the 2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications*, pages 105–114. IEEE Computer Society.

[4] Ali, F. and Amin, M. (2013). The influence of physical environment on emotions, customer satisfaction and behavioural intentions in Chinese resort hotel industry. In *J. for Global Business Advancement*, volume 7, pages 249 – 266.

[5] Andreoli, R., De Chiara, R., Erra, U., and Scarano, V. (2005). Interactive 3d environments by using videogame engines. In *Ninth International Conference on Information Visualisation*, pages 515–520. IEEE.

[6] Araujo, F., Valente, J., Al-Zinati, M., Kuiper, D., and Zalila-Wenkstern, R. (2013). Divas 4.0: a framework for the development of situated multi-agent based simulation systems. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1351–1352. International Foundation for Autonomous Agents and Multiagent Systems.

[7] Aylett, R. and Luck, M. (2000). Applying artificial intelligence to virtual reality: Intelligent virtual environments. *Applied Artificial Intelligence*, **14**, 3–32.

[8] Banarjee, S., Grosan, C., and Abraham, A. (2005). Emotional ant based modeling of crowd dynamics. In *Symbolic and Numeric Algorithms for Scientific Computing , 2005. SYNASC*, pages 8—-pp. IEEE.

[9] Barbero, Á., González-Rodríguez, M. S., de Lara, J., and Alfonseca, M. (2007). Multi-Agent Simulation of an Educational Collaborative Web System. In *European Simulation and Modelling Conference*, pages 1–8.

[10] Barella, A., Carrascosa, C., and Botti, V. (2007a). Agent Architectures for Intelligent Virtual Environments. In *Intelligent Agent Technology, 2007. IAT '07. IEEE/WIC/ACM International Conference on*, pages 532–535. IEEE.

[11] Barella, A., Carrascosa, C., Botti, V., Martí, M., Mart, M., Barella, A., Carrascosa, C., Botti, V., and Martí, M. (2007b). Multi-agent systems applied to virtual environments: a case study. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, volume 1, pages 237–238. ACM.

[12] Barella, A., Valero, S., and Carrascosa, C. (2009). JGOMAS: New approach to AI teaching. *IEEE Transactions on Education*, **52**(2), 228–235.

[13] Barella, A., Ricci, A., Boissier, O., and Carrascosa, C. (2012a). MAM 5: Multi-Agent Model For Intelligent Virtual Environments. In *10th European Workshop on Multi-Agent Systems (EUMAS 2012)*, pages 16–30.

[14] Barella, A., Ricci, A., Boissier, O., and Carrascosa, C. (2012b). Mam 5: Multi-agent model for intelligent virtual environments. In *10th European Workshop on Multi-Agent Systems (EUMAS 2012)*, pages 16–30.

[15] Barsade, S. G. (2002). The ripple effect: Emotional contagion and its influence on group behavior. *Administrative Science Quarterly*, **47**(4), 644–675.

[16] Bechara, A., Damasio, H., Damasio, A. R., and Lee, G. P. (1999). Different contributions of the human amygdala and ventromedial prefrontal cortex to decision-making. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, **19**(13), 5473–81.

[17] Becker-Asano, C. and Wachsmuth, I. (2010). Affective computing with primary and secondary emotions in a virtual human. *Autonomous Agents and Multi-Agent Systems*, **20**(1), 32–49.

[18] Behrens, T. M., Hindriks, K. V., and Dix, J. (2011). Towards an environment interface standard for agent platforms. *Ann Math Artif Intell*, **61**(4), 261–295.

[19] Bergeron, B. P. (2006). *Developing serious games*. Charles River Media.

[20] Billhardt, H., Julián, V., Corchado, J. M., and Fernández, A. (2014). An Architecture Proposal for Human-Agent Societies. In *Highlights of Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection*, volume 430 of *Communications in Computer and Information Science*, pages 344–357. Springer International Publishing.

[21] Billhardt, H., Julián, V., Corchado, J. M., and Fernández, A. (2015). Human-agent societies: challenges and issues. *International Journal of Artificial Intelligence*™, **13**(1), 28–44.

[22] Boeing, A. and Bräunl, T. (2007). Evaluation of real-time physics simulation systems. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, pages 281–288. ACM.

[23] Bordini, R. H., Hübner, J. F., and Wooldridge, M. J. (2007). *Programming multi-agent systems in AgentSpeak using Jason*. J. Wiley, Chichester, England; Hoboken, NJ.

[24] Bos, D. O. (2006). EEG -based emotion recognition. *The Influence of Visual and Auditory Stimuli*, pages 1–17.

[25] Bosse, T., Duell, R., Memon, Z. A., Treur, J., and Van Der Wal, C. N. (2009). A multi-agent model for emotion contagion spirals integrated within a supporting ambient agent model. In *Principles of Practice in Multi - Agent Systems*, pages 48–67. Springer.

[26] Bosse, T., Hoogendoorn, M., Klein, M. C. A., Treur, J., van der Wal, C. N., and van Wissen, A. (2013). Modelling collective decision making in groups and crowds: Integrating social contagion and interacting emotions, beliefs and intentions. *Autonomous Agents and Multi-Agent Systems*, **27**(1), 52–84.

[27] Broekens, J. (2007). Emotion and reinforcement: Affective facial expressions facilitate robot learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **4451 LNAI**, 113–132.

[28] Busso, C., Deng, Z., and et al (2004). Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, ICMI '04, pages 205–211, New York, NY, USA. ACM.

[29] Canento, F., Fred, A., Silva, H., Gamboa, H., and Lourenço, A. (2011). Multimodal biosignal sensor data handling for emotion recognition. In *Sensors, 2011 IEEE*, pages 647–650. IEEE.

[30] Castrillón-Santan, M., Lorenzo-Navarro, J., and Hernández-Sosa, D. (2014). Conteo de personas con un sensor RGBD comercial. *Revista Iberoamericana de Autom á tica e Inform á tica Industrial RIAI*, **11**(3), 348–357.

[31] Cheng, F. F. (2013). Design of Remote Control System. In *Advanced Materials Research*, volume 722, pages 397–401. Trans Tech Publ.

[32] Chong, C.-Y. and Kumar, S. P. (2003). Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, **91**(8), 1247–1256.

[33] Compton, M., Barnaghi, P., Bermudez, L., Garc\'\iA-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., and Others (2012). The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web*, **17**, 25–32.

[34] Cook, D. J. and Das, S. K. (2007). How smart are our environments? An updated look at the state of the art. *Pervasive and Mobile Computing*, **3**(2), 53–73.

[35] Corchado, E., WoÅºniak, M., Abraham, A., de Carvalho, A. C., and SnÃ¡Å¡el, V. (2014). Recent trends in intelligent data analysis. *Neurocomputing*, **126**(0), 1–2. Recent trends in Intelligent Data Analysis Selected papers of the The 6th International Conference on Hybrid Artificial Intelligence Systems (HAIS 2011) Online Data Processing Including a selection of papers from the International Conference on Adaptive and Intelligent Systems 2011 (ICAIS 2011).

[36] Costa, Â., Castillo, J. C., Novais, P., Fernández-Caballero, A., and Simoes, R. (2012). Sensor-driven agenda for intelligent home care of the elderly. *Expert Systems with Applications*, **39**(15), 12192–12204.

[37] Costa, A., Novais, P., and Simoes, R. (2014). A caregiver support platform within the scope of an ambient assisted living ecosystem. *Sensors*, **14**(3), 5654–5676.

[38] Council on Communications and Media (2009). Impact of Music, Music Lyrics, and Music Videos on Children and Youth. *PEDIATRICS*, **124**(5), 1488–1494.

[39] Cui, L., Wang, X., and Zhang, Y. (2016). Reinforcement learning-based asymptotic cooperative tracking of a class multi-agent dynamic systems using neural networks. *Neurocomputing*, **171**, 220–229.

[40] de O. Saraiva, F., Bernardes, W. M. S., and Asada, E. N. (2015). A framework for classification of non-linear loads in smart grids using Artificial Neural Networks and Multi-Agent Systems. *Neurocomputing*, **170**, 328–338.

[41] Delac, K., Grgic, M., and Grgic, S. (2005). Statistics in face recognition: analyzing probability distributions of PCA, ICA and LDA performance results. *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 289–294.

[42] Dhall, A., Goecke, R., Joshi, J., Wagner, M., and Gedeon, T. (2013). Emotion Recognition in the Wild Challenge (EmotiW) Challenge and Workshop Summary. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 371–372, New York, NY, USA. ACM.

[43] Dimuro, G. P. G. P., Costa, A. C. d. R., Gon\textbackslashtextbackslashccalves, L. V., Hubner, A., Gon\textbackslashccalves, L. V., Hubner, A., Gon\textbackslashtextbackslashccalves, L. V., and Hubner, A. (2008). Interval-valued hidden markov models for recognizing personality traits in social exchanges in open multiagent systems.

[44] Donia, M. M. F., Youssif, A. A. A., and Hashad, A. (2014). Spontaneous Facial Expression Recognition Based on Histogram of Oriented Gradients Descriptor. *Computer and Information Science*, **7**, 31–37.

[45] Dos Santos, C. T. and Osório, F. S.-b. (2004). An intelligent and adaptive virtual environment and its application in distance learning. In *Proceedings of the working conference on Advanced visual interfaces AVI 04*, page 362.

[46] Edle, M. J. S., Thakare, A. P., and Agarkar, A. M. (2011). Intelligent Illumination System to Prevail over Possible Diseases due to over-and.under-illumination. *International Journal of Enterprise Computing and Business Systems Systems*, **1**(2), 1–15.

[47] Egges, A., Kshirsagar, S., and Magnenat-Thalmann, N. (2004). Generic personality and emotion simulation for conversational agents. *Comp. Anim. Virtual Worlds*, **15**(1), 1–13.

[48] Esparcia, S., Sánchez-Anguix, V., and Aydogan, R. (2013). A negotiation approach for energy-aware room allocation systems. In *1st Workshop on Conflict Resolution in Decision Making (COREDEMA 2013)*, volume 365, pages 280–291. Springer.

[49] Fischinger, D., Einramhof, P., Papoutsakis, K., Wohlkinger, W., Mayer, P., Panek, P., Hofmann, S., Koertner, T., Weiss, A., Argyros, A., and Vincze, M. (2016). Hobbit, a care robot supporting independent living at home: First prototype and lessons learned. *Robotics and Autonomous Systems*, **75**, 60–78.

[50] Fox, N. A. and Calkins, S. D. (2003). The development of self-control of emotion: Intrinsic and extrinsic influences. *Motivation and emotion*, **27**(1), 7–26.

[51] Garca-Pardo, J. A., Carrascosa, C., Garcia-Pardo, J. A., Carrascosa, C., and García-Pardo (2011). Social welfare for automatic innovation. In *Multiagent System Technologies*, volume 6973, pages 29–40. Springer.

[52] Gelernter, D. (1992). *Mirror worlds - or the day software puts the universe in a shoebox: how it will happen and what it will mean*. Oxford University Press.

[53] Go, C. A. L. and Lee, W.-H. (2007). An Intelligent Belief-Desire-Intention Agent for Digital Game-Based Learning. In *Analysis and Design of Intelligent Systems using Soft Computing Techniques*, pages 677–685. Springer.

[54] GOELEVEN, E., De Raedt, R., LEYMAN, L., and Verschuere, B. (2008). The Karolinska Directed Emotional Faces: A validation study. *COGNITION & EMOTION*, **22**(6), 1094–1118.

[55] Gonzalez-Gomez, J., Gonzalez-Quijano, J., Zhang, H., and Abderrahim, M. (2010). Towards the sense of touch in snake modular robots for search and rescue

operations. In *ICRA 2010 Workshop "Modular Robots: State of the Art"*, pages 63–68.

[56] Gonzalez-Jorge, H., Riveiro, B., Vazquez-Fernandez, E., Mart\'\inez-Sánchez, J., and Arias, P. (2013). Metrological evaluation of Microsoft kinect and asus xtion sensors. *Measurement*, **46**(6), 1800–1806.

[57] Gouaïch, A., Michel, F., and Guiraud, Y. (2005). *MIC*: A Deployment Environment for Autonomous Agents*, pages 109–126. Springer Berlin Heidelberg, Berlin, Heidelberg.

[58] Griffiths, P. (1997). *What emotions really are : the problem of psychological categories*. University of Chicago Press.

[59] Guinard, D., Trifa, V., Mattern, F., and Wilde, E. (2011). From the internet of things to the web of things: Resource-oriented architecture and best practices. In *Architecting the Internet of Things*, pages 97–129. Springer.

[60] Hale, K. S. and Stanney, K. M. (2002). *Handbook of Virtual Environments: Design, Implementation, and Applications*. Human Factors and Ergonomics. Taylor & Francis.

[61] Han, D.-M. and Lim, J.-H. (2010). Smart home energy management system using IEEE 802.15. 4 and zigbee. *Consumer Electronics, IEEE Transactions on*, **56**(3), 1403–1410.

[62] Hancher, M. D. and Hornby, G. S. (2006). A modular robotic system with applications to space exploration. *Space Mission Challenges for Information Technology*, **0**, 8.

[63] Hawe, G. I., Coates, G., Wilson, D. T., and Crouch, R. S. (2012). Agent-based simulation for large-scale emergency response: A survey of usage and implementation. *ACM Computing Surveys (CSUR)*, **45**(1), 8.

[64] Holzapfel, A. and Stylianou, Y. (2007). A statistical approach to musical genre classification using non-negative matrix factorization. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 2, pages II—-693. IEEE.

[65] Intille, S. S. (2002). Designing a home of the future. *IEEE pervasive computing*, **1**(2), 76–82.

[66] Ioannou, S. V., Raouzaiou, A. T., Tzouvaras, V. A., Mailis, T. P., Karpouzis, K. C., and Kollias, S. D. (2005). Emotion recognition through facial expression analysis based on a neurofuzzy network. *Neural Networks*, **18**(4), 423–435.

[67] Jain, D. and Kobti, Z. (2011). Simulating the Effect of Emotional Stress on Task Performance Using OCC. In C. Butz and P. Lingras, editors, *Advances in Artificial Intelligence*, volume 6657 of *Lecture Notes in Computer Science*, pages 204–209. Springer.

[68] James, W. (2007). *What Is an Emotion?* Wilder Publications.

[69] Jenison, R. L., Schoffelen, J., Brunet, N., Oostenveld, R., and Bastos, A. (2014). Directional influence between the human amygdala and orbitofrontal cortex at the time of decision-making. *PLoS ONE*, **9**(10), e109689.

[70] Jia, L. and Zhenjiang, M. (2007). Entertainment Oriented Intelligent Virtual Environment with Agent and Neural Networks. In *IEEE International Workshop on Haptic, Audio and Visual Environments and Games, 2007. HAVE 2007*, pages 90–95.

[71] Journal, I., Technological, F., and Singla, K. (2014). Audio noise reduction using different filters 1,2. **1**(11), 1373–1375.

[72] Kavitha, R. and Thiyagarajan, N. (2014). Distributed intelligent street lamp monitoring and control system based on zigbee. *International Journal of Science and Research (IJSR) PP*, pages 2319–7064.

[73] Kazemi, A., Zarandi, M. H. F., and Husseini, S. M. M. (2009). A multi-agent system to solve the production–distribution planning problem for a supply chain: a genetic algorithm approach. *Int J Adv Manuf Technol*, **44**(1-2), 180–193.

[74] Kim, M. H., Joo, Y. H., and Park, J. B. (2005). Emotion detection algorithm using frontal face image. In *International Conference on Computer Application in Shipbuilding*.

[75] Kitano, H., Tadokoro, S., Noda, I., Matsubara, H., Takahashi, T., Shinjou, A., and Shimada, S. (1999). Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*, volume 6, pages 739–743. IEEE.

[76] Koenig, N. and Howard, a. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, **3**, 2149–2154.

[77] Koole, S. L. (2009). The psychology of emotion regulation: An integrative review. *Cognition & Emotion*, **23**(1), 4–41.

[78] Kushner, D. (2011). The making of arduino. *IEEE Spectrum*, **26**.

[79] Langner, O., Dotsch, R., Bijlstra, G., Wigboldus, D. H. J., Hawk, S. T., and van Knippenberg A. (2010). Presentation and validation of the Radboud Faces Database. Cognition and Emotion, 24(8), 1377–1388.

[80] Leccese, F., Cagnetti, M., and Trinca, D. (2014). A Smart City Application: A Fully Controlled Street Lighting Isle Based on Raspberry-Pi Card, a ZigBee Sensor Network and WiMAX. *Sensors*, **14**(12), 24408–24424.

[81] Leon, E., Clarke, G., Callaghan, V., and Doctor, F. (2010). Affect-aware Behaviour Modelling and Control Inside an Intelligent Environment. *Pervasive Mob. Comput.*, **6**(5), 559–574.

[82] Lewis, M., Haviland-Jones, J. M., and Barrett, L. F. (2008). *Handbook of emotions*. Guilford Press.

[83] Lidoris, G. and Buss, M. (2006). A Multi-Agent System Architecture for Modular Robotic Mobility Aids. In *European Robotics Symposium 2006*, number 511670, pages 15–26.

[84] Lindquist, K. A., MacCormack, J. K., and Shablack, H. (2015). The role of language in emotion: predictions from psychological constructionism. *Frontiers in psychology*, **6**, 444.

[85] Lopes, P. N., Salovey, P., Côté, S., Beers, M., and Petty, R. E. E. (2005). Emotion regulation abilities and the quality of social interaction. *Emotion*, **5**(1), 113–118.

[86] Lunenburg, J. J. M., van den Dries, S., Appeldoorn, R. P. W., Wijnands, R. W. J., and van de Molengraft, M. J. G. (2016). Tech United Eindhoven @Home 2016 Team Description Paper. Technical report, Eindhoven University of Technology.

[87] Lungu, V. (2013). Newtonian emotion system. In *Intelligent Distributed Computing VI*, pages 307–315. Springer.

[88] Maaoui, C. and Pruski, a. (2010). Emotion recognition through physiological signals for human-machine communication. *Cutting Edge Robotics*, pages 317–333.

[89] Mafik, V. and Pechoucek, M. (2004). *Social knowledge in multi-agent systems*, volume 2086/2001 of *Lecture Notes in Computer Science*, pages 211–245. Springer Berlin / Heidelberg.

[90] Mago, V. K. and Devi, M. S. (2007). A multi-agent medical system for Indian rural infant and child care. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 1396–1401.

[91] Mangina, E., Carbo, J., and Molina, J. M. (2009). *Agent-based ubiquitous computing*. Atlantis Press : World Scientific, Amsterdam; Paris.

[92] Masthoff, J. (2011). Group recommender systems: Combining individual models. In *Recommender systems handbook*, pages 677–702. Springer.

[93] Mateevitsi, V., Haggadone, B., Leigh, J., Kunzer, B., and Kenyon, R. V. (2013). Sensing the environment through SpiderSense. In *Proceedings of the 4th Augmented Human International Conference*, pages 51–57. ACM.

[94] McCarthy, J. F. and Anagnost, T. D. (1998). MusicFX: An Arbiter of Group Preferences for Computer Supported Collaborative Workouts. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, CSCW '98, pages 363–372, New York, NY, USA. ACM.

[95] McCrae, R. R. and John, O. P. (1992). An introduction to the five-factor model and its applications. *Journal of personality*, **60**(2), 175–215.

[96] McDonald, N. M., Messinger, D. S., Acerbi, A., Lombo, J. A., and Sanguineti, J. J. (2011). The development of empathy: How , when, and why. *Moral Behavior and Free Will: A Neurobiological and Philosophical Aprroach*, pages 341–368.

[97] Meftah, I. T., Thanh, N. L., and Ben Amar, C. (2010). Towards an algebraic modeling of emotional states. In *2010 Fifth International Conference on Internet and Web Applications and Services (ICIW)*, pages 513–518.

[98] Mehrabian, A. (1980). *Basic Dimensions for a General Psychological Theory : Implications for Personality , Social , Environmental , and Developmental Studies* . Oelgeschlager, Gunn & Hain.

[99] Mehrabian, a. (1997a). Analysis of affiliation-related traits in terms of the PAD Temperament Model. *The Journal of psychology*, **131**(1), 101–117.

[100] Mehrabian, A. (1997b). Comparison of the PAD and PANAS as models for describing emotions and for differentiating anxiety from depression. *Journal of psychopathology and behavioral assessment*, **19**(4), 331–357.

[101] Melchers, M. C., Li, M., Haas, B. W., Reuter, M., Bischoff, L., and Montag, C. (2016). Similar Personality Patterns Are Associated with Empathy in Four Different Countries. *Frontiers in Psychology*, **7**, 290.

[102] Michel, P. and El Kaliouby, R. (2003). Real Time Facial Expression Recognition in Video Using Support Vector Machines. In *Proceedings of the 5th International Conference on Multimodal Interfaces*, ICMI '03, pages 258–264, New York, NY, USA. ACM.

[103] Munera, E., Poza-Lujan, J.-L., Posadas-Yagüe, J.-L., Simó-Ten, J.-E., and Noguera, J. F. B. (2015a). Dynamic Reconfiguration of a RGBD Sensor Based on QoS and QoC Requirements in Distributed Systems. *Sensors*, **15**(8), 18080–18101.

[104] Munera, E., Alcobendas, M. M., Poza-Lujan, J.-L., Yagüe, J. L. P., Simo-Ten, J., and Noguera, J. F. B. (2015b). Smart Resource Integration for Robot Navigation on a Control Kerned Middleware Based System. *International Journal of Imaging and Robotics*, **15**(4), 117–129.

[105] Nayak, D., Mahapatra, A., and Mishra, P. (2013). A survey on rainfall prediction using artificial neural network. *International Journal of Computer ...*, **72**(16), 32–40.

[106] Niewiadomska-Szynkiewicz, E. and Sikora, A. (2015). Progress in Automation, Robotics and Measuring Techniques. *Advances in Intelligent Systems and Computing*, **351**, 181–190.

[107] Nogueira, P. A., Rodrigues, R., Oliveira, E., and Nacke, L. E. (2013). A hybrid approach at emotional state detection: Merging theoretical models of emotion with data-driven statistical classifiers. In *2013 IEEE WIC ACM International WI and IAT*, volume 2, pages 253–260.

[108] OANCEA, B., CIUCU, T. C., and Ciucu, S. C. (2003). Time series forecasting

using neural networks. *Proceedings of the CKS 2013 International Conference*, pages 1402–1408.

[109] Omicini, A., Ricci, A., and Viroli, M. (2008). Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, **17**(3), 432–456.

[110] Orkin, J. and Roy, D. (2011). Semi-automated dialogue act classification for situated social agents in games. In *Agents for games and simulations II*, pages 148–162. Springer.

[111] Ortony, A., Clore, G., and Collins, A. (1988). *Cognitive Structure of Emotions*. Cambridge University Press.

[112] Ortony, A., Colby, B. N., Ortony, A., Clore, G. L., and Collins, A. (1990). *The Cognitive Structure of Emotions*, volume 18. Cambridge University Press.

[113] Pan, M.-S., Yeh, L.-W., Chen, Y.-A., Lin, Y.-H., and Tseng, Y.-C. (2008). A WSN-based intelligent light control system considering user activities and profiles. *Sensors Journal, IEEE*, **8**(10), 1710–1721.

[114] Papoutsidakis, M., Piromalis, D., Neri, F., and Camilleri, M. (2014). Intelligent Algorithms Based on Data Processing for Modular Robotic Vehicles Control. *WSEAS Transactions on Systems*, **13**, 242–251.

[115] Park, M., Chitta, S., Teichman, a., and Yim, M. (2008). Automatic Configuration Recognition Methods in Modular Robots. *The International Journal of Robotics Research*, **27**(3-4), 403–421.

[116] Pennebaker, J. W. (2011). *The Secret Life of Pronouns: What Our Words Say about Us*. BLOOMSBURY.

[117] Pham, T. D. and Tran, D. (2012). Emotion recognition using the emotiv epoc device. In T. Huang, Z. Zeng, C. Li, and C. S. Leung, editors, *Neural Information*

Processing, number 7667 in Lecture Notes in Computer Science, pages 394–399. Springer Berlin Heidelberg.

[118] Piana, S., Odone, F., Verri, A., and Camurri, A. (2014). Real-time Automatic Emotion Recognition from Body Gestures. *arXiv preprint arXiv:1402.5047*, pages 1–7.

[119] Plutchik, R. (1980). A general psychoevolutionary theory of emotion. *Theories of emotion*, **1**, 3–31.

[120] Potiron, K., El Fallah Seghrouchni, A., and Taillibert, P. (2013). *From Fault Classification to Fault Tolerance for Multi-Agent Systems*. SpringerBriefs in Computer Science. Springer.

[121] Pripfl, J., Kortner, T., Batko-Klein, D., Hebesberger, D., Weninger, M., Gisinger, C., Frennert, S., Eftring, H., Antona, M., Adami, I., Weiss, A., Bajones, M., and Vincze, M. (2016). Results of a real world trial with a mobile social service robot for older adults. In *2016 11th ACM / IEEE International Conference on Human-Robot Interaction (HRI)*. Institute of Electrical and Electronics Engineers (IEEE).

[122] Raad, D. and Boele (2000). *The Big Five Personality Factors: The psycholexical approach to personality.* Hogrefe & Huber Publishers.

[123] Ranathunga, S., Cranefield, S., and Purvis, M. K. (2011). Interfacing a Cognitive Agent Platform with Second Life. In *Agents for Educational Games and Simulations - International Workshop, AEGS 2011, Taipei, Taiwan, May 2, 2011. Revised Papers*, volume 7471, pages 1–21.

[124] Rea, C., MacDonald, P., and Carnes, G. (2010). Listening to classical, pop, and metal music: An investigation of mood. *Emporia State Research Studies*, **46**(1), 1–3.

[125] Rehman, S. U. and Nadeem, A. (2011). AgentSpeak (L) Based Testing of Autonomous Agents Software Engineering, Business Continuity, and Education.

In *Software Engineering, Business Continuity, and Education*, volume 257, pages 11–20. Springer.

[126] Ricci, A., Viroli, M., and Omicini, A. (2007a). A General Purpose Programming Model & Technology for Developing Working Environments in MAS. In *5th International Workshop "Programming Multi-Agent Systems" (PROMAS 2007)*, number July 2015, pages 54–69.

[127] Ricci, A., Viroli, M., and Omicini, A. (2007b). "Give Agents their Artifacts": The A&A Approach for Engineering Working Environments in MAS. In *6th International Joint Conference "Autonomous Agents & Multi-Agent Systems" (AAMAS 2007)*, volume 5, pages 601–603.

[128] Ricci, A., Viroli, M., and Omicini, A. (2008). The A&A programming model and technology for developing agent environments in MAS. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4908 LNAI, pages 89–106. Springer.

[129] Richardson, L., Amundsen, M., and Ruby, S. (2013). *RESTful Web APIs*. " O'Reilly Media, Inc.".

[130] Richert, W. and Coelho, L. P. (2013). *Building Machine Learning Systems with Python* . Packt Publishing, Birmingham.

[131] Rincon, J., Costa, A., Novais, P., Julian, V., and Carrascosa, C. (2016a). A dynamic emotional model for agent societies. In *Advances in Practical Applications of Scalable Multi-agent Systems. The PAAMS Collection*, pages 169–182. Springer.

[132] Rincon, J. A., Garcia, E., Julian, V., and Carrascosa, C. (2014a). D eveloping A daptive A gents S ituated in I ntelligent V irtual E nvironments. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 98–109. Springer.

[133] Rincon, J. A., Carrascosa, C., and Garcia, E. (2014b). D eveloping I ntelligent V irtual E nvironments using MAM 5 M eta- M odel. In *International Conference*

on Practical Applications of Agents and Multi-Agent Systems, pages 379–382. Springer.

[134] Rincon, J. A., Carrascosa, C., and Garcia, E. (2014c). Developing Intelligent Virtual Environments Using MAM 5 Meta-Model. In *Advances in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection*, pages 379–382. Springer.

[135] Rincon, J. A., Julian, V., and Carrascosa, C. (2015a). A n E motional-based H ybrid A pplication for H uman- A gent S ocieties. In *10th International Conference on Soft Computing Models in Industrial and Environmental Applications*, volume 368, pages 203–214.

[136] Rincon, J. A., Julian, V., and Carrascosa, C. (2015b). A pplying a S ocial E motional M odel in H uman- A gent S ocieties. In *Workshop WIHAS'15. Intelligent Human-Agent Societies.*, volume 524 of *CCIS*, pages 377–388.

[137] Rincon, J. A., Julian, V., and Carrascosa, C. (2015c). S ocial E motional M odel. In *13th International Conference on Practical Applications of Agents and Multi-Agent Systems*, volume 9086 of *LNAI*, pages 199–210.

[138] Rincon, J. A., Julian, V., and Carrascosa, C. (2015d). Social Emotional Model . In *Advances in Practical Applications of Agents , Multi - Agent Systems , and Sustainability : The PAAMS Collection*, pages 199–210. Springer.

[139] Rincon, J. A., Poza, J. L., Julian, V., Posadas, J. L., and Carrascosa, C. (2016b). Extending mam5 meta-model and jacalive framework to integrate smart devices from real environments. *Plos ONE*, (10.1371/jo), 1–27.

[140] Ritchings, T., Anastassakis, G., and Panayiotopoulos, T. (2001). Multi-agent Systems as Intelligent Virtual Environments. In F. Baader, G. Brewka, and T. Eiter, editors, *KI 2001: Advances in Artificial Intelligence*, number 2174 in Lecture Notes in Computer Science, pages 381–395. Springer.

[141] Roncancio, C., Jaime, G., and Zalama, E. (2010). Modeling Virtual Agent Behavior in a Computer Game to Be Used in a Real Environment. In *Trends in Practical Applications of Agents and Multiagent Systems*, pages 623–630. Springer.

[142] Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, **39**, 1161–1178.

[143] Salzmann, C. and Gillet, D. (2013). Smart device paradigm standardization for online labs. In *4th IEEE Global Engineering Education Conference (EDUCON)*.

[144] Sanchez-Anguix, V., Julian, V., Botti, V., and García-Fornes, A. (2013). S tudying the I mpact of N egotiation E nvironments on N egotiation T eams' P erformance. *Information Sciences*, **219**, 17–40.

[145] Sanchez-Anguix, V., Aydogan, R., Julian, V., and Jonker, C. (2014). Unanimously acceptable agreements for negotiation teams in unpredictable domains. *Electronic Commerce Research and Applications*, **13**(4), 243–265.

[146] Sanchis, A., Julian, V., Corchado, J. M., Billhardt, H., and Carrascosa, C. (2014). U sing N atural I nterfaces for H uman- A gent I mmersion. In *WIHAS'14 - Intelligent Human-Agent Societies*, pages 1–10. Springer.

[147] Satyanarayanan, M. (2002). A catalyst for mobile and ubiquitous computing. *Pervasive Computing, IEEE*, **1**(1), 2–5.

[148] Saunier, J. and Jones, H. (2014). Mixed Agent / Social Dynamics for Emotion Computation. In *International conference on Autonomous agents and mult-agent systems*, pages 645–652. International Foundation for Autonomous Agents and Multiagent Systems.

[149] Saunier, J., Carrascosa, C., Galland, S., and Kanmeugne, P. S. (2015). Agent Bodies: An Interface Between Agent and Environment. In *Agent Environments for Multi-Agent Systems IV*, pages 25–40. Springer.

[150] Scherer, K. R. and Zentner, M. R. (2001). Emotional Effects of Music : Production Rules. *Music and emotion: Theory and research*, pages 361–392.

[151] Schmickl, T. (2011). How to Engineer Robotic Organisms and Swarms? In *Bio-Inspired Self-Organizing Robotic Systems*, number 216342, pages 25–52. Springer.

[152] Schmidt, A. and Van Laerhoven, K. (2001). How to build smart appliances? *Personal Communications, IEEE*, **8**(4), 66–71.

[153] Schuller, B., Rigoll, G., and Lang, M. (2003). Hidden Markov model-based speech emotion recognition. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 2, pages II—-1. IEEE.

[154] Scott, T., Green, W. B., and Stuart, A. (2005). Interactive effects of low-pass filtering and masking noise on word recognition. *Journal of the American Academy of Audiology*, **114**(11), 867–878.

[155] Senechal, T., Rapp, V., and Prevost, L. (2011). Facial feature tracking for emotional dynamic analysis. *Advances Concepts for Intelligent Vision ...*, pages 495–506.

[156] Stuckler, J., Schwarz, M., and Behnke, S. (2016). Mobile Manipulation, Tool Use, and Intuitive Interaction for Cognitive Service Robot Cosero. *Frontiers in Robotics and AI*, **3**, 58.

[157] Surakul, K., Sodsee, S., and Smanchat, S. (2015). A Control of Multiple Drones for Automatic Collision Avoidance. *Information Technology Journal*, **11**(1).

[158] Thalmann, D., Musse, S. R., and Braun, A. (2007). *Crowd simulation*, volume 1. Springer.

[159] Thompson, W. F. and Quinto, L. (2011). Music and emotion: Psychological considerations. *The Aesthetic Mind: Philosophy and Psychology*, pages 357–375.

[160] Tian, J. and Tianfield, H. (2003). A Multi-agent Approach to the Design of an E-medicine System. In *Multiagent System Technologies*, pages 85–94. Springer.

[161] Tivatansakul, S. and Ohkura, M. (2013). Healthcare system design focusing on emotional aspects using augmented reality #x2014; relaxed service design. In *Computational Intelligence in Healthcare and e-health (CICARE), 2013 IEEE Symposium on*, pages 88–93.

[162] Tsonos, D., Stavropoulou, P., Kouroupetroglou, G., Deligiorgi, D., and Papatheodorou, N. (2014). Emotional prosodic model evaluation for greek expressive text-to-speech synthesis. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **8514 LNCS**(PART 2), 166–174.

[163] Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, **10**(5), 293–302.

[164] Unluturk, M. S., Oguz, K., and Atay, C. (2009). Emotion Recognition Using Neural Networks. In *Proceedings of the 10th WSEAS International Conference on Neural Networks*, NN'09, pages 82–85, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society (WSEAS).

[165] Upton, E. and Halfacree, G. (2014). *Raspberry Pi user guide*. John Wiley & Sons.

[166] van der Zwaag, M. D., Westerink, J. H. D. M., and van den Broek, E. L. (2011). Emotional and psychophysiological responses to tempo, mode, and percussiveness. *Musicae Scientiae*, **15**(2), 250–269.

[167] Van Oijen, J., Van Doesburg, W., and Dignum, F. (2011). Goal-based communication using bdi agents as virtual humans in training: An ontology driven dialogue system. In *Agents for games and simulations II*, pages 38–52. Springer.

[168] Villarrubia, G., De Paz, J. F., Bajo, J., and Corchado, J. M. (2014). Ambient Agents: Embedded Agents for Remote Control and Monitoring Using the PANGEA Platform. *Sensors*, **14**(8), 13955.

[169] Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, **57**(2), 137–154.

[170] Visutsak, P. (2012). Emotion Classification Using Adaptive SVMs. **1**(3), 279–282.

[171] Vukadinovic, D. and Pantic, M. (2005). Fully automatic facial feature point detection using Gabor feature based boosted classifiers. In *Systems, Man and Cybernetics , 2005 IEEE International Conference on*, volume 2, pages 1692–1698. IEEE.

[172] Whitman, B. and Smaragdis, P. (2002). Combining Musical and Cultural Features for Intelligent Style Detection. In *Ismir*, pages 5–10, Paris, France.

[173] Yoder, M. A. and Kridner, J. (2015). *BeagleBone cookbook*. O'Reilly Media.

[174] Yu, C.-H. and Nagpal, R. (2011). A self-adaptive framework for modular robots in a dynamic environment: theory and applications. *The International Journal of Robotics Research*, **30**(8), 1015–1036.

[175] Yu, C.-h., Yu, C.-h., Nagpal, R., and Nagpal, R. (2008a). Distributed consensus and self-adapting modular robots. In *IROS-2008 workshop on Self-Reconfigurable Robots and Applications*, pages 1–4.

[176] Yu, C.-h., Yu, C.-h., Nagpal, R., and Nagpal, R. (2008b). Sensing-based shape formation on modular multi-robot systems: A theoretical study. In *Applied Sciences*, number Aamas, pages 12–16.

[177] Zambonelli, F., Jennings, N. R., and Wooldridge, M. (2003). Developing multiagent systems: The gaia methodology. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, **12**(3), 317–370.

[178] Zykov, V., Williams, P., Lassabe, N., and Lipson, H. (2008). Molecubes Extended : Diversifying Capabilities of Open-Source Modular Robotics. *Iros 2008*, page 12.