ALMA MATER STUDIORUM – UNIVERSITY OF BOLOGNA
SCHOOL OF SCIENCE

Bachelor degree in
INFORMATION SCIENCE FOR MANAGEMENT (8014)

ENRICHING COMPANION ROBOTS WITH ENHANCED REMINISCENCE ABILITIES

Bachelor degree's dissertation in
SOFTWARE ENGINEERING

RAPPORTEUR:
Prof. Paolo Ciancarini

PRESENTED BY:
Bruno Marafini

CO-RAPPORTEUR:
Dr. Andrea Giovanni Nuzzolese & Dr. Alessandro Russo

SESSION 17° July 2018
ACADEMIC YEAR 2018/2019

ENRICHING COMPANION ROBOTS WITH ENHANCED REMINISCENCE ABILITIES

AMS Alma Mater Studiorum, Bologna

ABSTRACT

MARIO is a robot, part of a robotics company called KOMPAÏ Robotics that deals with the production and management of robots who take care of elderly people who suffer from dementia or who still need and aid. I'm going to deal and develop with some new aspects in this robot.

Bruno Marafini

Experimental Thesis in Computer Science

Reminiscence Project – Enriching companion robots with enhanced reminiscence abilities

Bruno Marafini, student of Computer Science

University of Bologna

Author Note

Prepared for: Professor Paolo Ciancarini, Software Engineering

Doctor Andrea Giovanni Nuzzolese, researcher at the STLab in Roma

Alessandro Russo, researcher at CNR, Rome

"Gli algoritmi condizionano le nostre scelte. Analogico e digitale non sono più distinti.

L'unica possibilità è lasciarsi contaminare."

Table of Contents

List of figures

List of Tables

List of Abbreviations

CNR	National Centre of Research
CGA	Comprehensive Geriatric Assessment
MON	Mario Ontology Network
RBS	Rule Based Network
PDA	Personal Data Assistance
ES	Expert Systems
KBS	Knowledge Base Systems
CBS	Case Based Systems
FL	Fuzzy Logic Systems
ANNS	Artificial Neural Networks
GA	Genetic Algorithms
ICT	Information and Communication Technologies
RT	Reminiscence Therapy
LR	Life Reviews
PGC	Philadelphia Geriatric Centre
RFS	Reminiscence Functions Scale
FP	Facebook Pensieve
CIRCA	Computer Interactive Reminiscence and Conversation Aid

UKCRC	UK Computing Research Committee

Acknowledgement

It's a pleasure to thank the many people who have helped contribute to the completion of this work, predominantly my supervisors, Professor Andrea Giovanni Nuzzolese and Alessandro Russo, both researchers and developers of some of the main aspects of project Mario at CNR Rome, and Professor Paolo Ciancarini together with the coordinator of the degree, Professor Gianluigi Zavattaro.

Their supervision has been invaluable and has made this thesis possible. My gratitude is also extended to the staff of the International Centre of Research (CNR) in Rome and of course to my family that was the one to perform many tests on the functions I developed.

I should like to extend particular thanks to my first supervisor, A.G. Nuzzolese for his wisdom, encouragement, guidance, and humour throughout the years of this Computer Science Degree (3 years), without which, this work would undoubtedly not be what it is today.

I would also like to thank my colleagues at the University of Bologna, AMS, Alma Mater Studiorum for their friendship, assistance and support, over the years, especially my close circle of friends - you know who you are! Thanks also to my dear friends outside the university, your interest in my work and your kind words of encouragement will never be forgotten and of course to my girlfriend for her continued love and support.

Finally, as I said before, I would like to thank my family for their unyielding love, support and confidence in me, especially my grandparents apart from my mum and dad.

For Luigi, Barbara, Armando, Lina, Renata

And Gianni (even if you're not here with us today)

Abstract

In this document I will go on discussing a project conceived by Professor Andrea Giovanni Nuzzolese and Alessandro Russo, both researchers and developers of some of the main aspects of project Mario at CNR Rome.

MARIO is a robot, part of a robotics company called KOMPAÏ Robotics that deals with the production and management of Robots who take care of elderly people who suffer from dementia or who still need an aid; more generally speaking, there is talk of weak and lonely people within an organization and / or institutions (nursing homes ...) or in their own homes.

There are numerous characteristics of MARIO, which ultimately contribute to all those which are the manufacturing objectives of KOMPAÏ Robotics, namely:

1. **Safety** - Falling is a constant risk that can have serious consequences. KOMPAÏ provides audits, alerts from key individuals and can also request assistance in an emergency.
2. **Health monitoring** - KOMPAÏ can manage medical parameters and connect with health professionals. KOMPAÏ collects, collects and stores all data from medical devices in the cloud.
3. **Make every day better** - Sometimes it can be difficult to make sense of time and place. KOMPAÏ helps to recall useful information and offers interactive entertainment.
4. **Social connectivity** - When we feel lonely, it is a great comfort to go beyond hearing a voice and seeing your loved one! KOMPAÏ allows you to feel close to families and friends, talking to them like them.
5. **Independence** - When seated comfortably, getting up can be complicated and someone is not always there to help you. With KOMPAÏ, just call the robot, get up and go!

6. **Made and carefully conceived** - Our team designed KOMPAÏ using patient and care focus groups. The goal was to bring both cognitive and physical support, in a friendly, empathetic, and reassuring way, even if without the intention of replacing the only presence of human help.

My project, or rather my contribution to MARIO, is to look for a specific method which let the robot show a specific set of photos to the user according to the expressions, feelings and emotions, the user will reveal.

Example: the robot randomly chooses a marriage photo and the user suddenly start to laugh and to express positive feelings with positive words; the robot will try to understand if it's a good photo for the user or not, and in the first case will continue to show the same kind of pictures while in the second case, will change completely set of photos to be shown.

The pleasure of the subject expressed in relation to a photo must be subject to an index of interest between predefined and specified values that may be to show a certain interest in a picture or the subjects within the image or the situation that surrounds them.

Keywords: CNR, project, Mario

Abstract (italiano)

In questo documento discuterò un progetto concepito e curato principalmente dal Professor Andrea Giovanni Nuzzolese insieme ad altri collaboratori del CNR di Roma, Centro di Ricerca Nazionale, con anche Alessandro Russo, entrambi ricercatori e sviluppatori di alcune delle applicazioni e funzioni fondamentali di un progetto avente come oggetto un robot: project MARIO.

MARIO è un robot, parte di una società di robotica chiamata KOMPAÏ robotica che si occupa della produzione e gestione di robot che si prendono cura di persone anziane che soffrono di demenza o chi ancora bisogno di un aiuto; più in generale, si parla di persone deboli e solitarie all'interno di un'organizzazione e / o istituzioni (case di cura...) o nelle loro case.

Numerose ovviamente sono le caratteristiche di questo ROBOT:

1. **Sicurezza** - Cadere è un rischio costante che può avere conseguenze gravi. KOMPAÏ fornisce audit, avvisi da parte di persone chiave e può anche richiedere assistenza in caso di emergenza.
2. **Monitoraggio della Salute** – KOMPAÏ può gestire i parametri medici e connettersi con i professionisti della salute. KOMPAÏ raccoglie, raccoglie e archivia tutti i dati dai dispositivi medi nel cloud.
3. **Rendere ogni giorno migliore** – A volte può essere difficile dare un senso di tempo e luogo. KOMPAÏ aiuta a recuperare informazioni utili e offre intrattenimento interattivo.
4. **Connettività sociale** – Quando ci sentiamo soli, è un grande conforto andare oltre il sentire una voce e vedere la persona amata! KOMPAÏ ti permette di sentirti vicino a famiglie ed amici, parlando con loro come loro.

5. **Indipendenza** – Quando si è seduti comodamente, alzarsi può essere complicato e non c'è sempre qualcuno per aiutarti. Con KOMPAÏ, basta chiamare il robot, alzarsi ed andare!
6. **Creato e concepito con cura** – Il team ha progettò KOMPAÏ utilizzando gruppi di focus con pazienza e cura. L'obiettivo era portare il supporto sia cognitivo che fisico, in un accogliente, empatico e modo rassicurante, anche se senza l'intenzione di sostituire l'unica presenza di aiuto umano.

Il mio progetto, o piuttosto il mio contributo a MARIO, è alla ricerca di un metodo specifico che permetta il robot di mostrare un insieme specifico di foto all'utente secondo le espressioni, i sentimenti e le emozioni, si rivelerà l'utente.

Esempio: il robot sceglie casualmente una foto del matrimonio e l'utente improvvisamente inizia a ridere e ad esprimere sentimenti positivi con parole positive; il robot cercherà di capire se è una buona foto per l'utente o non, e nel primo caso continuerà a mostrare lo stesso tipo di immagini, mentre nel secondo caso, cambia completamente set di foto da mostrare.

Il piacere del soggetto espresso in relazione a una foto deve essere soggetto a un indice di interesse tra i valori predefiniti e specificati che può essere a mostrare un certo interesse in un'immagine o soggetti entro l'immagine o la situazione che li circonda.

Parole_Chiave: CNR, project, Mario

Declaration

I hereby declare that with effect from the date on which this thesis is deposited in the Library of the University of Bologna, Alma Mater Studiorum, I permit the Librarian of the University to allow the thesis to be copied in whole or in part without reference to me on the understanding that such authority applies to the provision of single copies made for study purposes or for inclusion within stock of another library.

IT IS A CONDITION OF USE OF THIS THESIS THAT ANYONE WHO CONSULTS IT MUST RECOGNISE THAT THE COPYRITH RESTS WITH THE AUTHOR AND THAT NO QUOTATION FROM THE THESIS AND NO INFORMATION DERIVED FROM IT MAY BE PUBLISHED UNLESS THE SOURCE IS PROPERLY ACKNOWLEDGED

Bruno Marafini

Chapter 1

Introduction

1.1 Background

MARIO addresses the difficult challenges of loneliness, isolation and dementia in older persons through innovative and multi-faceted inventions delivered by service robots. The effects of these conditions are sever and life-limiting,

They burden individuals and societal support systems. Human intervention is costly simulation mediated by robots.

The proposal is coordinated by the nursing unit of NUI, involves managers of the City of Stockport developing social care and community support programs to older people, and also involves an important hospital in Italy pushing research boundaries in comprehensive geriatric assessment. Pilot activities and interaction with caregivers, older persons and stakeholders (within a standards based medical assessment interaction with caregivers, older persons and stakeholders (within a standards based medical assessment methodology) is present from beginning to finish and provides a user led design ethos.

The proposal brings together the well-known Kompai platform and commercial footprint of ROBOSOFT, control expertise of RURobots, advances from the computer laboratory at CNR, and an innovative robot application development platform by Ortellio. Dedicated telecommunication (CNET) and exploitation experts (R2M) widen technical and market outreach potential.

From this unique combination, clear advances are made in the use of semantic data analytics, personal interaction, and unique applications tailored to better connect older persons to their care providers, community own social circle and also to their personal interests.

Each objective is developed with a focus on loneliness, isolation and dementia.

The impact centres on deep progress toward EU scientific and market leadership in service robots and a user driven solution for this major societal challenge.

The competitive advantage is the ability to treat tough challenges appropriately. The path to market deployment by bringing Mario solutions to end users is clearly developed.

1.2 Objective

MARIO is an assistive robot that has to support a set of knowledge-intensive tasks aimed at (i) helping patients affected by dementia to feel more autonomous and less lonely, (ii) supporting carers in their activity to assess the patient's cognitive condition.

Examples of knowledge-intensive tasks are the Comprehensive Geriatric Assessment (CGA) and the triggering of proper entertainment activities. MARIO has also to address a number of behavioural tasks such as to drive the patient to a specific location (e.g. the bathroom) and identifying searched objects (e.g. keys).

In order to enable this tasks MARIO features a set of abilities implemented by pluggable software components. MARIO abilities, when executed, contribute to and benefit from a common knowledge base.

For example, MARIO ability to accompany a patient to the bathroom retrieves the information needed about the physical environment from MARIO knowledge base and stores the information that such an event happened at a certain date/time in the same knowledge base.

Another example is the ability to entertain the patient by playing music, which retrieves music tracks from the knowledge base and stores liking feedback about them in order to reuse such knowledge in future executions. As for the CGA the associated ability retrieves questions to be

posed to the patient from the knowledge base and stores the obtained answers and all associated relevant metadata.

1.3 Aims of the Research

I'm going to explain what the problem was I had to solve based on all the information I had to analyse.

Mainly what I had to do was, creating a new method inside the class *Reminiscence.java* in order to display photos to the patient in an order established by the grade of like for the selected picture and not randomly.

Inside the JSON file, *photoPatterns.json* there are some casual *preconditions* with the relative *question, answer type* and *the expected answer*.

Before this modification what the project did, was:

- **Randomly** display, if selected, the first photo to the user
- Based on some indexes and on what the user was saying as response, the system randomly tries to understand (based on some JSON statements) if the user likes or not the photo
- Either the user liked or not the photo, the system used to randomly select the next photo to display
- Of course when I say randomly, I intend that the system goes inside an array of the 'uploaded pictures' and based on some **patterns**, that I'm going to explain in a bit, displays the next picture.

1.4 Thesis outline

Chapter 2 discusses *Reminiscence and Ontologies on Companion Robots*, and so the significance of *Reminiscence* generally speaking and the utilization of these in application to companion robots – in addition to this it discusses the importance of some *Intelligent Techniques (KBS)* and the *benefits of Reminiscence for older Adults.* Then it introduces the concept of *ontologies* always related with companion robots, specifying *Mario's Ontology Network.*

Chapter 3 talks about *Requirements Analysis and System Design;* I've decided to talk about all the necessary *user requirements analysis and some requirements specifications*, with an overview of the *architecture of MARIO, where I've modified the code, a component diagram in UML, MARIO software, a class diagram, a sequence diagram and a pseudo-code of my algorithm.*

Chapter 4 discusses a general *System Implementation and Testing of my algorithm* with a general *usage scenario* and a simple link from which you can get the class I've modified on *GitHub.*

In the final part, Chapter 5, you can find my conclusions and some of my ideas about some future work on the same project (there are still a lot of apps that could be implemented in Mario).

Chapter 2

Reminiscence and Ontologies on Companion Robots

2.1 Introduction

In care for older adults, behavioural and psychological symptoms of dementia (BPSD) like agitation and aggression are distressing for patients and their caretakers, often resulting in premature institutionalization with increased costs of care.

To improve mood and mitigate symptoms, as a non-pharmaceutical approach, emotion-oriented therapy like reminiscence work is adopted in face-to-face communication.

Telecommunication support is expected to be provided by robotic media as a bridge for digital divide for those with dementia and facilitate social interaction both verbally and nonverbally. The purpose of this case is to explore the conditions in which robotic media can effectively attract attention from older adults with dementia and promote their well-being.

Reminiscence intervention as a communication aid for older adults has shown effect through emotional responses and conversation engagement by triggering long-term memory even for people suffering from dementia (1).

The importance of conversation among people with dementia, care staffs, family members and volunteers can be seen in studies like the **CIRCA** computer reminiscence program and the networked interaction therapy based on Web technology (2).

IP videophones and video-sharing tools have been adapted to networked sessions in reminiscence therapy where interviews with caretakers showed that there was a reduction in verbal abuse (3).

As it becomes a big issue in dementia care to improve various BPSD such as agitation, depression and aggression, telecommunication media has the potential to improve symptoms of dementia and the user's relationship with care-takers and aid other social interactions.

2.2 Reminiscence

Reminiscence is the process of using episodic memory to mentally transport oneself back in time to relive an event or experience from the past (4), and is widely accepted as a regular enjoyable activity of older people (5), and an accepted therapeutic intervention (6) (7) (8). Bluck and Levine (1998) (9) define reminiscence as:

"Reminiscence is the volitional or non-volitional act or process of recollecting memories of one's self in the past. It may involve the recall of particular or generic episodes that may or may not have been previously forgotten, and that are accompanied by the sense that the remembered episodes are veridical accounts of the original experiences. This recollection from autobiographical memory may be private or shared with others".

There are various types of reminiscence which can be broadly categorized as oral and silent (10). Silent reminiscence naturally involves one person; however, oral reminiscence can take the form of 'shared reminiscence', where people review a past event or experience where all were present, or 'storytelling reminiscence' where one person relays a past event or experience to others who were not present when it occurred (11). Studies have shown that the use of memory prompts can be visual, such as photographs; audio such as music; tactile, such as physical memorabilia objects, and olfactory in that different scents can be used.

2.2.1 Benefits of Reminiscence for Older Adults

Reminiscence plays an important role in the lives of older adults and affords them immense pleasure. Many perfect the art of storytelling and enjoy its social benefits. The telling of stories of past events and experiences defines family identities and is an integral part of many cultures. Episodic memory exercises such as reminiscing and storytelling have been proven to give therapeutic benefits to older people. Studies show that episodic memory is often the first type of memory to become impaired in older adults. This type of memory dysfunction inhibits reminiscence, which has been proven to be of particular benefit to older people. Losing the ability to recollect past memories is not only disadvantageous, but can prove quite detrimental, especially to many older adults.

Benefits of reminiscence also include improved self-esteem and life satisfaction. Krakovsky (12) states that naturally nostalgic people have high self-esteem and are less prone to depression. Recalling past events and experiences can aid older adults as they work through current situations and problems.

Looking back over ‘the good old days’ also helps older adults maintain current and past relationships and often is of great benefit in accepting the past and coming to terms with earlier life events.

Researchers recognize the healthcare benefits to older people from the therapeutic stimulus offered by reminiscing activities and various reminiscence healthcare initiatives exist:

Reminiscence Therapy (RT)

Reminiscence Therapy (RT) is the structured use of reminiscence as a medical intervention by professionals to support and counsel older people. RT has been found to improve self-esteem and life satisfaction among other people. RT involves reflection and discussion of past life experiences and events, which is often themed and involves the use of memory prompts. RT affords comfort and security by acknowledging and respecting an older person's life history. Family care-givers also use RT and it is a highly rated psychosocial intervention in dementia care.

RT has the potential to help individuals with dementia as well as family caregivers and to improve staff care. Therapeutic purposes of RT for people with dementia include reducing social isolation, offering an enjoyable and stimulating activity, promoting self-worth, and providing a way to sustain relationships with loved ones.

Although reminiscence often involves pleasant memories to promote enjoyment, it can also involve serious or sad memories for therapeutic or cathartic purposes. RT also offers benefits for family caregivers, including new ways of interacting with a relative. Additionally, engaging in RT with residents with dementia gives an opportunity for staff to better understand the resident's current behaviours and personality.

Several reviews have been published on randomized control trials of RT interventions, both for the general population of older adults and for those with dementia.

These reviews shown problems with available evidence that include differing RT forms, limited number of studies, and small samples.

Despite these problems, RT appears to be a promising intervention for individuals with dementia, with outcomes including improvements in mood, cognition, and behaviour.

Life Review (LR)

The term ‘Life Review’ was first coined by Butler ((13) suggested that as people approach the end of their lives it is beneficial to look bac and take stock of accomplishments and unresolved issues as a form of ‘psychological preparation for death’.

Therefore, reminiscing generates the material needed for LR. LR is the chronological review of a person’s past life experiences. Participants are encouraged to reminisce and evaluate experiences and often a ‘life story book’ is produced, which in turn is passed on to children.

LR has been proven to enhance relationships, often between an aged parent and adult child, and also has been found to be an effective method of enhancing life-satisfaction, self-worth, and the psychological well-being of older adults. Haight and Haight (14) developed Life review and Experiencing Forms (LREF) to measure life satisfaction and psychological well-being.

Place Based Reminiscence

Place based reminiscence is based on the premise that human experiences occur within the context of a certain place. Research conducted by Chaudhury (2003) found that by focussing on personally meaningful past places they can act as ‘mnemonic anchors’ in autobiographical reminiscence.

(15) Chaudhury conducted place-based research with the cognitively intact and impaired residents of nursing homes.

Results showed that place-based reminiscence is an effective method of eliciting rich narratives of older people’s past experiences and proved both enjoyable and therapeutic for the older participants.

Measuring Reminiscence

Measurement tools have been developed to access older people and investigate the functions, purpose, effects and benefits of reminiscence, and many reminiscence studies have been conducted which use these tools to measure the benefits of reminiscence for older people.

Morris spearheaded the development of tools to assess older people, that are widely used by clinicians and researchers.

These include the Philadelphia Geriatric Centre (PGC) Morale Scale and the Observed Emotion Rating Scale. Webster devised and validated a Reminiscence Functions Scale (RFS), a 43-item questionnaire to assess the functions of reminiscence over a life course.

Cappeliez et al. (2007) defined a taxonomy of reminiscence functions as presented in Table 5. This table presents 8 different reminiscence functions, the nature of that reminiscence, and parallel self-representation characteristics. Cappeliez et al. (2007) categorise these 8 functions into 3 groups: **self-focus (**coherence and meaning); **guidance** (knowledge-based); and **social bonding** (emotion management).

- Self-focus (includes integrative, death preparation, obsessive and escapism) where reminiscence ins intrapersonal and remains private
- Guidance (includes instrumental and transmissive) where reminiscence involves the recall and sharing of a personal knowledge or experience to assist with current problems.
- Social bonding (includes narrative and intimacy) where reminiscence is performed for the maintenance of relationships.

Benefits reported for reminiscence include improvements in mood, perceptions of life, satisfaction, happiness, well-being, self-worth and self-esteem Lin et al. (2003).

2.3 Using Technology to Support Reminiscence

Technology has been used to support reminiscence using various methods, for example, in the development of electronic methods. (16) Mulvenna believe there are three main modes for the use of reminiscence tools: use by an individual; shared use, perhaps with a relative or carer in the same physical space; and lastly, shared use where individuals are remotely distant from one another using the internet. A variety of digital aids have been developed, which are deployed across different platforms: online, PC based or mobile.

Since reminiscence is recall of past events, the reminiscence tools discussed here are referred to as retrospective memory aids. However, not all were developed specifically for older adults.

2.4 Digitalised Memories and Life-Logging

Arguably, the first mention of digital storage of extensive quantities of personal data and media was by Vannevar Bush in 1945 with the introduction of his **original concept of the first personal digital assistant, the Memex**. Bush (2001) envisioned the Memex as a digital device that would act as an extension of one's memory.

An easily accessible supplement, which would store personal data, records, communications and even reading books.

This **'life-logging'** vision inspired many researchers and is regarded as a corner stone upon which much work has been conducted since .

2.5 Intelligent Techniques

Intelligent techniques prove effective tools in software development as they provide developers with the means to create 'adaptive' and 'aware' applications. There is a variety of intelligent techniques: Expert Systems (ES): including Knowledge-Based Systems (KBS), Rule-Based Systems (RBS) and Case-Based Systems (CBS); Fuzzy Logic-based systems (FL); Artificial Neural Networks (ANNs); and Genetic Algorithm-based systems (GA).

Intelligent techniques can be used independently or combined together to produce hybrid system. The choice of intelligent technique in software development is based on the suitability of the technique to the current situation.

Knowledge-Based Systems

It's going to be analysed only the main case we are going to talk about: **KBS.**

Knowledge is the information about a specific domain needed by a computer program to enable it to exhibit intelligent behaviour with regard to a specific problem. Knowledge includes information about both real-world entities and the relationships between them.

It can also take the form of procedures for combining and operating on information – Computer programs that encapsulate such knowledge are called *Knowledge-based systems.*

A knowledge-based systems (KBS) is a computer system which generates and uses knowledge from different sources, data and information. These systems aid in solving problems, especially complex ones, by using artificial intelligence concepts.

These systems are mostly used in problem-solving procedures and to support human learning, decision making and actions.

2.6 Nowadays: Technology use among Seniors

Nowadays although seniors consistently have lower rates of technology adoption than the general public, this group, is more digitally connected than ever. In fact, some groups of seniors – such as those who are younger, more affluent and more highly educated – report owning and using various technologies at rates similar to adults under the age of 65.

Still, there remains a notable digital divide between younger and older Americans. And many seniors who are older, less affluent or with lower levels of educational attainment continue to have a distant relationship with digital technology.

Four-in-ten seniors now own smartphones, more than double the share that did so in 2013.

As we can see (Figure 2 - Pew Research Centert) **roughly four-in-ten seniors are smartphone owners**; for more statistics about what technology has brought to old people nowadays have a look to *Pew Research Centre* *Pew Research Centre - Website*.

Research conducted by Hanson (2010) discusses a wave in the increase In digital products and services and highlights how those who do not use current technologies, particularly older people, could easily become excluded. Hanson et al. (2010) state that this is largely due to two factors: a disinterest in such technologies by older people, and the fact that many technologies are simply not designed and built to accommodate the strengths and weaknesses of older people. He concludes that these are problems that will not simply 'go away' with the next generation of older adults, since the problems associated with age will still be clear, as will the increase in newer technologies which must be 'learnt', and those unable to engage with such technologies will quickly become excluded.

He recommends that further work be conducted which focuses on the strengths rather than the weaknesses of older adults, and remarks that if the needs of today's older adults cannot be met, then evidently, the needs of tomorrow's older adults certainly will not be met.

2.7 Robot Revolution: why technology for older people must be designed with care and respect

Many countries around the world have ageing populations as I already said before and a growing prevalence of dementia. Japan, in particular, (even Italy but with a lower percentage) is a '*super-ageing'* society, with a population getting older faster than anywhere else in the world due to **long life expectancy and low birth rates**.

In 2015, an article in (17) pointed out that 'Japan will be at the forefront of devising ways to tackle the social, economic and medical challenges posed by a super-ageing society'.

A high-tech innovator, (18), the country is producing robots for people with dementia – to provide companionship, improve safety in the home, and help with therapy.

Other countries are jumping on board (19) with initiatives to incorporate service robots into dementia care.

But we must make sure that people, especially those living with dementia, are firmly at the centre of research and development.

Technology, after all, (20) should be for and by people, not something imposed on them.

2.7.1 Robots for Dementia Care

Robotic devices can help with physical caregiving tasks, monitor behaviour and symptoms, and provide cognitive support.

They can be classified into 7 major categories: powder-assisted robots that transfer patients, from beds and wheelchairs; assistive robots for personal mobility; toiletry assistance robots; bathing assistance robots monitoring robots with sensor system; social interaction robots (21); and therapeutic robots (22).

Robots in the first four categories could be widely used in aged care to assist elders with physical mobility limitations.

Those in the latter categories could be widely used in aged care to assist elders with physical mobility limitations.

Those in the latter categories have particular application for people with dementia who experience who experience difficulty with memory, thinking and communication, as well as changes in mood, behaviour and personality.

2.7.2 Social Robots

Robots designed for social and therapeutic purposes can look like cute animals – such as PARO, a baby seal robot – or like small humanoids – such as SATO, ROMEO or MARIO.

For people living with memory loss robots can remind them about things they often forget, such as prompting them to take medication and eat meals, pointing out the location of household items and helping with their use. Robots can also provide companionship and entertainment, such as engaging people in games, dancing and singing.

Robots can support people with dementia to live independently and help reduce negative behavioural and psychological symptoms.

They can also support human caregivers by providing watchful eyes and helping hands. Robots do not experience stress and burnout and there are other practical benefits, too. Robots that look like cuddly animals can be used in place of real animals for pet therapy. A robotic cat, for instance, doesn't need food, water or a litter box and won't scratch if it is squeezed a bit too hard.

2.8 Ontologies

Ontology is the philosophical study of the nature of being, becoming, existence, or reality, as well as the basic categories of being and their relations. Traditionally listed as a part of the major branch of philosophy known as metaphysics, ontology often deals with questions concerning what entities exist or may be said to exist and how such entities may be grouped, related within a hierarchy, and subdivided according to similarities and differences. A very simple definition of ontology is that it is the examination of what is meant, in context, by the word 'thing'. (23).

An ontology is a conceptualisation of a domain into a human-understandable, but machine-readable format consisting of entities, attributes, relationships, and axioms (24) (25).

Ontologies can provide a rich conceptualisation of the working domain of an organisation, representing the main concepts and relationships of the work activities. These relationships could represent isolated information such as an employee’s home phone number, or they could represent an activity such as authoring a document, or attending a conference.

We use the term ontology to refer to the classification structure and instances within a knowledge base.

2.9 MON – MARIO Ontology Network

2.9.1 MON Knowledge areas

The MON is composed of different ontologies that cover the following knowledge areas relevant to MARIO in order to make it a cognitive agent able to support older patient affected by dementia:

- Personal sphere (i.e. people information, information about relationship among people etc.);
- Life events and patterns (i.e. information about everyday events, memories etc.);
- Social and multimedia content (i.e. online social network community etc.);
- Environment (i.e. information about: room, furniture etc.);
- Health sphere (i.e. information about living patterns, health patterns, vital signs etc.)
- Emotional sphere (i.e. information about emotions, sentiments etc.);
- Open knowledge (i.e. speech-derived data etc.);
- Regulatory sphere (i.e. information about: norms, rules etc.);
- MARIOcpetion (i.e. information about MARIO abilities etc.).

2.9.2 Ontology design methodology

The MON is designed by following best design practices and pattern-based ontology engineering aimed at extensively re-using **Ontology Design Patterns (ODPs)** for modelling ontologies.

ODPs are modelling solutions that can be re-used in order to solve recurrent ontology design problems. Hence, they enable design by-reuse methodologies.

For example, ODPs can be reused by means of their specialisation or composition according to their types and to the scope of the new ontology that is going to be modelled.

The design methodology I followed is the same as the one that my Mentors followed and it's based on the **eXtreme Design** (XD is an agile design methodology developed in the context of the NeON project. XD is inspired by the eXtreme Programming (XP) in fact like XP, it emphasises incremental development and recommends pair development, test driven development, refactoring, and a divide-and-conquer approach to problem-solving).

2.9.3 MON Modules

In the (Figure 6 - Top Level MON) you can see the top level of the MON. the top level is identified by the **OWL ontologies** connected to the MARIO ontology (*i.e. Mario.owl)* by green arrows. The arrows, independently from their colours, represent *owl:imports* axioms.

Each ontology module connected to the top level MARIO ontology identifies a knowledge are as introduced before and is itself a broader conceptualisation built on top of more specific ontology modules.

Such modules are, in turn, built by means of re-use of ontology design patterns according to out ontology design methodology. In next paragraphs the ontology modules of the MON (*i.e. framester, cga, affordance and tagging)* are going to be analysed.

Other ontology modues are part of our ongoing work and will be developed and released by next pilots.

These modules will re-use the following ontologies:

- The **computer based Patient Record (CPR) Ontology,** which provides a modelling solution for describing patient released records;
- **DogOnt**, which provides a suitable solution for describing domotic devices, the home environment and how architectural elements and furniture are placed inside the home;
- the **Emotion Ontology (MFOEM) Emotion Ontology** for representing phenomena such as emotions and moods;
- the **Middle Layer Ontology for Clinical Care (MLOCC)** that guarantees the exploitation of well-founded and formalised medical concepts;
- the **RECommendations Ontology (RECO)** that provides domain-independent means to describe user profiles and preferences in a coherent and context-aware way;
- the **Symptom Ontology (SYMP)** that provides means to describe symptoms associated with diseases.

Framester

Many resources belonging to different domains are now being published on-line using **Linked data** principles to provide easy access to structured data on web. This includes many linguistic resources that are already a part of **Linked Data,** but made available mainly for the purpose of being used by NLP applications.

When dealing with robot understanding, we need to integrate knowledge about the robot's physical context, linguistic knowledge, and knowledge about the world in general.

Designing a knowledge base for an assistive robot can be therefore considered a direct application of the Linked Data paradigm, which provides interoperability across existing Linked Open Data (world's background knowledge), linguistic knowledge, user's knowledge and robot's sensor knowledge.

Framester is intended to work as a knowledge graph/linked data hub to connect lexical resources, NLP results, linked data and ontologies. It is bootstrapped from an existing resources, notably the RDF versions of **FrameNet, WordNet, VerbNet and BabelNet** by interpreting their semantics as a subset of Fillmore's frame semantics and semiotics.

State of the Art

Two of the most important linguistic linked open data resources are **WordNet** and **FrameNet.** they have already been formalised as semantic web resources, *e.g. in OntoWordNet, WordNetRDF, FrameNet RDF, etc.*

For example FrameNet allows to represent textual resources in terms of Frame Semantics. The usefulness of FrameNet is limited by its limited coverage and non-standard semantics.

An evident solution would be to establish valid links between FrameNet and other lexical resources such as WordNet, VerbNet and BabelNet to create wide – coverage and multi – lingual extensions of FrameNet. by overcoming these limitations NLP-based applications such as question answering, machine reading and understanding, etc. would eventually be improved.

Within MARIO these are important requirements, hence it has been developed **Framester:** a frame-based ontological resource acting as a hu between e.g. FrameNet, WordNet, VerbNet, BabelNet, DBpedia, Yago, DOLCE-Zero and leveraging this wealth of links to create an interoperable *predicate space* formalised according to frame semantics and semiotics.

Data designed according to the predicates in the predicate space created by Framester result to be more accessible and interoperable, modulo alignments between specific entities or facts.

The closest resources to Framester are **FrameBase** and **Predicate Matrix**.

FrameBase is aimed at aligning linked data to FrameNet frames, based on similar assumptions as Framester's: full-fledged formal semantics for frames, detour-based extension for frame coverage, and rule-based lenses over linked data.

However, the coverage of FrameBase is limited to an automatically learnt extension (with resulting inaccuracies) of FrameNet.WordNet mappings, and the alignment to linked data schemas is performed manually.

Anyway, Framester could be combined with FrameBase (de)reification rules so that the two projects can mutually benefit from their results.

Predicate Matrix is an alignment between predicates existing in FrameNet, VerbNet, WordNet and PropBank.

It does not assume a formal semantics, and its coverage is limited to a subset of lexical senses from those resources.

A RDF version of Predicate Matrix has been created in order to add it to the Framester linked data cloud, and (ongoing work) to check if those equivalences can be reused in semantic web applications.

Ontology Description

Framester (Figure 7 – Framester) uses the D&S (Descriptions and Situations) knowledge pattern, which allows to distinguish the rectification of the intension of a predicate (*a description*) from the reification of the extensional denotation of a predicate (*a situation*). A description *d* can define or reuse concepts $c^1,\ldots,c^n$ that can be used to *classify* entities $e^1 \ldots e^m$ involved in a situation *s* that is expected to be compatible with *d*.

For those reasons, D&S perfectly fits the core assumptions of Fillmore's frame semantics, by which a frame is a schema for conceptualising the interpretation of a natural language text (and beyond), it denotation (a frame occurrence) is a situation, and the elements (or semantic roles) of a frame are aspects of a frame, which can be either obligatory, optional, inherited, reused, etc.

Furthermore, D&S takes into account a semiotic theory to integrate linguistic and formal semantics. It can therefore support additional frame semantics assumptions such as *evocation* and *semantic typing*.

2.10 Older people and difficulties with new technologies: a state of the art

In parallel with the growth in the older population is the increase in the use of computing technology in all aspects of everyday life. Many older people, however, are not adopting and fully using such technologies.

Technology may play a problematic role in older people's life (26). In 2001, the Dutch Office of Social and Cultural Planning (27) summarized some of the threats of new technology, but also some opportunities.

Many older people use the opportunities, but the number of users is lower than in the younger age group. In 1998, mobile phones were owned by 37% in age group 35-44 and by 10% in age group 75+; a PIN-card by 97% and 75%; PC by 74% and 5% respectively. The presence of a PC is higher among men, the better educated, higher income and more-person households. Age has a major impact on PC availability. Over 50% of the older people experience problems in using a PC, a VCR, and a mobile phone. People, who are familiar with computers in their work, stay on as users after retirements. Eventually, most older people will find the way to the digital world. However, the introduction of ticket machines, automatic teller machines, etc. comes too fast for some of them (Figure 1 - Technology and Older People).

2.11 Sentiment analysis

Sentiment analysis in companion robots consists in the task of automatically determining from text the attitude, emotion, or some other affectual state of the author.

In my implementation I had to use it in order to extract opinions from a spoken text, and so to recognise the attitude (positive, negative or objective) of an opinion holder on a certain topic (photo).

Mainly, the patient or somebody else, gives an opinion on a photo (that can be **positive, neutral or negative**) and then the system based on this shows another picture related to the same pattern if *Positive or Neutral* or a picture not related at with the one shown before if *Negative.*

2.12 Development: Hypothesis and Solution

Before starting to modify the source code I had to make some hypothesis in order to develop my thesis and an outline able to solve the problem I had.

At the beginning there were a lot of hypothesis, most of them not applicable to this problem, but each analysed in detail.

- **The first hypothesis** was the one to develop and implement an algorithm able to show pictures to the patient related to the sentiment analysis expressed on each photo by the user itself too complexed for the project itself; basically modifying all the code in the **photo viewer** algorithm in the *Reminiscence,java class* with the ability to pic the degree of pleasure of patient itself and generating with another algorithm the next picture to show.

- **The second hypothesis (the main one, the one I've developed as a thesis)** consists in developing new functions in the algorithm already specified in the *Reminiscence.java class* such as the photo viewer; so, based on the degree of pleasure of the patient, expressed as **POSITIVE, NEUTRAL, NEGATIVE**, showing pictures of the same patterns if the patient expresses a comment being part of the first 2 cases or clearing the **HashMap** containing the **list of pictures** of that pattern.

After my modifications to the project, applying the solution expressed in the **second hypothesis** now the picture viewer shows photos based on the **sentiment analysis;** briefly, each photo when shown, can be commented vocally by the user and the comment is analysed with a text to speech algorithm and an index automatically assigned to each sentence.

There's specified a classification on 3 kind of pleasures: **Neutral, Positive or Negative**.

In order to establish whether it's classifiable between one of these 3, there's a *.json file* with some positive, negative and neutral sentences so that with some **speech to text or text to speech algorithm** is able to understand whether the sentence is classifiable between one of these sentiments.

Once classified, between each sentiment, is assigned a **score** between 1.0 to 4.0 so that the system is able to recognise the sentiment assigned to the picture.

For example, let's suppose that the first picture shown is a marriage photo:

- If the comment of the user is something like – 'I like this picture!' – the sentiment recognised will be **POSITIVE** and so the next picture to be shown will be a photo showing another marriage;

- If the comment of the user is something like – 'This picture is ok' – the sentiment recognised will be **NEUTRAL** and so the next picture to be shown will be a photo showing another marriage, so it will behave the same as the **POSITIVE** one;
- If the comment of the user is something like – 'I don't like this photo!' – the sentiment recognised will be **NEGATIVE** and so the next picture to be shown will be a photo showing not the same situation of the first shown. In addition to this the algorithm, if **NEGATIVE** is detected, clears the *HashMap* with *the related list of the related pattern*.

There was a project built in 1995, called *'Keeping the Older People Mobile'* (28), the conclusion was that older people made little use of new technologies such as ATM, ticket machines and telephone cards.

Age was the most important predictor of usage, followed by education and gender. In the more recent study *'Mobilate',* age appeared again to be a relevant factor followed by residential location, education, and cognitive abilities (29). The follow-up showed that the same people used more technologies 5 years later. Old age as such was not preventing the use of technology.

Chapter 3

Mario Reminiscence:

Architecture & Design

3.1 Robosoft: Mario Architecture

MARIO is a Kompaï robot which addresses the difficult challenges of loneliness, isolation and dementia in older persons through innovative and multi-faceted inventions delivered by service robots.

The effects of these conditions are severe and life-limiting. They burden individuals and societal support systems.

Human intervention is costly but the severity can be prevented and/or mitigated by simple changes in self-perception and brain stimulation mediated by robots.

The goal is to provide assistance for dependent seniors and the handicapped, and even clients with autism.

Kompaï is able to **respond to voice commands** with some speech2text or some text2speech functions, **navigate through an apartment** and **help the client with cognitive details like creating a shopping list,** or **creating an audio notepad.** A built-in webcam enables easy videoconferencing between Kompaï 'clients' and family, friends and medical personnel. A **touch screen tablet,** front and center on the unit, can be used to control the robot during more advanced tasks or when the speech recognition (like in this case) requires additional input.

It can even work from remote localhost in order to let all its developers to test it during the development of a new app for example.

3.2 The problem

In this chapter I'm going to explain every single operation I did on the source code including some screenshots which let people understand better all the operation (I/O operations and more) and let understand better the final output.

Mainly what I had to do was, creating a new method inside the class *Reminiscence.java* in order to display photos to the patient in an order established by the grade of like for the selected picture and not randomly.

Inside the JSON file, *photoPatterns.json* there are some casual *preconditions* with the relative *question*, *answer type* and *the expected answer*.

Before this modification what the project did, was:

- **Randomly** display, if selected, the first photo to the user
- Based on some indexes and on what the user was saying as response, the system randomly tries to understand (based on some JSON statements) if the user likes or not the photo
- Either the user liked or not the photo, the system used to randomly select the next photo to display
- Of course when I say randomly, I intend that the system goes inside an array of the 'uploaded pictures' and based on some **patterns**, that I'm going to explain in a bit, displays the next picture.

3.3 The algorithm

The algorithm I've implemented is contained in the *Reminscence.java* class and all the changes I've done together with all the new code are contained in some parts of code called *personal contribution*.

Normally as I already said before, what the system used to do, was showing pictures randomly based on a sequence of called methods inside a recursive method called *getNextInteractionPattern()*.

So this method used to look for, firstly, some pictures with some related patterns, then something to pic from that Map with a List inside containing only pictures of the same kind (pictures that represent the same event).

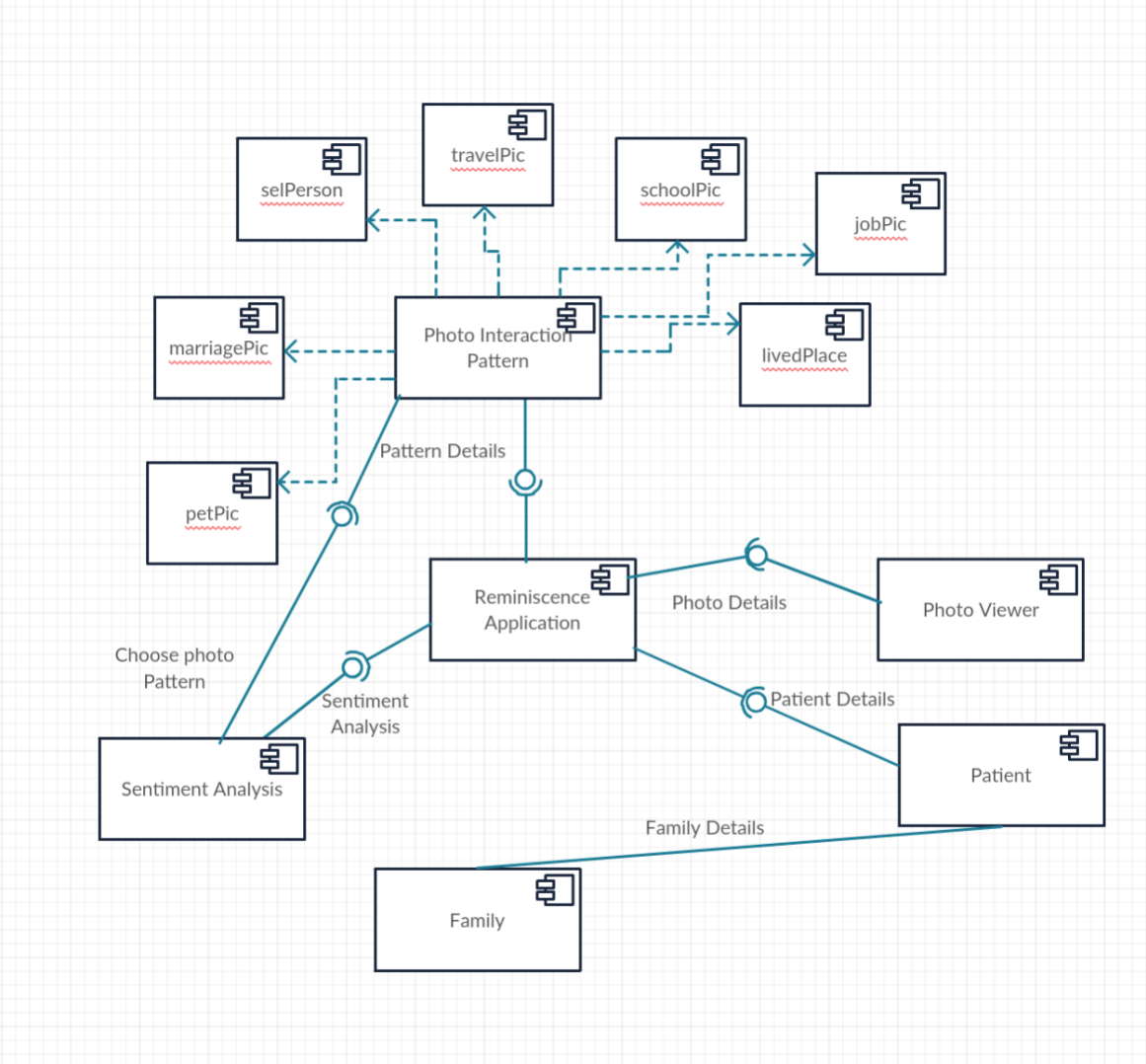
What I had to do in order to change this algorithm, was, giving as an input *SentimentScore sentiment* to the method *getNextInteractionPattern()* declaring sentiment of course out of that cycle and before itself; then inside each *if* cycle, declaring some new cases with *if … else* construct so that if the patient likes the pictures (always based on a range from 1.0 to 4.0) or he/she is neutral about it, the system keep showing phots of the same pattern, while if the patient doesn't like the picture the system is going to clear the array with all the pictures of that pattern, assuming of course that the patient doesn't like at all that set of photos, and keep executing the successive pattern so that the picture shown will be a different one.

It's a very interesting application considering that nowadays most of the apps on social robots are pretty boring; this, one based on sentiment analysis, tries to understand the patient itself, and of course do not exclude the possibility of having an update of the same software, once we will release it in some industries.

3.4 Mario Reminiscence: UML Component Diagram

In **unified modelling language (UML)** a **component diagram**, depicts how components are wired together to form larger components or **software systems**.

They are used to illustrate the structure of arbitrarily complex systems. In this case I'm going to introduce a simple UML Component Diagram of the part of the architecture I've implemented (**Error! Reference source not found.**).



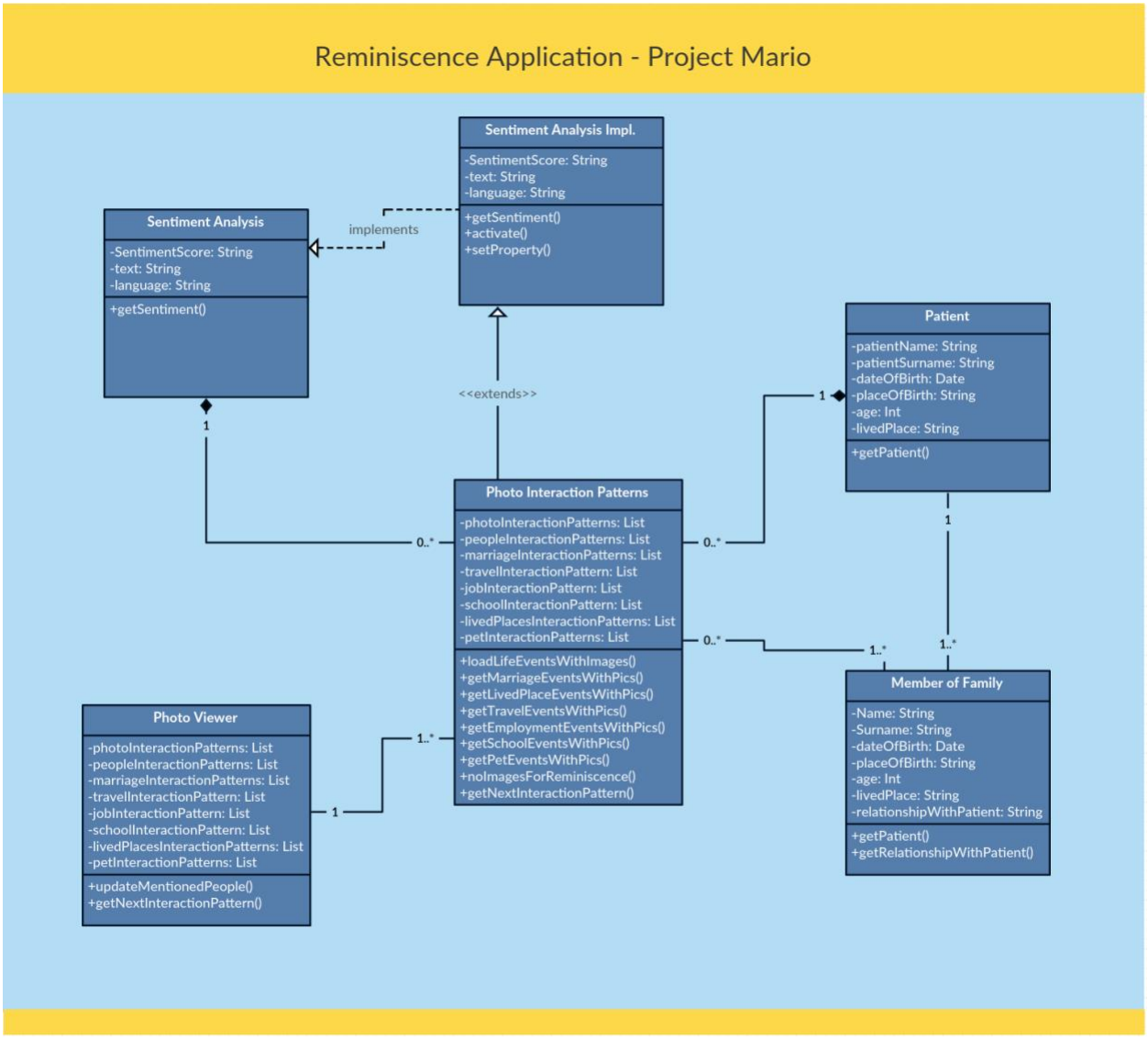
It shows all the relationships between the components that are part of my project; a UML Component Diagram of whole Project Mario would have been much bigger.

Inside my reminiscence application there are mainly **6 components:**

- **Reminiscence Application:** the name of the component which represent the main part of my project;
- **Sentiment Analysis:** it can be *POSITIVE, NEUTRAL OR NEGATIVE* and it represent the use of **natural language processing, text analysis, computational linguistics, and biometrics** to systematically identify, extract, quantify and study affective states and subjective information, so in this case to analyse the sentiment or the emotion expressed by the patient once commented the photo he/she has been shown;
- **Photo Viewer:** that's the main algorithm I had to modify, in order not to show anymore random photos to the user, but, based on sentiment analysis, to show them in a specific order, that I already explained in the **Algorithm section;**
- **Patient:** that component represent the patient whose characteristics are the one to be analysed;
- **Family:** finally his/her family that are mainly people whose tags are set during a photo upload and are necessary to let the robot recognise who is shown in a specific picture;
- **Photo Interaction Pattern:** that's one of the method I had to modify in order to let the photos be shown In another way – it contains all the patterns of the different Maps in Java containing a List of pictures separated by set.

3.5 Mario Reminiscence: UML Class Diagram

Now I'm going to show you a **class diagram**, and so a type o static structure diagram that is going to describe the structure of my system by showing all the system's classes, their attributes, operations (or methods), and the relationships among objects (**Error! Reference source not found.**).



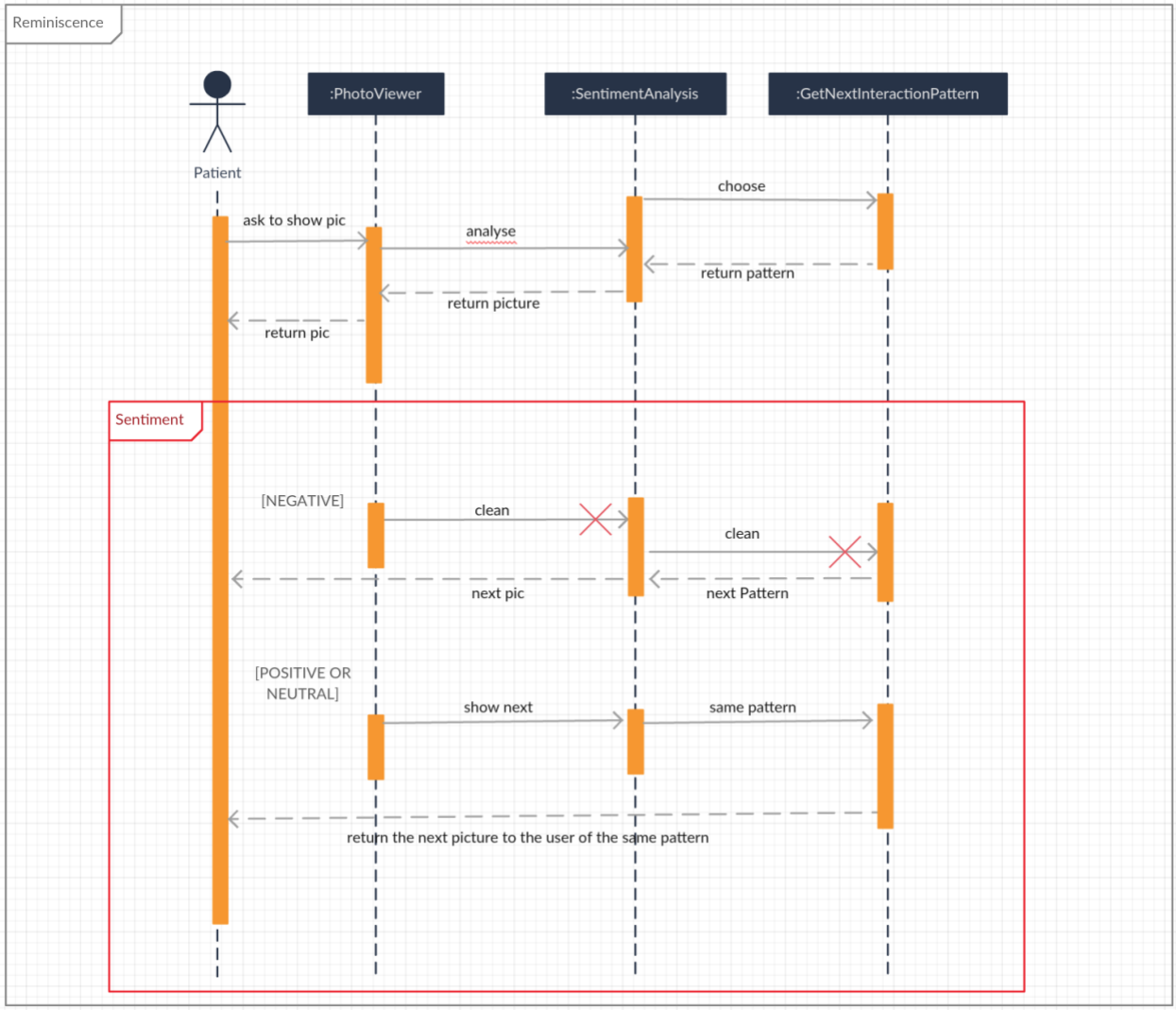
As you can see I have 6 main classes as follows:

- **PhotoInteractionPattern:** this class represent the main one I had to modify because it contains all the Maps with a list of Pattern inside, and so all the pictures uploaded by the user subdivided by patterns themselves. In addition to this, it contains all the relative methods with which is possible to show determinate pictures of one pattern, or calling them in a sequential way, showing in a recursive way randomly some pictures, obviously of different patterns. Just focus on the **main recursive method** that is *getNextInteractionPattern()*;
- **PhotoViewer:** another class in which is contained my contribution to this project – in this class is contained the algorithm I've made in order to show the photo in an ordinate way;
- **Patient:** this class represent the Patient and so, it's very useful in order to let the system understand whose photos are the one uploaded (there could be more than one patient, not active, but present) – even though just one patient can be active at a time;
- **MemberOfFamily:** easily this class contains all the information related to all the member of the family (of the patient of course), so that is possible to tag parents In the photos you upload – once a picture Is shown the robot will pronounce the name of all the people tagged in that precis picture;

- **SentimentAnalysis:** in this class there is just an enumeration, an *enum type,* so just a special data type that enables for a variable to be a set of predefined constants – in this case the constants are 3 (*Neutral, Positive, Negative*) and refers to the recognition of the sentiment during the comment of a user on a certain photo. As I already explained, depending on the recognition of one of the three, the system will do something different:
- **SentimentAnalysisImpl:** here you will find the implementation of **SentimentAnalysis.java** and so the explication of how the sentiment analysis works on this robot – plus, is explained how the robot recognise if a comment of the user is *Neutral, Positive or Negative*, - simply assigning a score, **SentimentScore**, in a range from [0 to 4] and specifying that if the score is less than 1.5 the sentiment will be negative, if the score is between 1.5 and 2.5 the sentiment will be neutral and if the score is more than 2.5 the sentiment will be positive.

3.6 Mario Reminiscence: UML Sequence Diagram

Finally I'm going to show you one of the latest engineering UML diagram that is a **Sequence Diagram** which shows object interactions arranged in time sequence; so it depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.



As you can see there are 2 different frames: the first one **Reminiscence** is simply the one which selects which photo should be shown to the user, so simply the mechanism to show a picture whereas the second one, **Sentiment**, shows the same things but with the aid of Sentiment Analysis and so as I already said, based on the Sentiment Score, whether to show a picture of the same pattern if the patient says a positive or neutral comment or show a picture of completely other pattern if the patient says a negative comment. Keep in mind that I've decided to clear the Map containing the set of photos for one particular pattern if the user says something bad or negative about the picture itself.

3.7 Algorithm

Inside this paragraph I'm going to specify the main architecture I've modified, what there was before and why (**Error! Reference source not found.**).

Inside the whole architecture of MARIO there are a lot of .java files representing all the functioning of the system, including some .json files which represent some set of sentences that the robot expects to hear from the patient with an expected answer (YES/NO).

Before my modifications the whole system used to show randomly some pictures, just following a simple order for choosing patterns, a sequential order – the whole code of *PhotoViewer* with all its patterns explained is contained in the class **Reminiscence.java**; that's why I focused my work on that file.

Each pattern has been specified as a Map containing the name of the Map and a list of images, being part of that specific pattern.

For example, let's take into consideration the **Marriage Pattern,** but of course think that each pattern has been implemented in the same way**:**

Firstly, this is how the Service and the Map are declared.

```
@Reference
    private MarriageService marriageService;
    private Map<Marriage, List<Image>> marriagesWithImages;
```

Then, what the systems does, is to load marriage interaction patterns as follows.

```
// load marriage interaction patterns
        marriageInteractionPatterns = PhotoInteractionPattern.loadMarriagePatterns();
        log.info("Ability " + ABILITY_NAME + ": " + marriageInteractionPatterns.size()
+ " Marriage Interaction Pattern(s) loaded");
```

What you have to do later is to check if there is any applicable pattern to a certain image.

```
private List<PhotoInteractionPattern> getApplicablePatterns(Image photo, ELProcessor pr
ocessor, List<PhotoInteractionPattern> patterns) {

        processor.defineBean("photo", photo);
        List<PhotoInteractionPattern> applicablePatterns = new ArrayList<>();
        //ELProcessor processor = buildELProcessor(photo);
        // for each pattern evaluate pattern precondition wrt the photo
        for (PhotoInteractionPattern pattern : patterns) {
            if ((Boolean)processor.getValue(pattern.getPrecondition(), Boolean.class))
{
                applicablePatterns.add(pattern);
            }
        }
        return applicablePatterns;
    }
```

You need to load all the events for that life event, in this case, a marriage; but of course keep in mind that you have to do it for all the patterns you declare..

```
private void loadLifeEventsWithImages() {
        // marriages
        marriagesWithImages = marriageService.getMarriageEventsWithPics();
        System.err.println("MARRIAGES WITH IMAGES: " + marriagesWithImages.size());

                // jobs
               ...
}
```

Then you need a method that permits you to iterate over life events (in this case, marriage) and over the list of images. Then it tries to find an image for that specific event with an applicable pattern so that after the image will be shown, it will be deleted (after use).

```
1.  private Map<Image, List<PhotoInteractionPattern>> getMarriagePicWithPatterns(ELProcesso
    r processor) {
2.          List<PhotoInteractionPattern> applicablePatterns;
3.
4.          // iterate over marriage events
5.          Iterator<Marriage> marriageIterator = marriagesWithImages.keySet().iterator();

6.          while (marriageIterator.hasNext()) {
7.              Marriage marriageWithPics = marriageIterator.next();
8.              // set marriage as bean in EL processor
9.              processor.defineBean("marriageEvent", marriageWithPics);
10.             // get list of pics for marriage
11.             List<Image> marriagePics = marriagesWithImages.get(marriageWithPics);
12.             // iterate over the list of images
13.             Iterator<Image> imagesIterator = marriagePics.iterator();
14.             while (imagesIterator.hasNext()) {
15.                 Image image = imagesIterator.next();
16.                 applicablePatterns = getApplicablePatterns(image, processor, marriageIn
    teractionPatterns);
17.                 if (!applicablePatterns.isEmpty()) {
18.                     // found an image for the event with applicable patters: return the
     image with the patterns
19.                     Map<Image, List<PhotoInteractionPattern>> result = new HashMap<>();

20.                     result.put(image, applicablePatterns);
21.                     // remove the image as it will be used
22.                     imagesIterator.remove();
23.                     return result;
24.                 }
25.                 else {
26.                     // no patterns for image, remove the image
27.                     imagesIterator.remove();
28.                 }
29.             }
30.             // if here, there are no images with applicable pattern or all images were
    used, so remove the event
31.             marriageIterator.remove();
32.         }
33.         // if here, there are no events with images having patterns or all events were
    used
34.         return Collections.emptyMap();
35.
36.     }
```

Finally there is the main method to get and show a picture of that pattern, of course this method will be inserted inside a recursive method so that it will be called more than once.

```
1.  private Map<Image, List<PhotoInteractionPattern>> getNextInteractionPattern(SentimentSc
    ore sentiment) {
2.
3.          // all images where used...reset dialogue context, reload data and retry recurs
    ively!
4.
5.          Map<Image, List<PhotoInteractionPattern>> imageWithInteractionPattern;
6.
7.          /*
8.           * Try to show pictures based on pattern and on sentiment analysis
9.           *
10.          * @author Bruno Marafini
11.          * @theme Personal Contribution
12.          */
13.
14.         // try with marriage pattern
15.         imageWithInteractionPattern = getMarriagePicWithPatterns(dialogueContext.proces
    sor);
16.         //sentiment = sentimentAnalyzer.getSentiment(utterance, currentLocale.getLangua
    ge());
17.         if (!imageWithInteractionPattern.isEmpty() && sentiment == null) {
18.             updateMentionedPeople(imageWithInteractionPattern.keySet().iterator().next(
    ));
19.             dialogueContext.totalNumQuestions++;
20.             dialogueContext.prompedAboutPerson = false;
21.             return imageWithInteractionPattern;
22.         } else if (!imageWithInteractionPattern.isEmpty() && sentiment == SentimentScor
    e.POSITIVE || sentiment == SentimentScore.NEUTRAL){
23.             updateMentionedPeople(imageWithInteractionPattern.keySet().iterator().next(
    ));
24.             dialogueContext.totalNumQuestions++;
25.             dialogueContext.prompedAboutPerson = false;
26.             return imageWithInteractionPattern;
27.         } else if (!imageWithInteractionPattern.isEmpty() && sentiment == SentimentScor
    e.NEGATIVE){
28.             marriagesWithImages.clear();
29.             sentiment = null;
30.         }
31. ...
32. ...
33. }
```

All these examples needed to let you understand how the system used to work before my modifications.

Now I'll simply show, inside the *Reminiscence.java class*, what I've modified:

```
1.  private Map<Image, List<PhotoInteractionPattern>> getNextInteractionPattern(SentimentSc
    ore sentiment) {
2.
3.          // all images where used...reset dialogue context, reload data and retry recurs
    ively!
4.
5.          Map<Image, List<PhotoInteractionPattern>> imageWithInteractionPattern;
6.
7.          /*
8.           * Try to show pictures based on pattern and on sentiment analysis
9.           *
10.          * @author Bruno Marafini
11.          * @theme Personal Contribution
12.          */
13.
14.         // try with marriage pattern
15.         imageWithInteractionPattern = getMarriagePicWithPatterns(dialogueContext.proces
    sor);
16.         //sentiment = sentimentAnalyzer.getSentiment(utterance, currentLocale.getLangua
    ge());
17.         if (!imageWithInteractionPattern.isEmpty() && sentiment == null) {
18.             updateMentionedPeople(imageWithInteractionPattern.keySet().iterator().next(
    ));
19.             dialogueContext.totalNumQuestions++;
20.             dialogueContext.prompedAboutPerson = false;
21.             return imageWithInteractionPattern;
22.         } else if (!imageWithInteractionPattern.isEmpty() && sentiment == SentimentScor
    e.POSITIVE || sentiment == SentimentScore.NEUTRAL){
23.             updateMentionedPeople(imageWithInteractionPattern.keySet().iterator().next(
    ));
24.             dialogueContext.totalNumQuestions++;
25.             dialogueContext.prompedAboutPerson = false;
26.             return imageWithInteractionPattern;
27.         } else if (!imageWithInteractionPattern.isEmpty() && sentiment == SentimentScor
    e.NEGATIVE){
28.             marriagesWithImages.clear();
29.             sentiment = null;
30.         }
31.
32.         // try with mentioned person
33.         imageWithInteractionPattern = selectPersonWithProfilePic(dialogueContext.proces
    sor, dialogueContext.mentionedPeople);
34.         if (!imageWithInteractionPattern.isEmpty() && !dialogueContext.prompedAboutPers
    on && sentiment == null) {
35.             String personIntroSentence = MessageFormat.format(getRandomString(introPers
    on), DefaultFunctions.getPersonNameWithRelationship((Person)(dialogueContext.processor.
    eval("person")),patient,currentLocale.getLanguage()));
36.             tts.speakAndWait(personIntroSentence, false, getAbilityTopic());
37.             dialogueContext.prompedAboutPerson = true;
38.             return imageWithInteractionPattern;
39.         } else if (!imageWithInteractionPattern.isEmpty() && sentiment == SentimentScor
    e.POSITIVE || sentiment == SentimentScore.NEUTRAL) {
40.             String personIntroSentence = MessageFormat.format(getRandomString(introPers
    on), DefaultFunctions.getPersonNameWithRelationship((Person)(dialogueContext.processor.
    eval("person")),patient,currentLocale.getLanguage()));
```

```
41.             tts.speakAndWait(personIntroSentence, false, getAbilityTopic());
42.             dialogueContext.prompedAboutPerson = true;
43.             return imageWithInteractionPattern;
44.         } else if (!imageWithInteractionPattern.isEmpty() && sentiment == SentimentScor
    e.NEGATIVE) {
45.             peopleWithProfilePics.clear();
46.             sentiment = null;
47.         }
48.
49.         // try with lived place pattern
50.         imageWithInteractionPattern = getLivedPlacePicWithPatterns(dialogueContext.proc
    essor);
51.         if (!imageWithInteractionPattern.isEmpty() && sentiment == null) {
52.             updateMentionedPeople(imageWithInteractionPattern.keySet().iterator().next(
    ));
53.             dialogueContext.totalNumQuestions++;
54.             dialogueContext.prompedAboutPerson = false;
55.             return imageWithInteractionPattern;
56.         } else if (!imageWithInteractionPattern.isEmpty() && sentiment == SentimentScor
    e.POSITIVE || sentiment == SentimentScore.NEUTRAL){
57.             updateMentionedPeople(imageWithInteractionPattern.keySet().iterator().next(
    ));
58.             dialogueContext.totalNumQuestions++;
59.             dialogueContext.prompedAboutPerson = false;
60.             return imageWithInteractionPattern;
61.         } else if (!imageWithInteractionPattern.isEmpty() && sentiment == SentimentScor
    e.NEGATIVE){
62.             livedPlacesWithImages.clear();
63.             sentiment = null;
64.         }
65.
66.         imageWithInteractionPattern = selectPersonWithProfilePic(dialogueContext.proces
    sor, dialogueContext.mentionedPeople);
67.         if (!imageWithInteractionPattern.isEmpty() && !dialogueContext.prompedAboutPers
    on && sentiment == null ) {
68.             String personIntroSentence = MessageFormat.format(getRandomString(introPers
    on), DefaultFunctions.getPersonNameWithRelationship((Person)(dialogueContext.processor.
    eval("person")),patient,currentLocale.getLanguage()));
69.             tts.speakAndWait(personIntroSentence, false, getAbilityTopic());
70.             dialogueContext.prompedAboutPerson = true;
71.             return imageWithInteractionPattern;
72.         } else if (!imageWithInteractionPattern.isEmpty() && (sentiment == SentimentSco
    re.POSITIVE || sentiment == SentimentScore.NEUTRAL)){
73.             updateMentionedPeople(imageWithInteractionPattern.keySet().iterator().next(
    ));
74.             dialogueContext.totalNumQuestions++;
75.             dialogueContext.prompedAboutPerson = false;
76.             return imageWithInteractionPattern;
77.         } else if (!imageWithInteractionPattern.isEmpty() && sentiment == SentimentScor
    e.NEGATIVE){
78.             peopleWithProfilePics.clear();
79.             sentiment = null;
80.         }
81.
82.         // try with travel place pattern
83.         imageWithInteractionPattern = getTravelPicWithPatterns(dialogueContext.processo
    r);
84.         if (!imageWithInteractionPattern.isEmpty() && sentiment == null) {
```

```
85.             updateMentionedPeople(imageWithInteractionPattern.keySet().iterator().next(
    ));
86.             dialogueContext.totalNumQuestions++;
87.             dialogueContext.prompedAboutPerson = false;
88.             return imageWithInteractionPattern;
89.         } else if (!imageWithInteractionPattern.isEmpty() && sentiment == SentimentScor
    e.POSITIVE || sentiment == SentimentScore.NEUTRAL){
90.             updateMentionedPeople(imageWithInteractionPattern.keySet().iterator().next(
    ));
91.             dialogueContext.totalNumQuestions++;
92.             dialogueContext.prompedAboutPerson = false;
93.             return imageWithInteractionPattern;
94.         } else if (!imageWithInteractionPattern.isEmpty() && sentiment == SentimentScor
    e.NEGATIVE){
95.             travelsWithImages.clear();
96.             sentiment = null;
97.         }
98.
99.         imageWithInteractionPattern = selectPersonWithProfilePic(dialogueContext.proces
    sor, dialogueContext.mentionedPeople);
100.                if (!imageWithInteractionPattern.isEmpty() && !dialogueContext.prompedAb
    outPerson && sentiment == null ) {
101.                    String personIntroSentence = MessageFormat.format(getRandomString(in
    troPerson), DefaultFunctions.getPersonNameWithRelationship((Person)(dialogueContext.pro
    cessor.eval("person")),patient,currentLocale.getLanguage()));
102.                    tts.speakAndWait(personIntroSentence, false, getAbilityTopic());
103.                    dialogueContext.prompedAboutPerson = true;
104.                    return imageWithInteractionPattern;
105.                } else if (!imageWithInteractionPattern.isEmpty() && (sentiment == Senti
    mentScore.POSITIVE || sentiment == SentimentScore.NEUTRAL)){
106.                    updateMentionedPeople(imageWithInteractionPattern.keySet().iterator(
    ).next());
107.                    dialogueContext.totalNumQuestions++;
108.                    dialogueContext.prompedAboutPerson = false;
109.                    return imageWithInteractionPattern;
110.                } else if (!imageWithInteractionPattern.isEmpty() && sentiment == Sentim
    entScore.NEGATIVE){
111.                    peopleWithProfilePics.clear();
112.                    sentiment = null;
113.                }
114.
115.                // try with school pattern
116.                imageWithInteractionPattern = getSchoolPicWithPatterns(dialogueContext.p
    rocessor);
117.                if (!imageWithInteractionPattern.isEmpty() && sentiment == null) {
118.                    updateMentionedPeople(imageWithInteractionPattern.keySet().iterator(
    ).next());
119.                    dialogueContext.totalNumQuestions++;
120.                    dialogueContext.prompedAboutPerson = false;
121.                    return imageWithInteractionPattern;
122.                } else if (!imageWithInteractionPattern.isEmpty() && (sentiment == Senti
    mentScore.POSITIVE || sentiment == SentimentScore.NEUTRAL)){
123.                    updateMentionedPeople(imageWithInteractionPattern.keySet().iterator(
    ).next());
124.                    dialogueContext.totalNumQuestions++;
125.                    dialogueContext.prompedAboutPerson = false;
126.                    return imageWithInteractionPattern;
127.                } else if (!imageWithInteractionPattern.isEmpty() && sentiment == Sentim
    entScore.NEGATIVE){
```

```
128.                     schoolsWithImages.clear();
129.                     sentiment = null;
130.                 }
131.
132.                 imageWithInteractionPattern = selectPersonWithProfilePic(dialogueContext
    .processor, dialogueContext.mentionedPeople);
133.                 if (!imageWithInteractionPattern.isEmpty() && !dialogueContext.prompedAb
    outPerson && sentiment == null) {
134.                     String personIntroSentence = MessageFormat.format(getRandomString(in
    troPerson), DefaultFunctions.getPersonNameWithRelationship((Person)(dialogueContext.pro
    cessor.eval("person")),patient,currentLocale.getLanguage()));
135.                     tts.speakAndWait(personIntroSentence, false, getAbilityTopic());
136.                     dialogueContext.prompedAboutPerson = true;
137.                     return imageWithInteractionPattern;
138.                 } else if (!imageWithInteractionPattern.isEmpty() && (sentiment == Senti
    mentScore.POSITIVE || sentiment == SentimentScore.NEUTRAL)){
139.                     updateMentionedPeople(imageWithInteractionPattern.keySet().iterator(
    ).next());
140.                     dialogueContext.totalNumQuestions++;
141.                     dialogueContext.prompedAboutPerson = false;
142.                     return imageWithInteractionPattern;
143.                 } else if (!imageWithInteractionPattern.isEmpty() && sentiment == Sentim
    entScore.NEGATIVE){
144.                     peopleWithProfilePics.clear();
145.                     sentiment = null;
146.                 }
147.
148.                 // try with pet pattern
149.                 imageWithInteractionPattern = getPetPicWithPatterns(dialogueContext.proc
    essor);
150.                 if (!imageWithInteractionPattern.isEmpty() && sentiment == null) {
151.                     updateMentionedPeople(imageWithInteractionPattern.keySet().iterator(
    ).next());
152.                     dialogueContext.totalNumQuestions++;
153.                     dialogueContext.prompedAboutPerson = false;
154.                     return imageWithInteractionPattern;
155.                 } else if (!imageWithInteractionPattern.isEmpty() && sentiment == Sentim
    entScore.POSITIVE || sentiment == SentimentScore.NEUTRAL){
156.                     updateMentionedPeople(imageWithInteractionPattern.keySet().iterator(
    ).next());
157.                     dialogueContext.totalNumQuestions++;
158.                     dialogueContext.prompedAboutPerson = false;
159.                     return imageWithInteractionPattern;
160.                 } else if (!imageWithInteractionPattern.isEmpty() && sentiment == Sentim
    entScore.NEGATIVE){
161.                     petsWithImages.clear();
162.                     sentiment = null;
163.                 }
164.
165.                 imageWithInteractionPattern = selectPersonWithProfilePic(dialogueContext
    .processor, dialogueContext.mentionedPeople);
166.                 if (!imageWithInteractionPattern.isEmpty() && !dialogueContext.prompedAb
    outPerson) {
167.                     String personIntroSentence = MessageFormat.format(getRandomString(in
    troPerson), DefaultFunctions.getPersonNameWithRelationship((Person)(dialogueContext.pro
    cessor.eval("person")),patient,currentLocale.getLanguage()));
168.                     tts.speakAndWait(personIntroSentence, false, getAbilityTopic());
169.                     dialogueContext.prompedAboutPerson = true;
170.                     return imageWithInteractionPattern;
```

```
171.                } else if (!imageWithInteractionPattern.isEmpty() && (sentiment == Senti
    mentScore.POSITIVE || sentiment == SentimentScore.NEUTRAL)){
172.                    updateMentionedPeople(imageWithInteractionPattern.keySet().iterator(
    ).next());
173.                    dialogueContext.totalNumQuestions++;
174.                    dialogueContext.prompedAboutPerson = false;
175.                    return imageWithInteractionPattern;
176.                } else if (!imageWithInteractionPattern.isEmpty() && sentiment == Sentim
    entScore.NEGATIVE){
177.                    peopleWithProfilePics.clear();
178.                    sentiment = null;
179.                }
180.
181.                // try with job pattern
182.                imageWithInteractionPattern = getJobPicWithPatterns(dialogueContext.proc
    essor);
183.                if (!imageWithInteractionPattern.isEmpty()) {
184.                    updateMentionedPeople(imageWithInteractionPattern.keySet().iterator(
    ).next());
185.                    dialogueContext.totalNumQuestions++;
186.                    dialogueContext.prompedAboutPerson = false;
187.                    return imageWithInteractionPattern;
188.                } else if (!imageWithInteractionPattern.isEmpty() && sentiment == Sentim
    entScore.POSITIVE || sentiment == SentimentScore.NEUTRAL){
189.                    updateMentionedPeople(imageWithInteractionPattern.keySet().iterator(
    ).next());
190.                    dialogueContext.totalNumQuestions++;
191.                    dialogueContext.prompedAboutPerson = false;
192.                    return imageWithInteractionPattern;
193.                } else if (!imageWithInteractionPattern.isEmpty() && sentiment == Sentim
    entScore.NEGATIVE){
194.                    jobsWithImages.clear();
195.                    sentiment = null;
196.                }
197.
198.                imageWithInteractionPattern = selectPersonWithProfilePic(dialogueContext
    .processor, dialogueContext.mentionedPeople);
199.                if (!imageWithInteractionPattern.isEmpty() && !dialogueContext.prompedAb
    outPerson) {
200.                    String personIntroSentence = MessageFormat.format(getRandomString(in
    troPerson), DefaultFunctions.getPersonNameWithRelationship((Person)(dialogueContext.pro
    cessor.eval("person")),patient,currentLocale.getLanguage()));
201.                    tts.speakAndWait(personIntroSentence, false, getAbilityTopic());
202.                    dialogueContext.prompedAboutPerson = true;
203.                    return imageWithInteractionPattern;
204.                } else if (!imageWithInteractionPattern.isEmpty() && (sentiment == Senti
    mentScore.POSITIVE || sentiment == SentimentScore.NEUTRAL)){
205.                    updateMentionedPeople(imageWithInteractionPattern.keySet().iterator(
    ).next());
206.                    dialogueContext.totalNumQuestions++;
207.                    dialogueContext.prompedAboutPerson = false;
208.                    return imageWithInteractionPattern;
209.                } else if (!imageWithInteractionPattern.isEmpty() && sentiment == Sentim
    entScore.NEGATIVE){
210.                    peopleWithProfilePics.clear();
211.                    sentiment = null;
212.                }
213.
214.                // try with photo
```

```
215.                imageWithInteractionPattern = selectPhoto(dialogueContext.processor);
216.                if (!imageWithInteractionPattern.isEmpty()) {
217.                    updateMentionedPeople(imageWithInteractionPattern.keySet().iterator(
    ).next());
218.                    dialogueContext.totalNumQuestions++;
219.                    dialogueContext.prompedAboutPerson = false;
220.                    return imageWithInteractionPattern;
221.                } if (SentimentScore.POSITIVE != null && SentimentScore.NEUTRAL != null)
     {
222.                    getMarriagePicWithPatterns(dialogueContext.processor);
223.                    // getNextInteractionPattern
224.                    return new HashMap<Image, List<PhotoInteractionPattern>>(0);
225.                } else if (SentimentScore.NEGATIVE != null) {
226.                    photos.clear();
227.                    sentiment = null;
228.                }
229.
230.                imageWithInteractionPattern = selectPersonWithProfilePic(dialogueContext
    .processor, dialogueContext.mentionedPeople);
231.                if (!imageWithInteractionPattern.isEmpty()) {
232.                    String personIntroSentence = MessageFormat.format(getRandomString(in
    troPerson), DefaultFunctions.getPersonNameWithRelationship((Person)(dialogueContext.pro
    cessor.eval("person")),patient,currentLocale.getLanguage()));
233.                    tts.speakAndWait(personIntroSentence, false, getAbilityTopic());
234.                    dialogueContext.prompedAboutPerson = true;
235.                    return imageWithInteractionPattern;
236.                } else if (!imageWithInteractionPattern.isEmpty() && (sentiment == Senti
    mentScore.POSITIVE || sentiment == SentimentScore.NEUTRAL)){
237.                    updateMentionedPeople(imageWithInteractionPattern.keySet().iterator(
    ).next());
238.                    dialogueContext.totalNumQuestions++;
239.                    dialogueContext.prompedAboutPerson = false;
240.                    return imageWithInteractionPattern;
241.                } else if (!imageWithInteractionPattern.isEmpty() && sentiment == Sentim
    entScore.NEGATIVE){
242.                    peopleWithProfilePics.clear();
243.                    sentiment = null;
244.                }
245.
246.                // try with person
247.                imageWithInteractionPattern = selectPersonWithProfilePic(dialogueContext
    .processor);
248.                if (!imageWithInteractionPattern.isEmpty()) {
249.                    dialogueContext.totalNumQuestions++;
250.                    dialogueContext.prompedAboutPerson = false;
251.                    return imageWithInteractionPattern;
252.                } else if (!imageWithInteractionPattern.isEmpty() && (sentiment == Senti
    mentScore.POSITIVE || sentiment == SentimentScore.NEUTRAL)){
253.                    updateMentionedPeople(imageWithInteractionPattern.keySet().iterator(
    ).next());
254.                    dialogueContext.totalNumQuestions++;
255.                    dialogueContext.prompedAboutPerson = false;
256.                    return imageWithInteractionPattern;
257.                } else if (!imageWithInteractionPattern.isEmpty() && sentiment == Sentim
    entScore.NEGATIVE){
258.                    peopleWithProfilePics.clear();
259.                    sentiment = null;
260.                }
261.
```

```
262.                // no image with pattern(s)
263.                return new HashMap<Image, List<PhotoInteractionPattern>>(0);
264.            }
265.
266.
267.            private void updateMentionedPeople(Image selectedPhoto) {
268.                List<Person> lastPeopleInPhoto = DefaultFunctions.getPeopleInPhoto(selec
    tedPhoto, patient);
269.                // add to mentioned people
270.                for (Person mentionedPerson : lastPeopleInPhoto) {
271.                    dialogueContext.mentionedPeople.put(mentionedPerson.getId(), mention
    edPerson);
272.                }
273.                System.err.println("MENTIONED PEOPLE: " + dialogueContext.mentionedPeopl
    e.size());
274.            }
275.
276.
277.            private void askYesNoModal() {
278.                ModalButton yesButton = new ModalButton();
279.                yesButton.setAction(YES_ACTION);
280.                yesButton.setContext("neutral");
281.                yesButton.setName(speechBundle.getString("yes-button"));
282.                ModalButton noButton = new ModalButton();
283.                noButton.setAction(NO_ACTION);
284.                noButton.setContext("neutral");
285.                noButton.setName(speechBundle.getString("no-button"));
286.                List<ModalButton> buttons = new ArrayList<>(2);
287.                buttons.add(yesButton);
288.                buttons.add(noButton);
289.                String[] texts = loadDataFromBundle("yes-no-modal-texts", false);
290.                ShowModalMessage message = new ShowModalMessage(MessageFormat.format(spe
    echBundle.getString("yes-no-modal-
    heading"), userManager.getCurrentUser().getFirstName()), texts[RandomUtils.nextInt(0, t
    exts.length)]);
291.                message.setButtons(buttons);
292.                gui.showModal(message, getAbilityTopic());
293.            }
294.
295.        }
```

As you can see now, for example, in the first *try section*, the one for the marriage pattern, as I analyzed it before, there are 3 if's statements based on the **!absence of the image()** and on the **sentiment**.

```
// try with marriage pattern
        imageWithInteractionPattern = getMarriagePicWithPatterns(dialogueContext.processor);
        //sentiment = sentimentAnalyzer.getSentiment(utterance, currentLocale.getLanguage());
        if (!imageWithInteractionPattern.isEmpty() && sentiment == null) {
            updateMentionedPeople(imageWithInteractionPattern.keySet().iterator().next());
            dialogueContext.totalNumQuestions++;
            dialogueContext.prompedAboutPerson = false;
            return imageWithInteractionPattern;
        } else if (!imageWithInteractionPattern.isEmpty() && sentiment == SentimentScore.POSITIVE || sentiment == SentimentScore.NEUTRAL){
            updateMentionedPeople(imageWithInteractionPattern.keySet().iterator().next());
            dialogueContext.totalNumQuestions++;
            dialogueContext.prompedAboutPerson = false;
            return imageWithInteractionPattern;
        } else if (!imageWithInteractionPattern.isEmpty() && sentiment == SentimentScore.NEGATIVE){
            marriagesWithImages.clear();
            sentiment = null;
        }

```

More specifically, if there is no sentiment expressed on the picture you'll be shown a random picture of that pattern: if there is a POSITIVE or NEUTRAL sentiment expressed you'll be shown another picture of that same pattern, else if it's a NEGATIVE sentiment, the one expressed by the patient, the system will clear the array containing all the images of that pattern and will set the sentiment to null (if not the next picture would be based on the sentiment expressed on the one shown before), passing to the next method call.

Chapter 4

Mario Reminiscence: – Development

4.1 Setting up the environment

4.1.1 Which components you need to install/have

In this mini-chapter I'm going to explain what you need to install in order to set all the environment and be ready to start working with MARIO:

- **JAVA:** it's a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible (WORA principle)
- **MARVIN:** it's a toy language written in **Ruby** using **RTLK**. It's not meant for production use, but just as a way to explore writing compilers and interpreters
    - **RUBY:** this is one of the 2 dependencies needed by Marvin, it's a dynamic, reflective, object-oriented, general-purpose programming language
    - **LLVM:** this name was originally an initialism for Low Level Virtual Machine but now has evolved into an umbrella project that has little relationship to what most current developers think of as virtual machines
- **NODE.js:** it's a server-side platform built on Google Chrome's JavaScript engine
- **APACHE KARAF:** Karaf provides dual polymorphic container and application bootstrapping paradigms to the enterprise.
- **MAVEN:** it's a software project management and comprehension tool which can manage a project's build based on the concept of a project object model (POM).
- **MONGODB:** a document database with the scalability and flexibility that you want with the querying and indexing that you need.

4.2 First Setup

4.2.1 Karaf

1. Unzip the ***marvin-karaf.zip*** file and put its content in the ***MARIO-Framework*** folder;
2. Move to ***MARIO-Framework/marvin-karaf/bin*** and start Karaf with ***./karaf clean***;
3. When the Karaf console appears, check if the url ***localhost:8080*** allows accessing the Web console (use ***karaf*** as *username* and *password*)
4. Type ***logout*** in the Karaf console to shutdown Karaf

4.2.2 Marvin

1. Compile and install with maven the two projects in the ***ontology-lizard*** folder
    a. ***mvn clean install*** in ***ontology-lizard/lizard-master***
    b. ***mvn clean install*** in ***ontology-lizard/ontology-api***
2. Copy the ***Mario*** folder in ***MARIO-Framework/marvin-karaf***
3. Compile and install with maven the Marvin framework
    a. ***mvn clean install*** in ***marvin***
4. The ***marvin/karaf-kar*** project generates the Karaf xml feature (and the .kar file) for the Marvin framework
    a. Copy the ***marvin/karaf-kar/target/feature/feature.xml*** file in the ***deploy*** folder of Karaf (***MARIO-Framework/marvin-karaf/deploy***)
    b. Start Karaf with ***./karaf clean*** from ***MARIO-Framework/marvin-karaf/bin***

c. On the karaf console, install the Marvin feature with ***feature:install mario-marvin-kar***

4.2.3 MarioUI

1. Install the node modules for the MarioUI component
    a. Move to ***MARIO-Framework/MarioUI*** and install the modules with ***npm install***

4.3 Regular start-up, interaction and shutdown

4.3.1 starting Marvin and MarioUI

In order to start the core components of MARIO framework (Marvin and MarioUI), execute the following steps:

1. Make sure that **MongoDB** is up and running
2. This is not strictly required, but MongoDB is needed when creating new *patient profiles* for MARIO (additional details are provided later on)
3. Start the **Marvin** framework as follows
4. Move to ***MARIO-Framework/marvin-karaf/bin***
5. Start Karaf with ***./karaf clean***
6. On the Karaf console, install the Marvin feature with ***feature:install mario-marvin-kar***
7. Start the ***MARIO-Framework/MarioUI***
8. Launch MarioUI with ***npm start***

4.3.2 User Profiles and MarioUI

When Marvin is up and running, the so-called *Caregiver UI* is available starting from:

http://localhost:8080/mario/profile.xhtml

The *Caregiver UI* allows creating, configuring and managing user profiles and related settings.

When MarioUI is up and running, the user interface enabling users to interact with MARIO is available at:

http://localhost:3030/

4.3.3 Shutting down Marvin and MarioUI

In order to stop and shutdown the core components of the Mario framework (Marvin and MarioUI); execute the following steps:

1. Stop the **MarioUI** component by pressing *ctrl* + *c* in the corresponding terminal/shell window
2. Stop the **Marvin** framework from the karaf console with the *logout* command
3. The *logout* command shuts down the entire karaf container
4. As an alternative, it is possible to unistall the marvin karaf feature only, using the command *feature:uninstall Mario-marvin-kar*
5. In general it is preferable to shutdown the entire karaf container (and restart it if needed)

4.4 The *My Memories* application

The source code of the *My Memories* is available in *marvin/abilities/reminiscence/src*.

Under the assumption that the maven projects of the Marvin framework have been imported in the Eclipse IDE, the *My Memories* app projects appears as *eu.mario-project.marvin.abilities.reminiscence*.

Any change to the app's source code requires to re-compile and re-install the project with Maven. In the following main steps for re-compiling and re-installing the app using the command line are provided.

- Move to the root folder of the app's project (marvin/abilities/reminiscence)
- Re-install the project with maven *mvn clean install*

IMPORTANT: Any change made to the app requires to restart the core components of the MARIO framework (Marvin and MarioUI) as explained before.

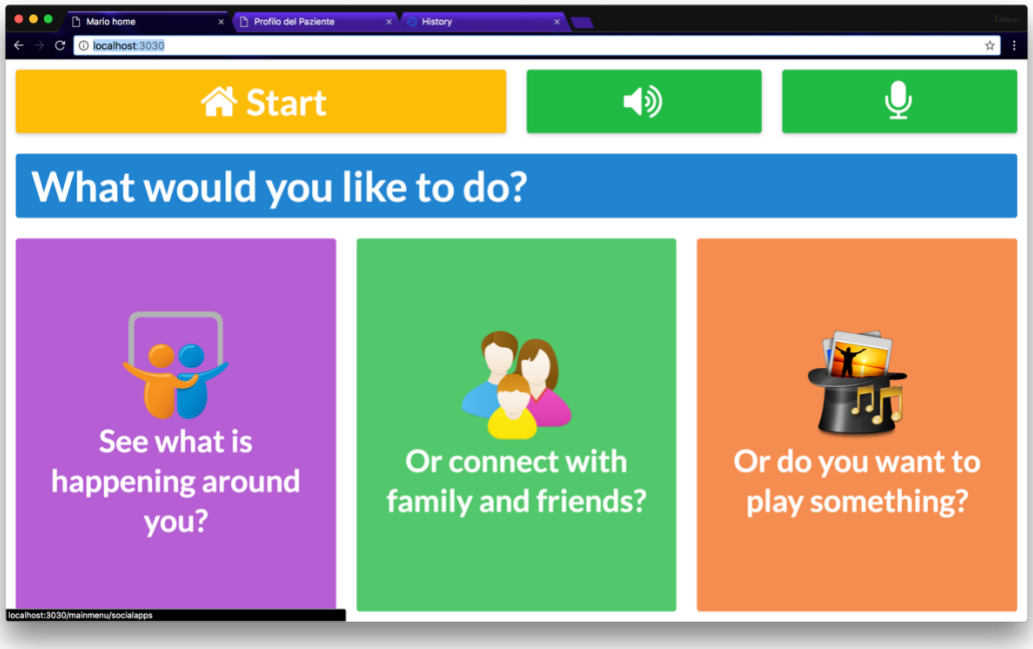
4.5 User Interaction - Demo

Following all the steps before you'll be able to establish a clear connection to execute Mario.

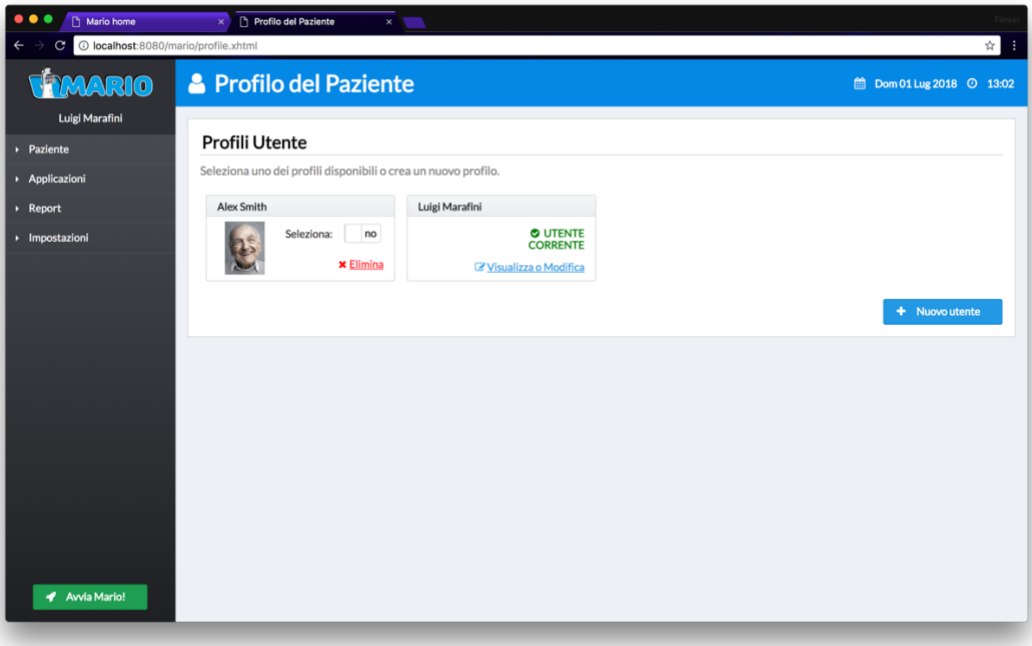
So, if you start from the terminal in the *marvin-karaf/bin* folder, you start karaf with *./karaf clean* and then In the command root of karaf you type *feature:install mario-marvin-kar*.

In another terminal window you need to go to *MARIO-Framework/MarioUI* folder and type *npm start* in order to start MarioUI for the actual patient.

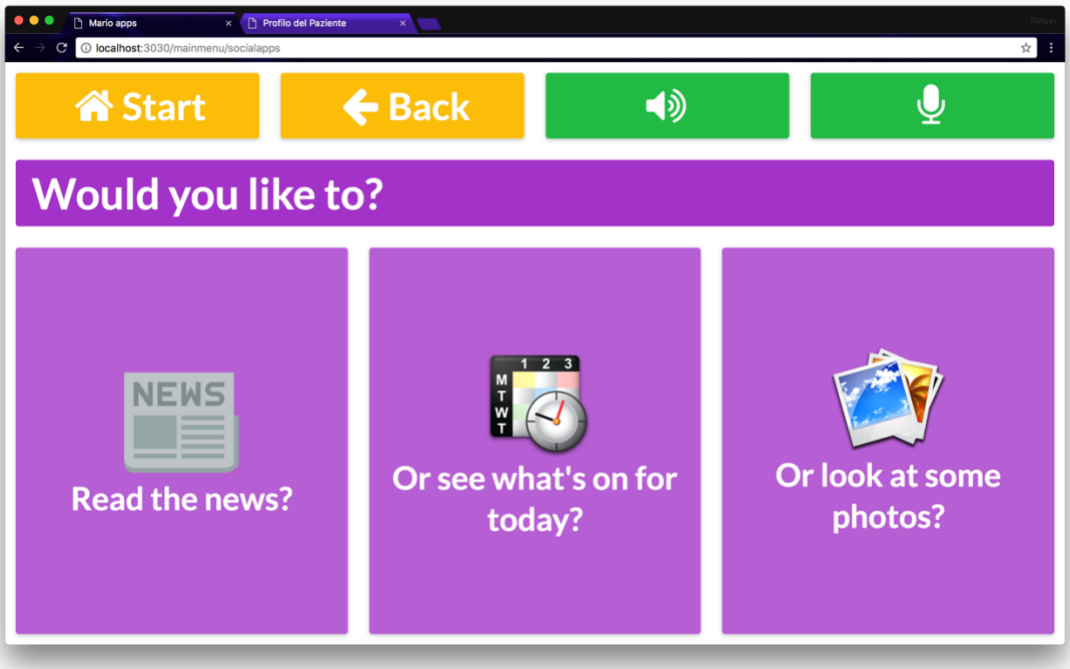
You'll type on the browser http://localhost:3030/ and you'll see this screen:



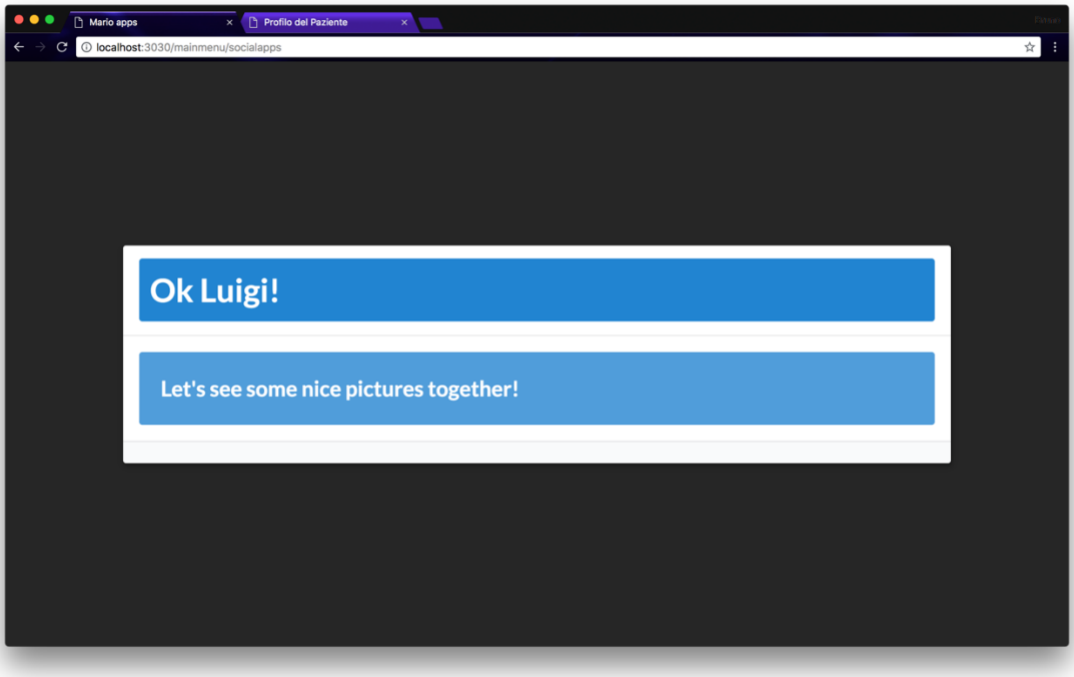
You'll type on the browser http://localhost:8080/mario/profile.xhtml and you'll see something like this:



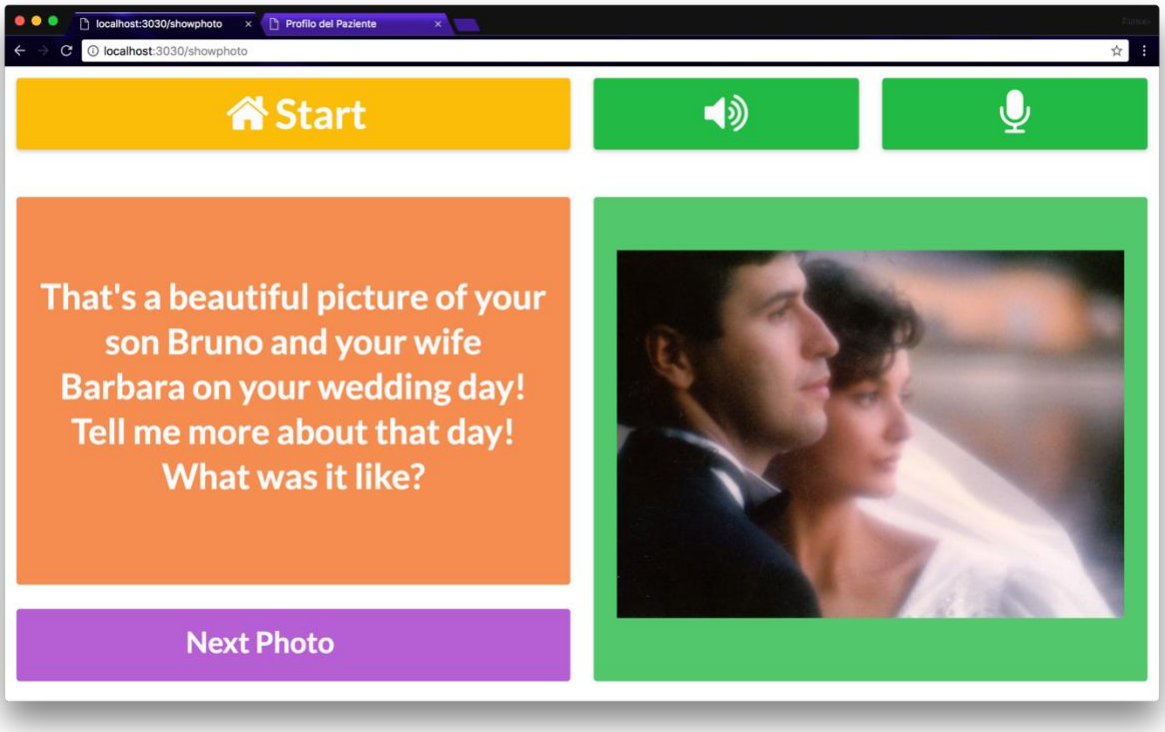
Staying on the first window you can do many things, but If you want to look at some pictures just click on *'See what is happening around you?'* and you will see this page:



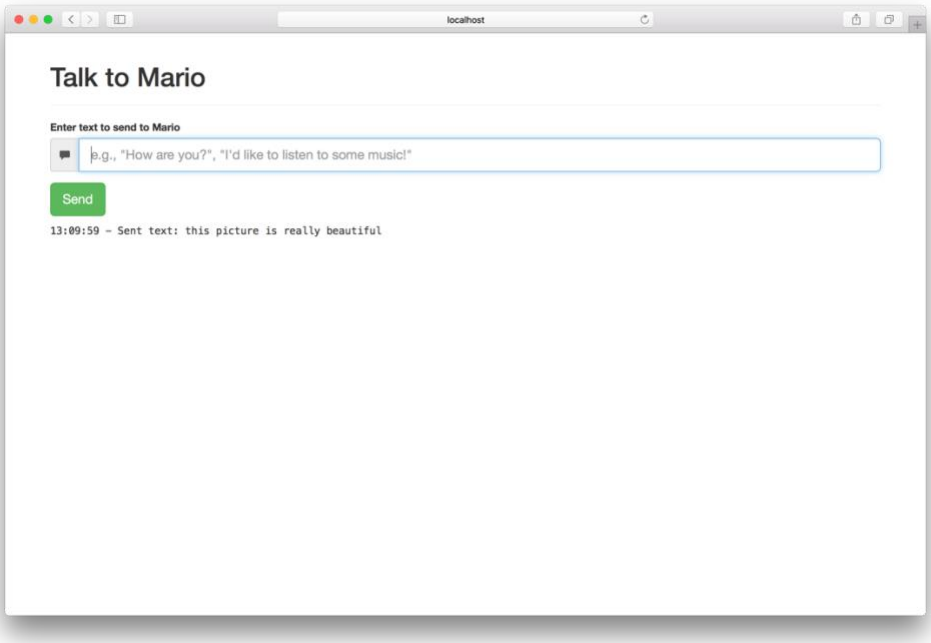
Here, just click on *'Or look at some photos?'*.



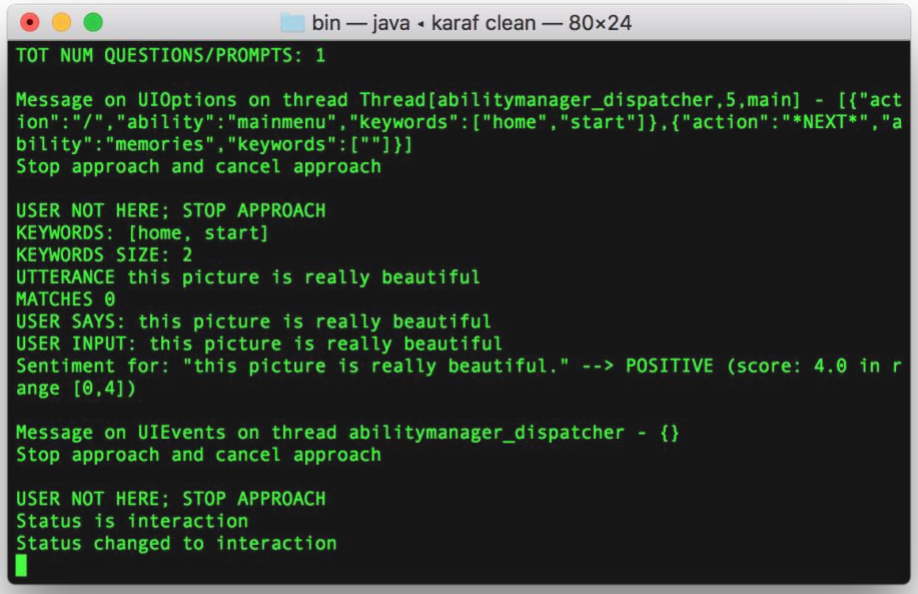
After loading the picture you'll be displayed the first picture with the first message recorded by Mario:



In order to say something to Mario from localhost you can type on another window of your browser this link, http://localhost:8080/static/talk.html, that will redirect you to a page like the screenshot below, from which you are able to interact with the robot.



After this sentence I said for example, in order to see, if sentiment analysis if working or not, you can simply open the terminal window to see what's going on, and you'll find that the sentence in that screen, *'This picture is really beautiful'* is connected to a positive sentiment.



4.6 Code

Personally I've decided to let you see all the modifications I've made to the code uploading all the code on GitHub.

So the link to the code is here and public, so accessible to everyone: Code of Reminiscence.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, I addressed many problems, starting from a simply demographic problem, going through elder people and difficulties with the use of technology, Robot Revolution and so the growth of robots for dementia care and then I've introduced my problem so the one of showing picture to the patient based on sentiment analysis.

The main focus of my thesis was the one to let you know MARIO (for the one who didn't know the project at all) and introduce its architecture and design with my contribution.

I've tried to establish a new kind of approach with an original solution in order to understand a new user experience.

The work I've presented has been already tested with some patients (especially with me and my grandmother) and has already been deployed in the main program of the project.

Surely, this application I've changed goes on with the already well-known reminiscence application presented by some of my Mentors last year that consists of a Knowledge-Base system, previously described, broadly as an interface engine and knowledge-base.

The interface engine acts as the search engine and the knowledge base acts as the knowledge repository.

It has been really a great pleasure to work with a Team of Researchers like the one at CNR in Rome because there is a really focused group: every day a set of objectives to reach and always concentrated on future work.

For this reason I intend to keep working with them for the development of future apps for MARIO.

5.2 Future Work

Many different adaptations, tests, and experiments have been left for the future due to lack of time. Future work concerns deeper analysis of particular mechanisms, new proposals to try different methods, or simply curiosity.

The research presented has many possibilities for further work. Although it focuses on older users as the target user group, the principles of intelligent, adaptable interfaces and dynamic multimodal input and output could be utilised in other areas where the diversity and needs of users are of paramount importance.

For example, work could be extended to include users with specific or combined disabilities, where the interface could adapt to address individual needs appropriately.

In addition to extending the work to other user groups, a number of areas requiring further investigation and also further development have been identified for mobile interfaces and multimodal interaction for older people. The results could be formally incorporated into sets of guidelines for the development of mobile device interfaces and the development of applications to support reminiscence.

Ongoing and future work includes the ability to acquire factual knowledge from the conversational interaction with the user, as well as the introduction of sentiment analysis capabilities to assess emotional aspects, identify triggers generating positive feelings to be favoured in reminiscence and based on these factors, the ability to learn, from what the patient says, more than just establishing if the patient likes or not the picture shown.

In addition to this, there are still a lot of apps, included in the robot system, not implemented (just declared); my intention is to keep developing apps for this robots in order to let it be perfect for distributing itself in big companies.

5.3 Acknowledgements

The research leading to these results has received funding from the European Union Horizon 2020 – the Framework Programme for Research and Innovation (2014-2020) under grant agreement 643808 Project MARIO “Managing active and healthy aging with use of caring service robots”.

Tables

Table 1

The use of new technologies by country (% using technology); source: MOBILATE survey 2000, N = 3950, weighted; a = total realized sample in country; b = number of people using at least one technology; c = % of users of total sample: users

	Finland	East Germany	West Germany	Hungary	Italy	Netherlands	Total
Automatic teller machine	65	89	79	3	22	77	66
Electronic cash (PIN)	35	37	32	61	19	79	41
Ticket automat	78	33	38	17	7	27	37
Automatic admission	37	10	32	0	93	16	35
Mobile telephones	63	16	12	32	32	40	32
Card-operated phone	16	23	39	28	37	36	30
Internet	18	2	3	1	6	13	8
Telebanking	6	3	3	5	2	6	4
% Of total sample [c]	83	73	77	16	75	75	67
Total of users [b]	508	562	574	95	451	465	2655
N = total sample [a]	610	768	750	605	600	617	3950

Table 2

Users (%) of new technologies by strata of the sample; source: MOBILATE survey 2000, N = 3950, not weighted

	Urban				Rural				total
	55 - 74		75+		55 - 74		75+		
	Male	Female	Male	Female	Male	Female	Male	Female	
Automatic teller machine	72	67	50	40	70	72	54	52	63
Electronic cash (PIN)	48	39	23	22	44	41	27	30	37
Ticket automat	42	47	39	39	25	30	31	34	36
Automatic admission	38	41	40	36	29	26	24	15	33
Mobile telephones	42	29	26	16	33	27	18	15	28
Card-operated phone	31	33	18	15	30	36	13	16	27
Internet	14	8	4	2	6	4	2	1	6
Telebanking	5	3	3	4	6	1	2	0	3
% of total sample	84	80	60	50	71	66	47	34	63
Total users	458	424	265	229	390	365	206	149	2486
Total sample	545	532	440	456	547	551	440	439	3950

Table 3

Logistic regression (by SPSS23) for the use of ATM, ticket dispenser and PIN-payment (1 = use of technology); significance based on Wald statistic and expr (B) = odds ratio of the row independent with the dependent; source MOBILATE 2000, N = 3950, weighted

	ATM		PIN payment		Ticket automat (selection of users of public transport)	
	Sig.	Exp(B)	Sig.	Exp(B)	Sig.	Exp(B)
Gender, 0= male and 1= female	0.46	1.07	0.12	1.15	0.63	0.95
Age, number of years	0.00	0.97	0.00	0.97	0.00	0.98
Education, years of education	0.00	1.11	0.00	1.08	0.01	1.04
Household size, 0= alone, 1= more than 1	0.10	0.84	0.00	0.72	0.78	1.03
ADL-scale, number of problems with Activities of Daily Living (sum-score 0= no problems, 20= many problems)	0.00	0.94	0.00	0.95	0.00	0.95
Urban (0) or rural (1) location	0.00	0.66	0.08	0.85	0.73	1.04
Working memory, Digit Symbol Test: a method for measuring the cognitive functioning (number of correct substitutions (0 to maximum 67)	0.00	1.03	0.00	1.03	0.00	1.02
Income, 0= low or middle and 1= high (income groups in each country divided in 3 more or less equal parts of one third of the sample)	0.00	0.23	0.00	0.53	0.00	0.40
Physical mobility, 0= not or moderately active and 1= (very) active (the sum-score of 10 statements concerning Activities of Daily Living, for instance: climbing stairs 0=without difficulty, 1= with difficulty, 2= can not)	0.01	1.43	0.85	0.98	0.68	1.07
Self rated physical mobility, 0= very poor and moderate 1= (very) good	0.85	1.02	0.00	0.68	0.01	1.36
Security feeling during day, 0= (very) insecure or moderately secure and 1= very secure	0.00	0.69	0.14	0.84	0.47	1.10
Security feeling during night, 0=(very) insecure or moderately secure and 1= very secure	0.16	1.18	0.04	0.79	0.92	1.01
Constant	0.00	6.25	0.53	1.35	0.21	2.03
Improved prediction	from 51% to 73%		from 71% to 73%		from 61% to 67%	

Table 4

Classification of Recommender Systems Research

Recommendation Approach	Recommendation Technique	
	Heuristic-based	Model-based
Content-based	Commonly used techniques: • TF-IDF (information retrieval) • Clustering Representative research examples: • Lang 1995 • Balabanovic & Shoham 1997 • Pazzani & Billsus 1997	Commonly used techniques: • Bayesian classifiers • Clustering • Decision trees • Artificial neural networks Representative research examples: • Pazzani & Billsus 1997 • Mooney et al. 1998 • Mooney & Roy 1999 • Billsus & Pazzani 1999, 2000 • Zhang et al. 2002
Collaborative	Commonly used techniques: • Nearest neighbor (cosine, correlation) • Clustering • Graph theory Representative research examples: • Resnick et al. 1994 • Hill et al. 1995 • Shardanand & Maes 1995 • Breese et al. 1998 • Nakamura & Abe 1998 • Aggarwal et al. 1999 • Delgado & Ishii 1999 • Pennock & Horwitz 1999 • Sarwar et al. 2001	Commonly used techniques: • Bayesian networks • Clustering • Artificial neural networks • Linear regression • Probablistic models Representative research examples: • Billsus & Pazzani 1998 • Breese et al. 1998 • Ungar & Foster 1998 • Chien & George 1999 • Getoor & Sahami 1999 • Pennock & Horwitz 1999 • Goldberg et al. 2001 • Kumar et al. 2001 • Pavlov & Pennock 2002 • Shani et al. 2002 • Yu et al. 2002, 2004 • Hofmann 2003, 2004 • Marlin 2003 • Si & Jin 2003
Hybrid	Combining content-based and collaborative components using: • Linear combination of predicted ratings • Various voting schemes • Incorporating one component as a part of the heuristic for the other Representative research examples: • Balabanovic & Shoham 1997 • Claypool et al. 1999 • Good et al. 1999 • Pazzani 1999 • Billsus & Pazzani 2000 • Tran & Cohen 2000 • Melville et al. 2002	Combining content-based and collaborative components by: • Incorporating one component as a part of the model for the other • Building one unifying model Representative research examples: • Basu et al. 1998 • Condliff et al. 1999 • Soboroff & Nicholas 1999 • Ansari et al. 2000 • Popescul et al. 2001 • Schein et al. 2002

Table 5

Reminiscence	Function	Self as...
Integrative	Drawing from memories a sense of meaning and coherence in one's life	Meaningful, coherent
Instrumental	Using past experiences to solve curreny difficlties and to cope	Active, competent
Transmissive	Recalling personal experiences and transmitting an instructive story	Experienced, wise
Narrative	Telling a story	Communicative, entertaining
Escapism	Escaping from the present and finding refuge in a glorified past	Blocked, futureless
Obsessive	Ruminating on unresolved past difficulties and traumas	Fragmented meaningless
Death preparation	Using memories to deal with the issue of death	Transcendent, spiritual
Intimacy	Keeping in contact with a departed person through memories	Connected, faithful

Figure 1 - Technology and Older People

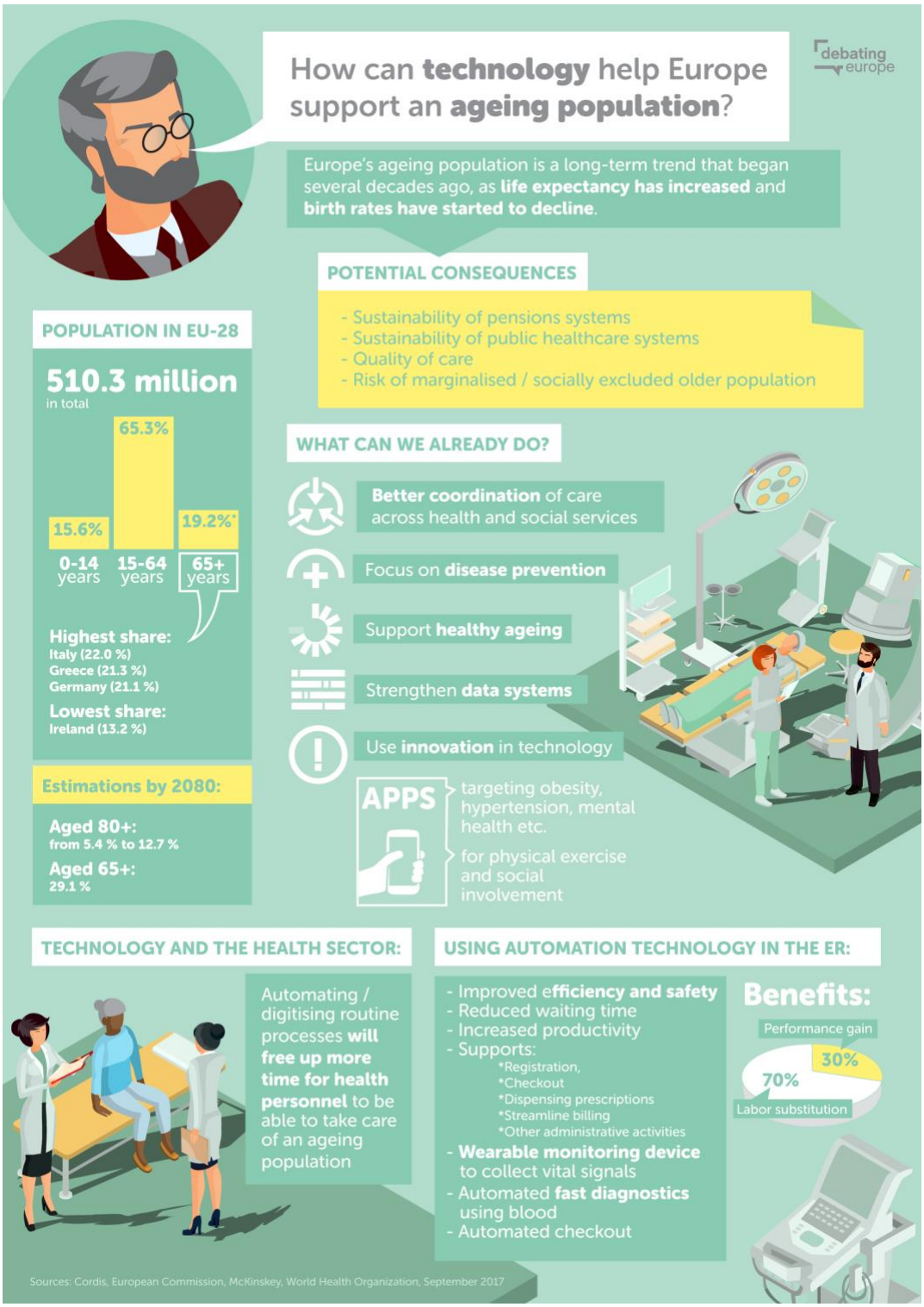


Figure 2 - Pew Research Centert

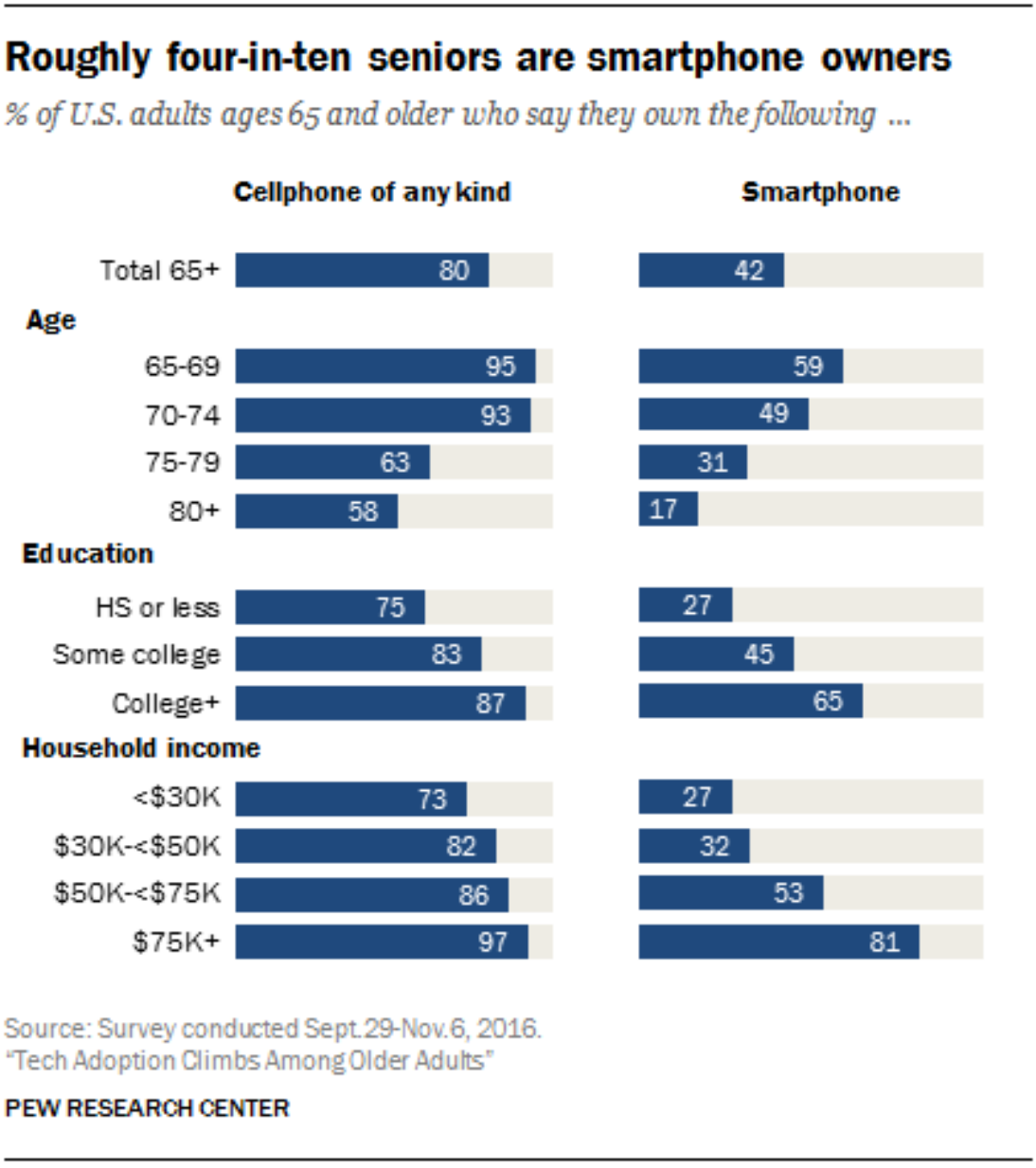


Figure 3 - Component Diagram

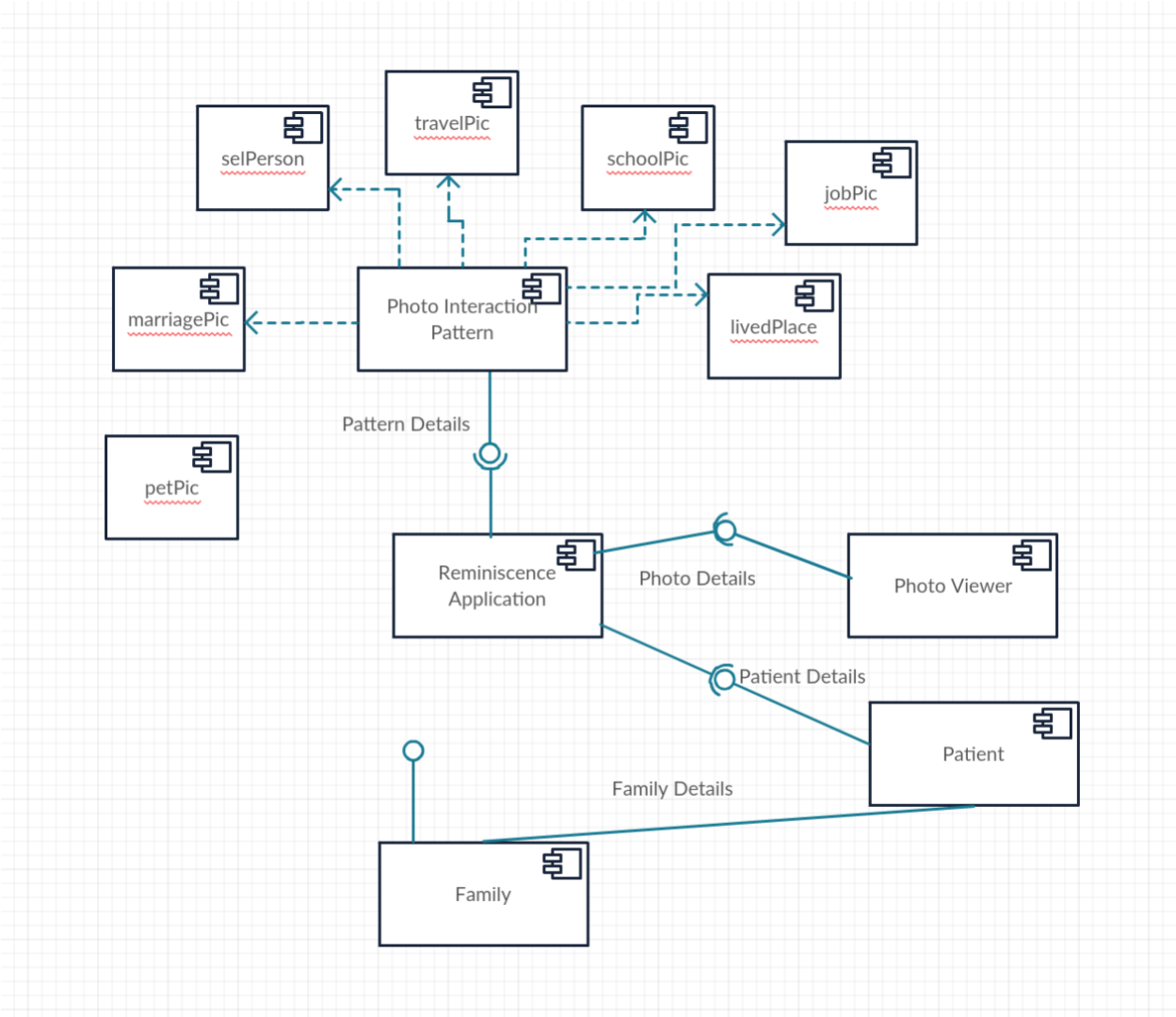


Figure 4 - Class Diagram

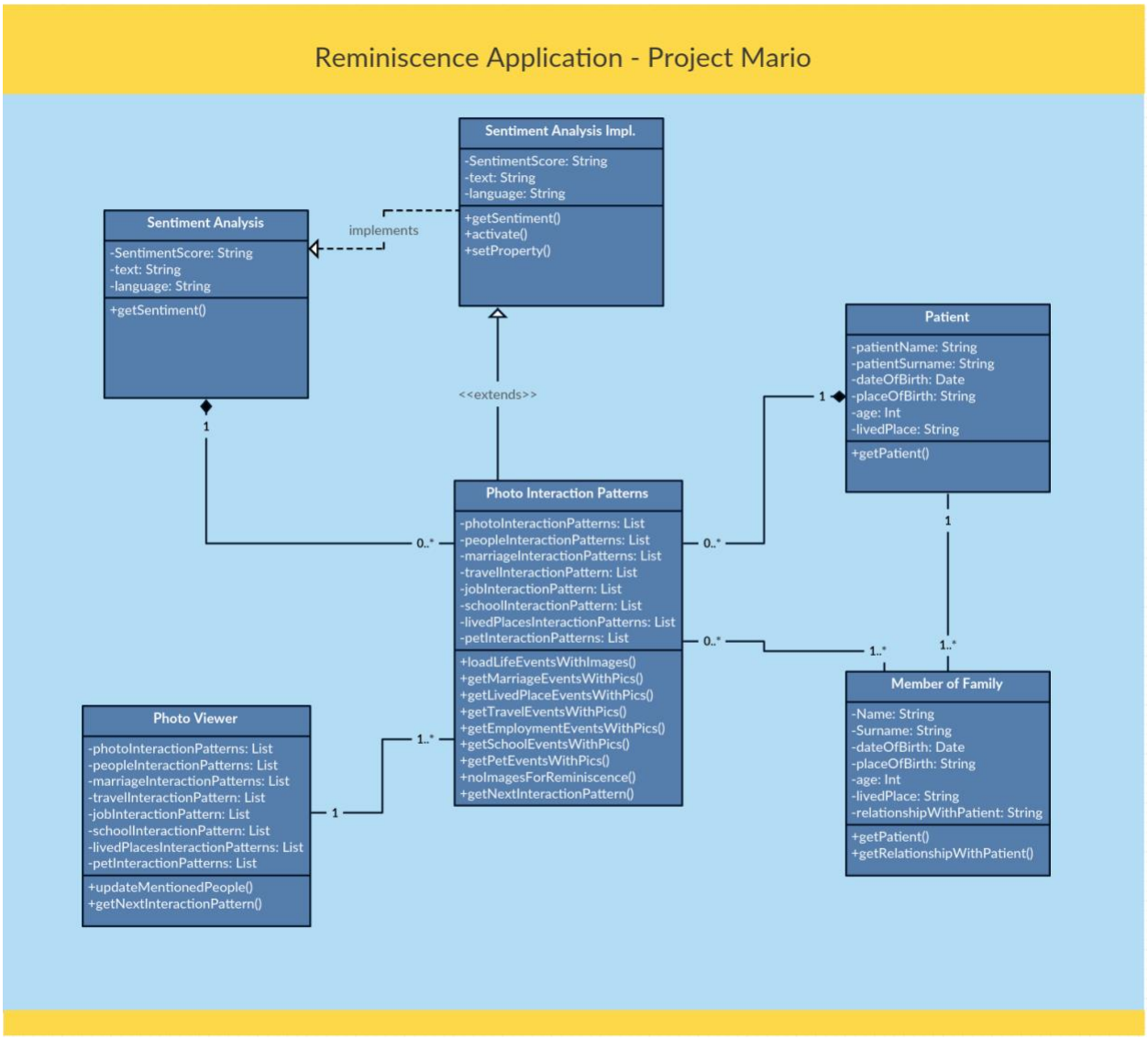


Figure 5 - Sequence Diagram

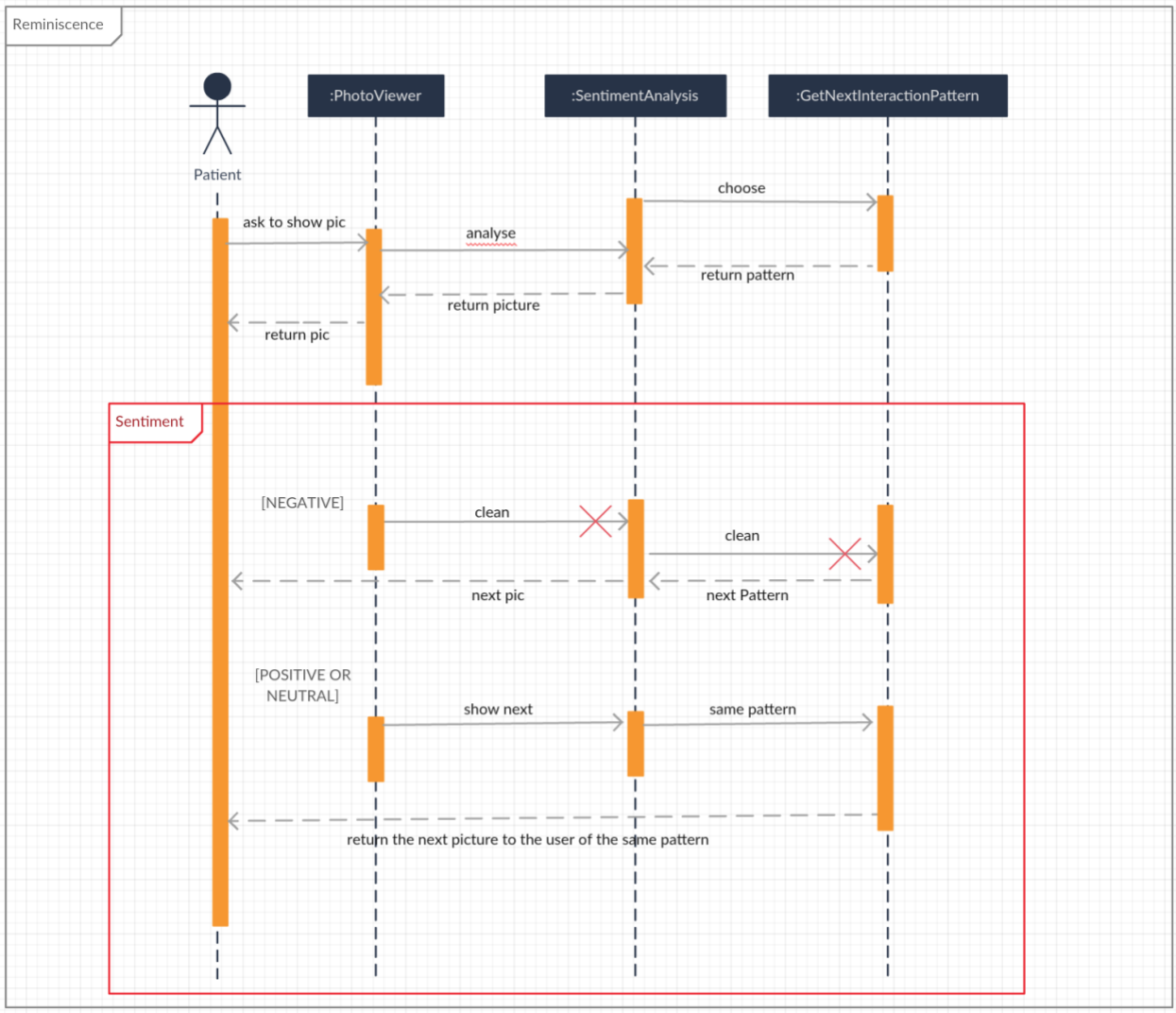


Figure 6 - Top Level MON

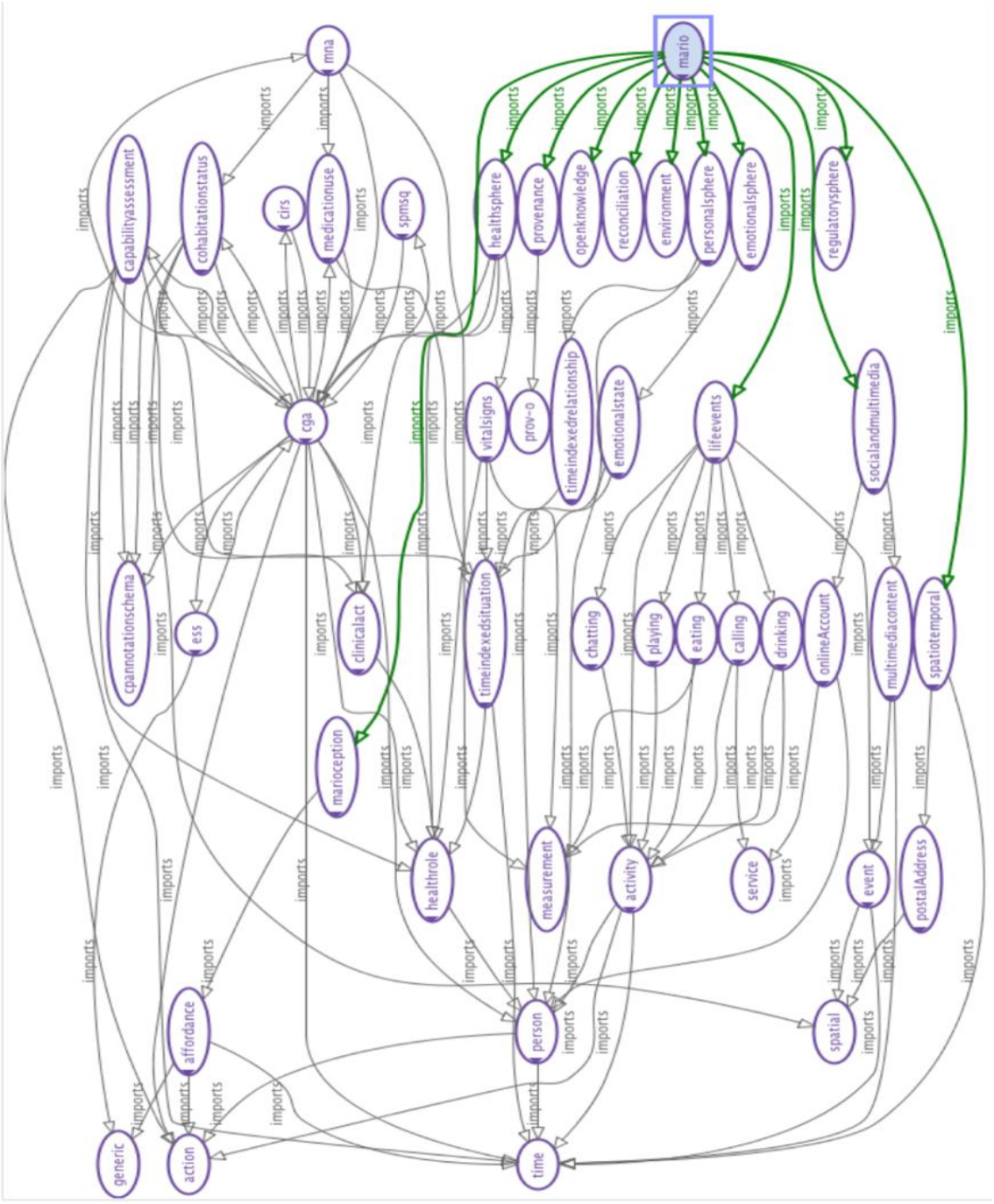
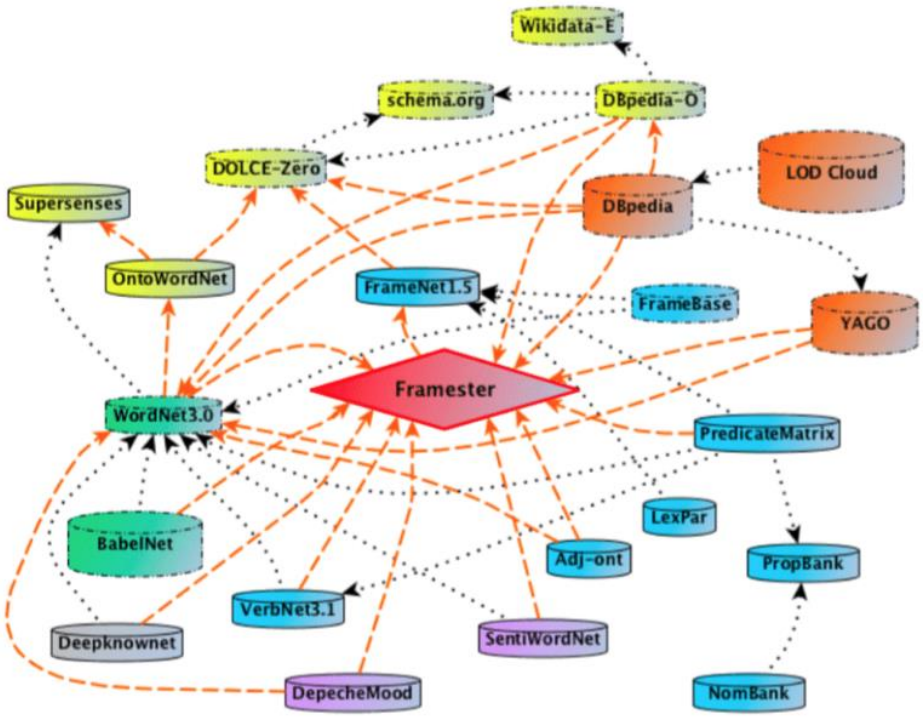


Figure 7 – Framester



Bibliography

1. **D. Brooker, and L. Duceb.** "Well-being and activity in dementia: A comparison of group reminiscence therapy, structured goal-directed group activity, and unstructured time," Aging & Mental Health, 4, pp. 354-358, 2000. [Online]

2. **A. Purves, A. Phinney, W. Hulko, G. Puurveen, A. J. Astell.** "Developing CIRCA-BC and exploring the role of the computer as a third participant in conversation," American Journal of Alzheimer's Disease & Other Dementias, Vol. 30, No.1, pp. 101-107, 2014. [Online]

3. **N. Kuwahara, S. Abe, K. Yasuda, and K. Kuwabara.** "Networked reminiscence therapy for individuals with dementia by using photo and video sharing," Proc. ASSETS, pp. 125-132, 2006. [Online]

4. *Memory and consciousness.* **Tulving, Endel.** 1, 1985, Canadian Psychology, Vol. 26, pp. 1–12.

5. *Anthony J Gibson OAM: clinical teacher.* **Cole, Barry L.** 6, 2004, Clinical and Experimental Optometry, Vol. 87, pp. 401-402.

6. *Integrative and instrumental reminiscence therapies for depression in older adults: Intervention strategies and treatment effectiveness.* **Cappeliez, P. and Watt, L. M.** 2, Aging & Mental Health, Vol. 4, pp. 166–177.

7. *Coping mediates the relationships between reminiscence and psychological well-being among older adults.* **Robitaille, Annie and Cappeliez, Philippe.** 7, Aging & Mental Health, Vol. 14, pp. 807–818.

8. *An Explanation of the Reminiscence Bump in the Dreams of Older Adults in Terms of Life Goals and Identity.* **Cappeliez, Philippe.** 1, 2008, Self and Identity, Vol. 7, pp. 25-33.

9. **Gary Kenyon, Ernst Bohlmeijer, William L. Randall.** *Storying Later Life: Issues, Investigations and Interventions in Narrative Gerontology.* Oxford : Oxford University Press, 2011.

10. *NURTURING THE COGNITIVE SKILLS IN HEALTH.* **Havighurst, Robert J.** 2, 1972, Journal of School Health, Vol. 42, pp. 73-75.

11. *Social enjoyment with electronic photograph displays: Awareness and control.* **Lindley, Siân E. and Monk, Andrew F.** 8, 2008, International Journal of Human-computer Studies \/ International Journal of Man-machine Studies, Vol. 66, pp. 587-604.

12. *Functional Changes in Children, Adolescents, and Young Adults with Cerebral Palsy.* **Krakovsky, Gina, et al.** 4, 2007, Research in Developmental Disabilities, Vol. 28, pp. 331-340.

13. **Hyams, Adriana and Scogin, Forrest.** Reminiscence/Life Review Therapy. [Online] 2014. [Cited: 2 3, 2018.] http://onlinelibrary.wiley.com/doi/10.1002/9781118625392.wbecp148/pdf.

14. *Psychological Characteristics of the Haight "Junkie".* **Bleckner, Janet E.** 1, 1974, Journal of Psychoactive Drugs, Vol. 6, pp. 21-28.

15. *Long-term concentrative meditation and cognitive performance among older adults.* **Prakash, Ravi, et al.** 4, 2012, Aging Neuropsychology and Cognition, Vol. 19, pp. 479-494.

16. **Craig, David, Nugent, Chris D. and Mulvenna, Maurice.** Healthcare Technologies for Older People: What Do Physicians Think? [Online] 2006. [Cited: 2 3, 2018.] http://uir.ulster.ac.uk/16511.

17. **McCurry, Justin.** The Lancet medical journal. *The Lancet.* [Online] October 17, 2015. http://www.thelancet.com/journals/lancet/article/PIIS0140-6736(15)00525-5/fulltext.

18. **Hay, Mark.** The country is producing robots. *Good.* [Online] June 24, 2015. https://www.good.is/articles/robots-elder-care-pepper-exoskeletons-japan.

19. **Griffiths, Andrew.** Robots help patients who suffer from dementia . *The Guardian.* [Online] July 8, 2014. https://www.theguardian.com/society/2014/jul/08/paro-robot-seal-dementia-patients-nhs-japan.

20. **Okayama University, Japan Advanced Institute of Science and Technology, University of Cambridge.** Dementia Care. *National Institutes of Health.* [Online] https://www.ncbi.nlm.nih.gov/pubmed/24339091.

21. **David Feil-Seifer, Maja J. Matarić.** Socially Assistive Robotics. *IEEEXPLORE.* [Online] April 19, 2011. http://ieeexplore.ieee.org/document/5751968/?reload=true&arnumber=5751968.

22. **K., Shibata T. - Wada.** Robot Therapy: A new Approach for Mental Healthcare of the Elderly . *Karger.* [Online] 2010. https://www.karger.com/Article/Abstract/319015.

23. **Ontology. *Wikipedia: The Free Encyclopedia.* [Online] [Cited: 2 3, 2018.] http://en.wikipedia.org/wiki/Ontology.**

24. *Ten years of Applied Ontology.* Guarino, Nicola and Musen, Mark A. 2015, Applied Ontology, Vol. 10, pp. 169-170.

25. Lee, Jae-Hyun and Suh, Hyo-Won. OWL-Based Product Ontology Architecture and Representation for Sharing Product Knowledge on a Web. [Online] 2007. [Cited: 2 3, 2018.] http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1604083.

26. V., Delaunay P. - Guerci A. - Consigliere S. - Toucs. DelaunayP, Toucs V. Gérontologie et nouvelles technologies. In: Guerci A, Consigliere S, editors. Living Old Age: Western world and Modernization. Genova: Erga edizioni; 2002; pp145-149. *Gérontologie et nouvelles technologies.* [Online]

27. M., Klerk. Klerk M. de Rapportage ouderen 2001, veranderingen in de leefsituatie (Report of older people 2001, Changes in life situation). The Hague: SCP; 2001. [Online]

28. Marcellini F, Mollenkopf H, Spazzafumo L, Ruoppila I. Acceptance and use of technological solutions by the older people in the outdoor environment: findings from a European survey. Zeitschrift für Gerontologie und Geriatrie 2000; 33(3):169-177. [Online]

29. Marcellini F, Marcucci A, Freddi A. Older people's acceptance and use of technological devices in everyday life. In: Guerci A, Consigliere S, editors. Living Old Age: Western World and Modernization. Genova: Erga edizioni; 2002; pp150-163. [Online]

30. *Article Title.* Last Name, First, Middle. Year, Journal Title, pp. Pages From - To.

31. Last Name, First, Middle. *Book Title.* City Name : Publisher Name, Year.

32. Debating Europe. *Debate.* [Online] September 19, 2017. http://www.debatingeurope.eu/2017/09/19/can-technology-meet-challenge-europes-ageing-population/#.WnL5V5M-cXo.

33. *Effects of cognitive training interventions with older adults: a randomized controlled trial.* Ball, Karlene K., et al. 18, 2002, JAMA, Vol. 288, pp. 2271-2281.

34. —.Ball, Karlene K., et al. 18, 2002, JAMA, Vol. 288, pp. 2271-2281.

35. ***Evaluating feature selection methods for learning in data mining applications.*** **Piramuthu, Selwyn. 2, 2004, European Journal of Operational Research, Vol. 156, pp. 483-494.**

36. Dickinson, Anna, et al. Domesticating technology: in-home requirements gathering with frail older people? [Online] 2003. [Cited: 2 1, 2018.] http://researchrepository.napier.ac.uk/3789.

37. Eisma, Rose, et al. Mutual inspiration in the development of new technology for older people. [Online] 2003. [Cited: 2 1, 2018.] http://www-edc.eng.cam.ac.uk/~jag76/research/2003_include_general/paper.pdf.

38. ***Early user involvement in the development of information technology-related products for older people.*** **Eisma, Rosine, et al. 2, 2004, Universal Access in The Information Society, Vol. 3, pp. 131-140.**

39. ***A Constructivist Look at Life Roles.*** **Brott, Pamelia E. 2, 2005, Career Development Quarterly, Vol. 54, pp. 138-149.**

40. ***Psychological Characteristics of the Haight "Junkie".*** **Bleckner, Janet E. 1, 1974, Journal of Psychoactive Drugs, Vol. 6, pp. 21-28.**

41. Larkey, Leah S. Automatic essay grading using text categorization techniques. [Online] 1998. [Cited: 2 3, 2018.] http://dl.acm.org/citation.cfm?id=290965.

42. ***Facing the cold start problem in recommender systems.*** **Lika, Blerina, Kolomvatsos, Kostas and Hadjiefthymiades, Stathes. 4, 2014, Expert Systems With Applications, Vol. 41, pp. 2065-2073.**

43. Sahebi, Shaghayegh and Cohen, William. Community-Based Recommendations: a Solution to the Cold Start Problem. [Online] 2011. [Cited: 2 3, 2018.] http://d-scholarship.pitt.edu/13328.

44. Embarak, Ossama H. A method for solving the cold start problem in recommendation systems. [Online] 2011. [Cited: 2 3, 2018.] http://ieeexplore.ieee.org/xpl/articledetails.jsp?tp=&arnumber=5893824&contenttype=conference+publications&querytext=cold+start.

45. Abdollahi, Behnoush and Nasraoui, Olfa. A cross-modal warm-up solution for the cold-start problem in collaborative filtering recommender systems. [Online] 2014. [Cited: 2 3, 2018.] http://dl.acm.org/citation.cfm?id=2615569.2615665.

46. Kobsa, Alfred. Recent user modeling research in Germany. [Online] 1999. [Cited: 2 3, 2018.] https://link.springer.com/chapter/10.1007/978-3-7091-2490-1_51.

47. —. User modeling and user-adapted interaction. [Online] 1994. [Cited: 2 3, 2018.] https://link.springer.com/article/10.1007/s11257-005-6468-9.

48. Sebastiani, Paola, Ramoni, Marco F. and Cohen, Paul R. Sequence Learning via Bayesian Clustering by Dynamics. [Online] 2001. [Cited: 2 3, 2018.] http://link.springer.com/content/pdf/10.1007/3-540-44565-x_2.pdf.

49. *Editorial: Methods in Functional Genomics.* Sebastiani, Paola, Kohane, Isaac S. and Ramoni, Marco F. 2003, Machine Learning, Vol. 52, pp. 5-9.

50. Find and Browse Music on Last.fm — Last.fm. [Online] [Cited: 2 3, 2018.] http://www.last.fm/music/.

51. Liu, Ranran and Jin, Zhengping. An Improved Graph-based Recommender System for Finding Novel Recommendations among Relevant Items. [Online] 2015. [Cited: 2 3, 2018.] http://atlantis-press.com/php/download_paper.php?id=25845066.

52. *Recommender Systems: An Overview.* Burke, Robin D., Felfernig, Alexander and Göker, Mehmet H. 3, 2011, Ai Magazine, Vol. 32, pp. 13-18.

53. Montaner, Miquel, López, Beatriz and Rosa, Josep Lluís de la. EVALUATION OF RECOMMENDER SYSTEMS THROUGH SIMULATED USERS. [Online] 2004. [Cited: 2 5, 2018.] http://dblp.uni-trier.de/db/conf/iceis/iceis2004-3.html.

54. *Collaboration Analysis in Recommender Systems Using Social Networks.* Palau, Jordi, et al. 2004, Lecture Notes in Computer Science, pp. 137-151.

55. Montañés, Elena, et al. Collaborative tag recommendation system based on logistic regression. [Online] 2009. [Cited: 2 5, 2018.] http://ceur-ws.org/vol-497/paper_20.pdf.

56. *Research Paper Recommender System Evaluation: A Quantitative Literature Survey.* Beel, J., et al. Proceedings of the Workshop on Reproducibility and Replication in Recommender Systems Evaluation (RepSys) at the ACM Recommender System Conference (RecSys).

57. *Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions.* Adomavicius, G. and Tuzhilin, A. 6, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, pp. 734–749.

58. Adaptive Info. *AdaptiveInfo.* [Online] www.AdaptiveInfo.com.

59. *Recommender Systems: An Overview.* Burke, Robin D., Felfernig, Alexander and Göker, Mehmet H. 3, 2011, Ai Magazine, Vol. 32, pp. 13-18.

60. *Providing Justifications in Recommender Systems.* Symeonidis, Panagiotis, Nanopoulos, Alexandros and Manolopoulos, Yannis. 6, 2008, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 38, pp. 1262-1272.

61. Sancarlo, Daniele, et al. MARIO Project: A Multicenter Survey About Companion Robot Acceptability in Caregivers of Patients with Dementia. [Online] 2016. [Cited: 2 5, 2018.] https://link.springer.com/chapter/10.1007/978-3-319-54283-6_24.

62. ***An adaptive approach to dealing with unstable behaviour of users in collaborative filtering systems.*** **Rafeh, Reza and Bahrehmand, Arash. 3, 2012, Journal of Information Science, Vol. 38, pp. 205-221.**

63. ***Collaborative Filtering Bandits.*** **Li, Shuai, Karatzoglou, Alexandros and Gentile, Claudio. 2016, arXiv: Learning, pp. 539-548.**

64. Ilhami, Mirza and Suharjito. Film recommendation systems using matrix factorization and collaborative filtering. [Online] 2014. [Cited: 2 5, 2018.] http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7048228.

65. Weigert, Thomas, et al. Automated Code Generation for Industrial-Strength Systems. [Online] 2008. [Cited: 2 5, 2018.] http://dblp.uni-trier.de/db/conf/compsac/compsac2008.html.

66. Zheng, Yong. Criteria Chains: A Novel Multi-Criteria Recommendation Approach. [Online] 2017. [Cited: 2 5, 2018.] http://dblp.uni-trier.de/db/conf/iui/iui2017.html.

67. Hdioud, Ferdaous, Frikh, Bouchra and Ouhbi, Brahim. Multi-Criteria Recommender Systems based on Multi-Attribute Decision Making. [Online] 2013. [Cited: 2 5, 2018.] http://dl.acm.org/citation.cfm?id=2539176.

68. Zheng, Yong. Situation-aware multi-criteria recommender system: using criteria preferences as contexts. [Online] 2017. [Cited: 2 5, 2018.] http://dblp.uni-trier.de/db/conf/sac/sac2017.html.

69. Mikeli, Angeliki, Apostolou, Dimitris and Despotis, Dimitris K. A Multi-criteria Recommendation Method for Interval Scaled Ratings. [Online] 2013. [Cited: 2 5, 2018.] http://dl.acm.org/citation.cfm?id=2569232.

70. ***Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions.*** **Adomavicius, G. and Tuzhilin, A. 6, IEEE Transactions on Knowledge and Data Engineering, Vol. 17.**

71. Jambor, Tamas. Intelligent media indexing and television recommender systems. [Online] 2009. [Cited: 2 5, 2018.] http://dl.acm.org/citation.cfm?id=2227305.

72. ***A reliability-based recommendation method to improve trust-aware recommender systems.*** **Moradi, Parham and Ahmadian, Sajad. 21, 2015, Expert Systems With Applications, Vol. 42, pp. 7386-7398.**

73. Marcellini F, Mollenkopf H, Spazzafumo L, Ruoppila I. Marcellini F, Mollenkopf H, Spazzafumo L, Ruoppila I. Acceptance and use of technological solutions by the older people in the outdoor environment: findings from a European survey. Zeitschrift für Gerontologie und Geriatrie 2000; 33(3):169-177. [Online]

74. CREATE. CREATE is a 36 months research project (2015-2018) funded by the European Commission under Horizon2020. [Online] http://www.create-mobility.eu/.