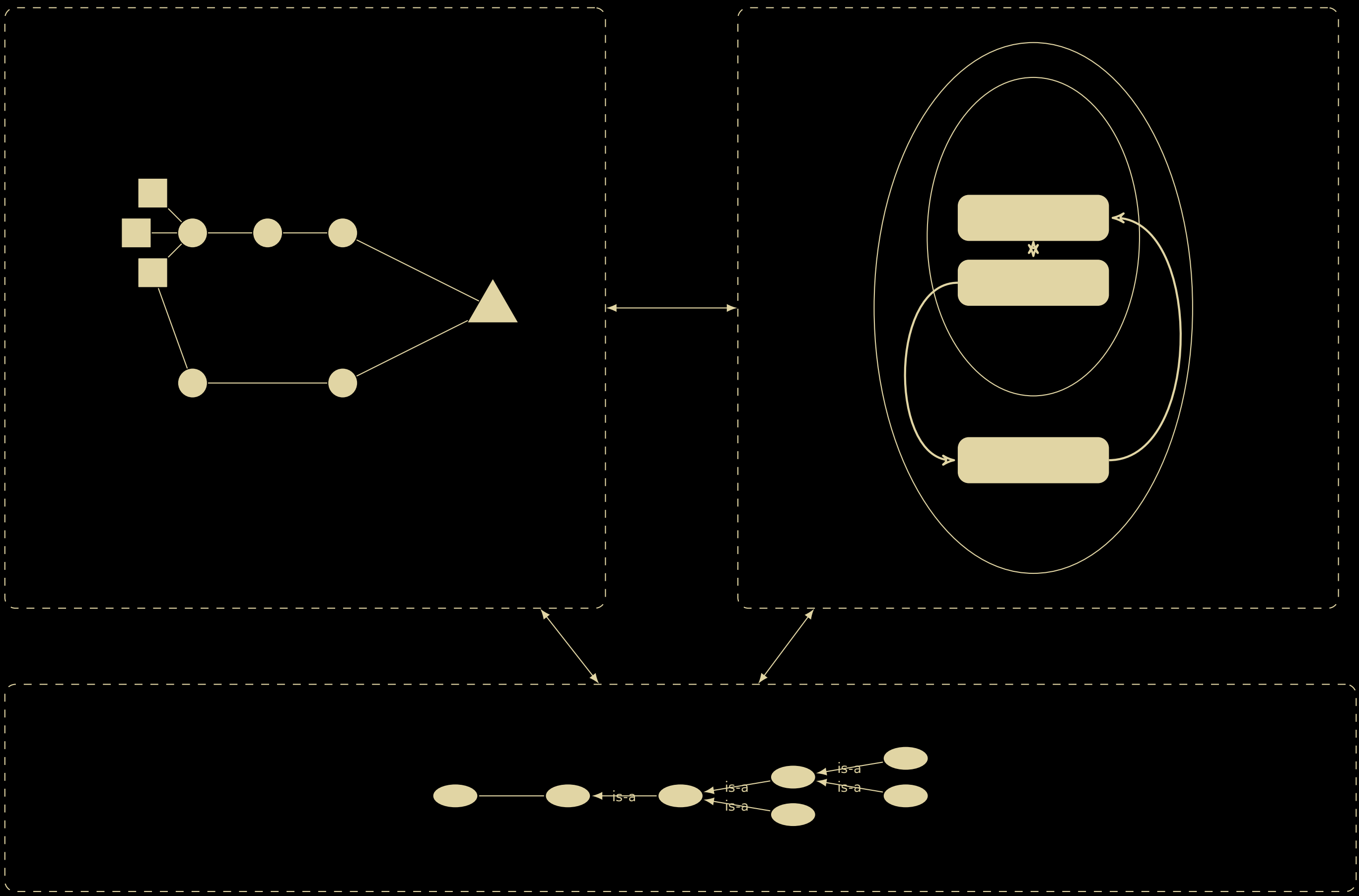


João Manuel Leitão Quintas

Context-based Human-Machine Interaction Framework for Artificial Social Companions

Tese de Doutoramento em Engenharia Electrotécnica e de Computadores, no ramo de especialização em Automação e Robótica,orientada por Doutor Paulo Jorge Carvalho Menezes e Doutor Jorge Manuel Miranda Dias e apresentada ao Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

June, 2018

· U C ·

Universidade de Coimbra

to Maria Isabel and Liliana

Abstract

This thesis addresses the problem of auto-adaptation of interfaces and interaction strategies in human-machine interaction by formulating an approach that looks into the complete combinations of components available to implement a functionality and considers context information at each moment to optimize the sequence of components (e.g. algorithms) that implement a given functionality. Our main purpose was to understand how agent's performance is affected when interaction workflows are incorporated in its information model and decision-making process.

In Human-Machine Interaction for Artificial Social Companions, we must incorporate features that allow an agent to be capable of delivering a sociable experience to the user. The associated technological challenges include active perception features, mobility in unstructured environments, understanding human actions, detect human behaviours and predict human intentions, access to large repositories of personal and social related data, adapt to changing context. These features are paramount for applications in the field of Active and Assisted Living (AAL), where the primary goal is to provide solutions that help people through ageing, by promoting active and healthy living.

The research question being addressed can be stated as: What approach can we follow to achieve adaptive interaction functionalities in artificial social companions?

To answer our research question, we conducted our study in four main phases including literature revision, building an Artificial Social Companion using state of the art development methodologies and technological components in high readiness levels, designing our framework proposing adjustments to the development of Artificial Social Companions, and concluding with integration and interoperability aspects.

Our hypothesis is that we can overcome limitations of current interaction functionalities by integrating contextual information to improve agent's performance when performing under very different conditions, to select most adequate algorithms to provide a given functionality, and to adapt interfaces and interaction patterns according to user intentions and emotional states.

To test our hypothesis, we started by assessing usability and user acceptance of our initial development of a Virtual Artificial Social Companion. This assessment was conducted together with end-users over a period of 12 weeks using validated methods and metrics like the Thinking-aloud protocol and the System Usability Scale, and showed a decreased acceptance of the system was mainly related with the lack

of flexibility and variety in interaction possibilities and the frequency of fails being higher than tolerable by end-users. After observing these results, we formulated our concept for designing a framework that captures the expected behaviour of the agent into descriptive scenarios, then translates these into the agent's information model and use the resulting representation in probabilistic planning and decision-making to control interaction. Our expectation was that adopting this framework could reduce errors and faults on agent's operation, resulting in an improved performance while interacting with the user.

To prove our concept, considering the needs and requirements typically reported by end-users of two different approaches of Artificial Social Companions, we designed and implemented a scenario for person detection features. In both systems, the user and agent could interact through multiple modalities, which included speech commands, gestures, and touch screen, but initially, they require to detect the person. Consequently, we described user's expectations in a Gherkin Scenario and translated it into our knowledge representation by means of an ontology. Following, we analyzed the performance of the selected algorithms for the person detection feature and modelled a decision process for planning and control the execution of the interaction taking into account the context of the operation. We adopted a Partially Observable Markov Decision Process to determine the interaction policy to be stored in the knowledge representation, which completes the description of the scenario.

The results, from our different experiments, confirmed that our approach can improve agent's performance, maintaining precision while improving specificity. Although, we consider that designing and implementing interaction workflows in artificial social companions is still challenging and it worth more research.

We believe this study will contribute to the field of Human-Machine Interaction, with specific application to Artificial Social Companions for Active and Assisted Living. It will help overcoming the limitations imposed by approaches that use pre-defined static models for agent's behaviour resulting in non-natural interaction, which will result in improving the usability of these systems.

Keywords: Human-Machine Interaction, Context, Active and Assisted Living, Artificial Social Companions, Adaptive Systems.

Resumo

Esta tese aborda o problema da auto-adaptação de interfaces e estratégias de interação na interação homem-máquina, formulando uma abordagem que analisa as combinações completas de componentes disponíveis para implementar uma funcionalidade e considera informações de contexto em cada momento para otimizar a seqüência de componentes. por exemplo, algoritmos) que implementam uma determinada funcionalidade. Nosso objetivo principal foi entender como o desempenho do agente é afetado quando os fluxos de trabalho de interação são incorporados em seu modelo de informação e processo de tomada de decisão.

Na Interação Homem-Máquina para Companheiros Sociais Artificiais, devemos incorporar recursos que permitam que um agente seja capaz de fornecer uma experiência sociável ao usuário. Os desafios tecnológicos associados incluem características de percepção ativa, mobilidade em ambientes não estruturados, compreensão de ações humanas, detecção de comportamentos humanos e previsão de intenções humanas, acesso a grandes repositórios de dados pessoais e sociais, adaptação a mudanças de contexto. Esses recursos são fundamentais para aplicações no campo da *Vida Ativa e Assistida* (AAL), onde o objetivo principal é fornecer soluções que ajudem as pessoas através do envelhecimento, promovendo uma vida ativa e saudável.

A questão que foi abordada pode ser enunciada como: Que abordagem podemos seguir para alcançar funcionalidades de interação adaptativa em companheiros sociais artificiais?

Para responder a esta questão, conduzimos o nosso estudo em quatro fases principais, incluindo revisão de literatura, construção de um Companheiro Social Artificial utilizando metodologias de desenvolvimento e componentes tecnológicos em níveis elevados de maturidade tecnológica, projetamos a nossa estrutura propondo ajustes para o desenvolvimento de Companheiros Sociais Artificiais e concluímos com aspectos de integração e interoperabilidade.

A nossa hipótese é que podemos superar as limitações das atuais funcionalidades para interação, integrando informações de contexto para melhorar o desempenho dos agentes a operarem em condições muito diferentes, selecionando os algoritmos mais adequados para fornecer uma determinada funcionalidade e adaptar a interação de acordo com as expectativas do utilizador.

Para testar a nossa hipótese, começámos por avaliar a usabilidade e a aceitação pelo utilizador de um Companheiro Social Artificial Virtual, desenvolvido inicialmente.

Esta avaliação foi realizada em conjunto com utilizadores finais durante um período de 12 semanas usando métodos e métricas validados, como o protocolo *Thinking-aloud* e a *System Usability Scale*, evidenciando uma aceitação reduzida do sistema devido a causas relacionadas principalmente com a falta de flexibilidade e variedade das possibilidades de interação e a frequência de falhas ser maior do que a tolerável pelos utilizadores. Depois de observar esses resultados, formulámos o nosso conceito para projetar uma estrutura que recolhe o comportamento esperado do agente em cenários descritivos, que posteriormente são incluídos no modelo de informação do agente que é utilizado no planeamento probabilístico e na tomada de decisões para controlar a interação. A nossa expectativa era que a adoção desta abordagem pudesse reduzir erros e falhas na operação do agente, resultando num melhor desempenho ao interagir com o utilizador.

Para provar o nosso conceito, considerando as necessidades e requisitos tipicamente mencionados pelos utilizadores finais de duas abordagens diferentes de Companheiros Sociais Artificiais, projetamos e implementamos um cenário para a funcionalidade de deteção de pessoas. Em ambos os sistemas, o utilizador e o agente podem interagir por meio de várias modalidades, incluindo comandos de fala, gestos e ecrã táctil, mas que inicialmente exigem a detecção da pessoa. Consequentemente, descrevemos as expectativas do utilizador num cenário *Gherkin* e traduzimos para a nossa representação de conhecimento através de uma ontologia. Seguidamente, analisamos o desempenho dos algoritmos selecionados para a funcionalidade de deteção de pessoas e modelamos um processo de decisão para planear e controlar a execução da interação, levando em conta o contexto da operação. Adotamos um Processo de Decisão de Markov Parcialmente Observável para determinar a política de interação a ser armazenada na representação de conhecimento, que completa a descrição do cenário.

Os resultados da nossa experiência confirmaram que a nossa abordagem pode melhorar o desempenho do agente, mantendo a precisão e melhorando a especificidade. Contudo, consideramos que projetar e implementar políticas de interação em companheiros sociais artificiais ainda é um desafio merecedor de mais investigação.

Acreditamos que este estudo contribuirá para o campo de Interação Homem-Máquina, específicamente na aplicação de Companheiros Sociais Artificiais no domínio da *Vida Ativa e Assistida*. A abordagem proposta poderá ajudar a superar as limitações impostas pela utilização de modelos estáticos, pré-definidos, para o comportamento do agente, resultando numa interação não natural; contribuindo assim para a melhoria da usabilidade desses sistemas.

Palavras-chave: Interação Homem-Máquina, Contexto, Vida Ativa e Assistida, Companheiros Sociais Artificiais, Sistemas Adaptativos.

Acknowledgment

The journey was long, sinuous and foggy. It would not be possible to travel this long without the guidance of helpful beacon lights spread out along the way.

First, a word of gratitude to my supervisors Professor Doutor Paulo Menezes and Professor Doutor Jorge Dias, for all the support, patience and push forward when most needed.

Second, to my family, for always believing in me and sacrificing some of our moments together because of this demand.

To my colleagues at Instituto Pedro Nunes - Laboratory of Automation and Systems (IPNlas), for supporting this objective; probably the persons who had to put up with all the joy and bad humor. This work could not be done without all of you. A much grateful thank you to all our team. Additionally, but not less important, to all the team members of the projects where I was involved in during these years as project manager at IPNlas.

Similarly, to my colleagues at University of Coimbra - Institute of Systems and Robotics (UC-ISR), for the inspiration and questions rise that helped me to think forward and sharing experiences.

Also, to my senseis and colleagues from Clube Karate Shukokai Coimbra, for all sweat, blood and pain that helped me to release the stress from a day, to endure psychologically and motivated self-discipline. Oss.

Finally, to the members of the jury, for the time and commitment on the evaluation of this thesis, as well as the anonymous reviewers of all scientific papers for their constructive opinions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Artificial Social Companions (ASC) are one type of artificial autonomous cognitive systems that integrate features aimed to engaging the user in a social-like experience (e.g. conversation, personal guidance or advice). These agents, often are designed to have personalised behaviours and can operate in a single or multiple person scenarios, usually giving assistance to people with diverse needs or preferences, where frequent and unpredictable changes on the context of operation may arise. Additionally, these systems typically integrate multimodal interaction providing both verbal and nonverbal (i.e. facial expressions, eye contact, gestures and posture, tone of voice) human-like dialogue enactment. This way, the desires and needs of end-users would be recognised and consequently addressed much more efficiently. The analysis of common ASC design aspects presented by Fong et. al. in [1] included cognition (planning, decision making), navigation, action, Human-Machine Interaction (HMI) (perception, environment sensing, interfacing with the end-user) and architecture development and middleware. These multitude of aspects in designing ASC result often in agents performing with a low rate of specificity, which have a negative effect on user acceptance. Given the complexity and lack of mature approaches addressing all the aforementioned aspects in an integrated way, Fong et. al. pointed out the need to extend on HMI capabilities in order to address issues imposed by social interaction. Although the study of social interaction is outside the scope of this thesis, the interested reader can find more detailed information on the studies produced by Severinson et. al. in [2] or by Goodrich et. al. in [3].

Attending to these initial considerations, we put two questions that will drive this thesis: *What if we could create ASCs that could auto-adapt their interfaces and interaction strategies to the surrounding context in order to meet user expectations and characteristics? Would this improve ASCs' usability, acceptance and perceived usefulness?*

1.1 Motivation

Our motivation to study the development of Artificial Social Companions (ASC) is two folded. On one hand, one particularly interesting domain of application, where such systems are slowly gaining acceptance as a mean to promote social stimulation, is Active and Assisted Living (AAL). In AAL, the primary goal is to provide solutions that help people through ageing, by promoting active and healthy living. Part of being active and healthy include socialising. In a vast number of cases, this activity is done in care centres or nursing homes. In such scenarios, the demand for social stimulation as part of care service aggravates the need for scarce qualified human resources. Thus, technological solutions are seen as a benefice, which allow human carers to concentrate in tasks more demanding or more specialised; but the challenge of adoption remains associated to user acceptance. On the other hand, ASC typically require the integration of several Human-Machine Interaction (HMI) features (e.g. multi-modal perception is often required to deliver a set of advanced features commonly identified as needs by the end-users of such applications), but fully integrated approaches are still not well matured to work properly in real world environments.

1.1.1 Relevance of Artificial Social Companions (ASC) for Health, Demographic Change and Wellbeing

To understand the relevance of Artificial Social Companions (ASC), first consider the current societal challenge related to the demographic changes happening in Europe. The European population of adults aged 65 and above is estimated to correspond to 20% in 2030 and expected to reach near 30% by 2060[1]. The increasing demand for healthcare and quality of life services to support the ageing population has inspired researchers worldwide to explore the applicability of new intelligent technologies to support older adults to cope with the challenges of ageing and live independently for longer periods of time. Artificial Social Companions (ASC) are a promising solution to the increasing challenges of eldercare. Lets consider the scenario of promoting autonomy at home, which represents a set of relevant interaction contexts satisfying the basic needs of the elderly (i.e. primary end-users). The associated strategic goals, in this type of service provision, are to support the elderly in daily life activities and to promote psycho-social and physical wellbeing. Hence, in this scenario it is safe to assume that the users' basic expectations are avoiding the feeling of loneliness, promoting safety and guidance through activities of the daily life. These expectations may be addressed by the features expected from virtual companions. On the other

[1]According to "The 2015 Ageing Report: Economic and budgetary projections for the 28 EU Member States (2013-2060)"

hand, carers, family and friends (i.e. secondary end-users) focus also on supporting the social aspects of elderly daily life. They often followup potentially emergent situations identified by the institution itself or by third parties, by assessing the elderly cognitive and mental states through screening and referral activities. To these type of end-users, key issues to be addressed include activities such as preparing meals, plating up and distributing lunch and snacks, collecting clothes for washing, drying, ironing and repairing, cleaning housing floor space. These activities require physical interaction with the environment and therefore may be addressed by the features expected from social robots. In these scenarios, it is relevant to integrate context information about a person within his/her home environment in the broadest sense possible. In the particular case of interacting with elderly people, it is important to identify Activities of Daily Life (ADLs) in order to provide correct services that can assist the user to attain his/her intended goals. Therefore, ASCs must be developed that are capable of 1) using a priori knowledge, either hard-coded or from experience, and 2) evolving in time.

1.1.2 Technological impact of Artificial Social Companions (ASC)

Considering the new wave of cyber-physical applications (e.g. including ASC), we observed the shift from low-level raw observation data and their direct/hardwired usage, data aggregation and fusion, to high-level formal context modeling, activity recognition and behavior analysis and change detection. It is envisioned that this trend will continue towards a further higher level of abstraction, achieving situation, activity and goal awareness to facilitate the construction of human-machine systems and human-system interaction. We can say that this tendency is in line with the trends of the last decade, where we observed an emerging focus on auto-adaptable and self-re-configuring ambient intelligence systems in order to support smarter habitats.

The technological challenges associated with the development of these re-configurable and auto-adaptive systems include active perception features, mobility in unstructured environments, understanding human actions, detect human behaviours and predict human intentions, access to large repositories of personal and social related data, adapt to changing context. In the case of social artificial agents, systems must incorporate features that allow an agent to be capable of delivering a sociable experience with the user.

Moreover, within the current trends in developing ASCs, we find the willingness to bring these systems out of laboratory conditions (i.e. achieving a higher Technology Readiness Level (TRL)). That means, these systems must be adaptable to a high variety of unconstrained environments. However, the typical way we interact with ASCs, up until this point in time, always involves some form of direct or conscious command, in order for it to perform a specific function for us. On the other hand, if we look at the communication between people, it is actually a lot more interesting and much more informative because humans take into account far more than what is explicitly expressed. We take into account contextual information that is implicitly perceived and processed at the sub-conscient level; thus, reducing our mental workload by means of more automatic processes in memory and reasoning. This information is particularly useful to help us intuit feelings and emotions, which lead to adapting our behaviours in empathetic ways. This degree of sophisticated features are still lacking in our communication with ASCs. The challenge is for the next generation of human-machine systems to evolve beyond the imposed limits by explicit interaction towards being capable of perceiving context as humans do and learn from shared experiences.

1.2 Problem

The problems addressed in our study are tightly related with the way Interaction Design aspects are considered when implementing Artificial Social Companions (ASC).

Addressing the way Interaction is implemented in these systems is particularly relevant, as implementing real-world systems requires handlers to design particular solutions from generic approaches resulting in static architectures. In these architectures, we can assume that the scalability and redundancy of some functionalities are already considered (e.g., security, fault tolerance, data storage). However, robust approaches for human-machine interaction are still a challenging topic. Taking into account some examples from recent works [4–9], we can identify key factors that prevent these systems from being ideal in terms of interaction. These factors, among others, are related with uncertainty associated with noisy inputs, variation in environment conditions, and unclear expectations from the user. In spite of the sophistication of state-of-the-art systems, we could not find a holistic approach to address human-machine interaction. Typically, each functionality is addressed individually and later integrated overall through well-defined and fixed interface protocols. However, the lack of redundancy and fallback strategies in terms of interaction functionalities often results in unexpected system behaviors (e.g., faults, errors or failures) creating barriers of adoption to new technologies or new interaction modalities.

Therefore, this thesis focus on problems of how to integrate the understanding of user needs in technology development and how this can result on building context-aware ASC.

1.2.1 Understanding user needs to improve technology

The challenge of user acceptance is of vital importance for future systems and still one of the major reasons for reluctance to deploy or introduce artificial social companions in AAL applications. It is commonly accepted that the misunderstanding of functionalities and handling of new technology often leads to rejection or even to fear by users, if the system is not behaving as expected.

Analysing the results from relevant projects that were developing ASC systems (i.e. CaMeLi[2], SocialRobot[3], GrowMeUp[4] were closely followed during this thesis), we identified common needs, which are repeatedly requested by end-users. In summary, users expect intuitive interaction with social agents. Therefore, natural interaction features (e.g. gestures, speech, etc.) and adaptation to the user's general profile, specific needs and intentions, were identified as high priority requirements. In the AAL domain, these features must address the needs of users with cognitive or physical skills degradation (e.g. elderly or impaired people) and aim to compensate these limitations, hence enriching user experience and accessibility. A list of most wanted HMI related functionalities in this domain can be summarized as follows:

- Agent should take up the role of a personal assistant, with a friendly and informal personality (a friendly and informal companion for the elderly);
- The agent should talk to elderly and provide them company in a natural dialogue with a broad vocabulary;
- The agent should guide the elderly to decide what they should do by giving them advice. Examples:
  - Oversee and assist with daily activities by providing guidance instructions;
  - Helps to manage and organize daily living activities;
  - Provide hygienic and basic household advising and personal support and also motivate to do there chars;
  - Stimulate and motivate the elderly to do physical exercises;
  - Monitor and guide the elderly while undergoing their physical exercises.
- The agent should draw the attention of the elderly by addressing him/her by his/her first name;

[2] `http://www.cameli.eu/` (accessed:31/07/2017)
[3] `http://mrl.isr.uc.pt/projects/socialrobot/` (accessed:31/07/2017)
[4] `http://www.growmeup.eu` (accessed:31/07/2017)

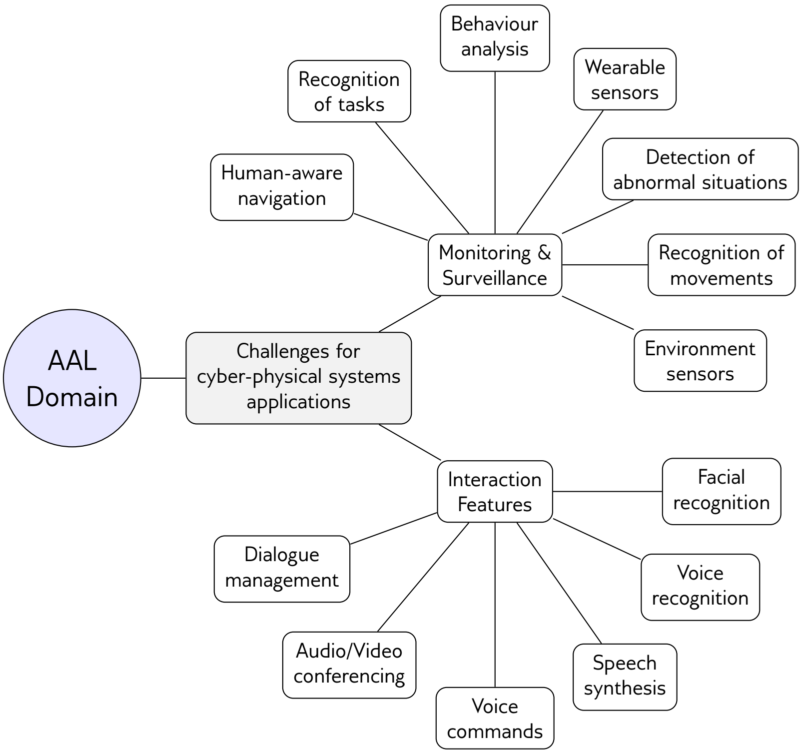


Figure 1.1: Scientific and technological challenges related with HMI related features, as summarized from end-user needs

- The elderly should be able to attract the attention of the agent using voice commands (thus if the elderly calls the agent it should recognize the elderly);
- The agent should be able to move around in the house and recognize familiar faces;
- Elderly want to be in control of the activity level of the agent, most of them chose the answer “Switch on-off as I choose”.

Taking into account this list, we conclude that interaction aspects are a central point for end-users. Therefore, we summarized the mostly requested features in figure 1.1, which were classified as Monitoring & Surveillance and Interaction Features.

1.2.2 Building context-aware interactive agents

Since Bower and Miller in [10–14] studied behavior modification on animals, several researchers from the field of cognitive psychology in [15–21] have been studying how memory, language comprehension, emotion and reasoning processes develop in humans. Barsalou in [22] demonstrated the existence of context-independent and context-dependent properties in concepts (i.e. mental representations that correspond to some distinct entity or classes of entities, its essential features) used for problem solving, metaphor and sentence comprehension. Mellers et.al. in [23] proposed that context has also impact on decision-making and action, and preference measurements. Also, researchers as Thibodeau and Boroditsky in [24–27], are refining the principle of linguistic relativity (also known as Sapir–Whorf hypothesis or Whorfianism) that claims language affects its speakers' reasoning process. In summary, it is commonly accepted that behaviours (i.e. from humans and other animals) are context-dependent and that context influences almost all aspects of behaviour mostly in an automatic manner (i.e. without a conscious reasoning effort). For example, assuming that human perception is heavily influenced by top-down predictions, it may be more difficult to detect, or recognize, out-of-context objects than familiar ones.

Therefore, ASCs should be able to cooperate and understand with users, but it requires sharing, at some extent, the same language and semantics, and the same "understanding of the world". Thus, sharing and learning context information becomes a relevant issue to allow an ASC adapting its services to the user's needs and desires, and improve their interaction patterns. To refine reasoning and to integrate context in order to provide correct services according user's expectations, we must develop models that are capable 1) of using a priori knowledge, either hard-coded or from experience, and 2) of evolving in time. However, we could not find available yet any satisfactory framework to reason and learn based in context information in ASC systems. For example, we attempted to address part of this problem in [28, 29].

Human-Machine Interaction (HMI) workflows typically have been modeled as deterministic processes, where the user is considered to be following fixed patterns for interaction. However, this puts limitations on interactive systems, as they are implemented as command driven approaches, which will fail under unforeseen situations. For example, a user tries to access a functionality using an unknown voice command; the agent will not execute the expected functionality and does not have any fallback plan to notify the user that it did not understand his expectation. Another example, the agent is waiting to interact with the user; the environment is very quiet; the agent adjusts the gains for the microphones to perceive sounds; suddenly pink noise[5] is perceived as a command and the agent executes a random functionality, which confuses the user, who was relaxing and silently reading a book in his living room. Hence, these types of approaches are not capable to adapt to the level of uncertainty in real world settings.

1.3 Summary overview of related works

Human-Machine Interaction technologies and corresponding cognitive capabilities of artificial agents have seen many developments in the last few decades. For example, as presented by Ferreira et al. in [30], solutions for multi-sensory active perception and attention allocation have greatly evolved; also, as described by Prado et al. in [31], similar developments happened on multimodal human emotion and dialogue analysis and human-like emotion and dialogue synthesis; moreover, also human behaviour analysis approaches were matured, as for example in [4]. On the other hand, Goodrich in [3] brought attention to the lack of strategies for dealing with acceptability and safety issues in autonomous agents operating in a human environment. These facts restrict most of the current socially interactive agents to highly controlled environments and specialised applications. Moreover, Sili in [32] summarised the state of the art for interaction models, which typically refer some degree of adaptation but require explicitly models to rule out the behaviour of the agent.

Furthermore, we can find in the literature, recent works that address adaptation processes involved in HMI, but they mainly focus on task planning. In [33] the authors formalized a general intermediate layer approach, which allowed automatic generation of property-enforcing layers to be used between an application program and a set of resources for which safety properties are defined and should be respected by the global system (i.e. the application, plus the intermediate layer, plus the set of resources). In [34] the authors focused on the organization aspects of the agent decisional abilities and on the management of human interaction as an integral part of the agent control architecture. Their proposed framework allowed the agent to accomplish its tasks and produce behaviors that support its engagement vis-a-vis its human partner and interpret similar behaviors from him. The framework was applied in a companion agent scenario in [35] within the scope of the Cogniron project. In [36] presented the agent control architecture SHARY, dedicated to agent action in presence or in interaction with humans. This architecture focused more in task planning but provided support to implement a supervision system adapted to HMI. Nevertheless, this approach is relevant still for state of the art implementation, as

[5] random noise having equal energy per octave, and so having more low-frequency components than white noise.

described by Devin et al. in [37] and by Lemaignan et al. in [38].

Attending to the state-of-the-art, we conclude that proposed approaches disregard aspects related with the dynamics of task execution. This means, after task planning is concluded the system may have different paths to choose from, but the decision process associated with the control of task execution is limited to a set of prior rules, which may or may not correspond to the optimal action to take for a specific situation. Moreover, typical approaches tend to design and implement algorithms that take into account several environment features (e.g. light, pose, etc.) to adapt its performance obtaining accurate results. An advantage of these approaches is to concentrate complexity in one algorithm leading to simple system architectures. On the other hand, a disadvantage of such approaches is their limitation to adapt to conditions under different scenarios, which typically requires manual adjustments to compensate changes of environment features and do not allow to incorporate additional capabilities (i.e. add new algorithm to operate in new conditions).

Additionally, based on the information from a survey that we conducted about the key elements of Human-Machine joint action, like in [39] and [40], we conclude that we commonly find approaches that treat context and interaction models separately resulting in monolithic architectures for components related with interaction functionalities. This aspect could be improved by exploring what approaches can successfully integrate both models and how they could be implemented. Furthermore, the most accepted definitions for context are those proposed by Schilit [41], Pascoe [42] and Dey [43], which can be summarized into a general notion that *Context* is defined as all information that characterizes a situation. The problem with this definition is that it is difficult to understand what information characterizes a situation. Moreover, the typical examples used to refer to context result as description for a place or an event (e.g., in the context of Kitchen, in the context of Meeting). This makes it hard to generalize the definition to other applications, for which their information domain was not previously formalized, modeled and represented.

Regarding user-adaptiveness, we have found, essentially, three types of user-adaptive ASC systems: systems that adapt without explicit knowledge about the user, systems that keep a static user model and systems that keep a dynamic user model. The works presented in [44] and [45] do not maintain an explicit model of the user. Instead, these systems achieve the user-adaptiveness as a collateral effect of their main goal. In fact, the system of [44] adapts to the user by monitoring accessible areas for vacuuming, and that of [45] adapts to the user by estimating their intention in the cooperative task of selecting ingredients for a recipe. Static user models, such as those in [46–48] can also be used for adaptation. These systems make use of immutable information on the user, such as their persona [46, 49], personality [47] and physical capabilities [48], to generate adapted behavior. The unchanging nature of the user models employed do not allow for systems to gain information on the user from direct interaction and hinder interaction to naturally fleeting characteristics of the user, such as their mood. Despite the lack of dynamism in the user model, these systems are very successful at adapting to these wider, unchanging traits of the user and achieve interesting results in their specific applications. Dynamic user models, such as those found in [50–52], can be used to adapt the system's behavior to the user's dynamic characteristics, thus achieving higher levels of adaptivity and potential interaction quality. The dynamic nature of the user model allows the system to learn from the user *in loco* while the interaction is taking place. Systems of this nature have been applied to strict Human-Robot Interaction (HRI), such as in [51], or in roboticized versions of classical Human-Computer Interaction (HCI) problems, such as learning assistance for children [50] and in robotic recommender systems [53].

In a similar manner, other authors developed approaches dedicated to generating task-oriented interactions of service robots or attempted to modeling the duration of the user interest during interaction with an artificial agent. Kim and Yoon, in [54], defended that "to obtain appropriate human aid for conducting tasks, a robot should be capable of generating meaningful questions regarding the task procedures in real time and applying the results to modify its task plans or behaviors". They concluded that few studies addressed the integration of robot task management and HRI in high-level task planning. For that purpose, they proposed a script-based scheme for task planning and HRI that supported the planning and is generated by it. Zhang et al., in [55], proposed a hidden semi-Markov model to track the change of users' interests. They were motivated by the observation that "users' preferences often change over time" but "most existing approaches that address time information remain primitive", thereby justifying their use of a probabilistic approach. Another application example, by Cheng et al. in [56], proposed a semantic web-based context ontological reasoning service for multimedia conferencing process management that automatically selected the appropriate means of notifications based on the conference time and the participant contact details. This last example demonstrates the relevance of research on context-based interaction approaches for improving automatic intelligent systems.

1.4 Open challenges, Aims and Expected Impact

Recently developed ASC systems providing assistance in domestic, professional and public environments are based on closed architectures. Thus, they are limited to operate in specific environment settings, equipment and data. In spite of modern implementations incorporate sophisticated perception models and sensors, most approaches neglect context during the perception process. For example, Jong-yi Hong et al. in [57] concluded that, despite research in context-aware systems increased between 2004 and 2007, most works addressed conceptual and research aspects. More recently, researchers are focusing efforts in the implementation of real-world applications.

1.4.1 Open challenges

Considering the above, there are still a set of open challenges that must be addressed by the researchers community in order to fully implement real life context-aware systems. In general terms, the scope of existing applications is limited to small regions and technologies related to context-aware systems are not standardized. Moreover, architecture models, context modelling and algorithms are customized for each problem, thus it is difficult to evaluate the performance between different systems. However, context awareness has been identified as a key feature for new applications in cyber-physical systems (e.g. including ASC), with particular interest for HMI development. Some of the open challenges identified by Murphy et al. in [58] and Hong et al. in [57] are related with:

- extracting the cognitive context from physical context, where user preferences, tasks and emotional state of the user are inferred from sensor data;
- evaluating which algorithms perform best specific replicable problems;
- performing more flexible dissemination of information within the decision-making hierarchy, instead of the classical one-way information flow (upwards in the hierarchy);
- selectively propagating information to distributed users with diverse needs, based on context sharing and information utility assessment;
- dealing with concurrence in large datasets of information with heterogeneous formats;
- resolving context conflicts when uncertain data is collected and used to infer contextual information.

Associated to these challenges, current systems use static and implicit representations for context, what prevents dynamic system adaptation through information sharing and learning. This problem is being partially addressed with ontological approaches to represent context. In spite of these efforts, context models are still lacking a formalism that promotes an objective representation of information and that allow the application of generalized algorithms. These claims are supported in the literature, according to Brdiczka et al in [59, 60] "... *computerized spaces and their devices require situational information, to respond correctly to human activity. In order to become context aware, computer systems must thus maintain a model describing the environment, its occupants, and their activities.* ... ". The typical approach to provide contextual information to the application is by manually defining the context models according to their end goal and taking into account particular user needs. This approach does not traduce the "real world" in the sense that it fails if the user needs evolution over time is taken into account. Furthermore, the same authors considered that "... *New activities and scenarios emerge in a smart environment, and others disappear. New services must be integrated into the environment, whereas obsolete services should be deleted. Thus, a fixed context model is not sufficient.* ...". Moreover, long-term maintenance, required by common approaches, have a negative economical impact to the user, i.e. having an expert periodically adjusting the system according user needs would be expensive. Thus, the research for more intelligent, self-learning and self-adaptable systems is justified facing the inefficiency of the common approaches.

1.4.2 Aims

The purpose of this thesis is to understand how agent's performance is affected when interaction workflows are incorporated in its information model and decision-making process. The research question being addressed can be stated as: *What approach can we follow to achieve adaptive interaction functionalities in artificial social companions?*

We believe that part of the solution to this problem is to incorporate redundancy and fallback strategies in terms of interaction functionalities that result in the agent's self-adaptation to its context (e.g. user model and environment conditions).

Our hypothesis is that, ASCs that integrate redundant algorithms, which can provide a functionality under different contexts, and represent this information on the knowledge model that is used by the decision process in charge of planning and action selection will operate with improved specificity.

In this hypothesis, we assume that any architecture can be described as a network topology of algorithms and we can find redundant sequences of algorithms (i.e., paths in the network) that implement the same functionality. In these conditions, auto-adaptation will be facilitated by introducing a decision process that considers the context at a given moment to select the best sequence of algorithms, in which requirements are satisfied by the current context.

To test this hypothesis, we are proposing a framework that captures the expected behaviour of the agent into descriptive scenarios, then translates these into the agent's information model and use the resulting representation in probabilistic planning and decision-making to control interaction.

Our expectation is that adopting this framework can not only reduce errors and faults but also extend agent's operation by providing automated adaptation to variable conditions, resulting in an improved performance while interacting with the user, which may result on improving ASC acceptance by the users.

1.4.3 Expected Impact

The proposing framework will help overcoming the limitations imposed by approaches that use pre-defined static models for agent's behaviour resulting in non-natural interaction, which will result in improving the usability of these systems. Thus, it will contribute to the advances on the field of HMI, with specific application to Artificial Social Companions for AAL. The foreseen impact of this study is to progress beyond the state of the art in two of the aforementioned open challenges. Particularly, regarding:

- evaluating which algorithms perform best specific replicable problems;
- achieve a more flexible dissemination of information within the decision-making hierarchy, instead of the classical one-way information flow (upwards in the hierarchy).

1.5 Research methodology

To answer our research question, we decided to setup our study in four main phases, like illustrated in Figure 1.2. First, we started with an extensive literature review, from where we drawn some conclusions allowing us to understand better our research field and identify more concretely the implication of our research. Second, we built an Artificial Social Companion, in the format of a virtual agent, following the state of the art development methodology and adopting technological components with high technology readiness levels. This phase, served us as our baseline study, where we could observe first hand the end-user's reaction to the introduction of ASC's in their daily lives. After observing the results from this second period, we initiate the third phase of our research, which consisted in designing a framework proposing some adjustments to the development methodology and implementation technical approach of ASC's. To complete our research, in the fourth phase, we looked for ways to manage integration and interoperability aspects with other systems.

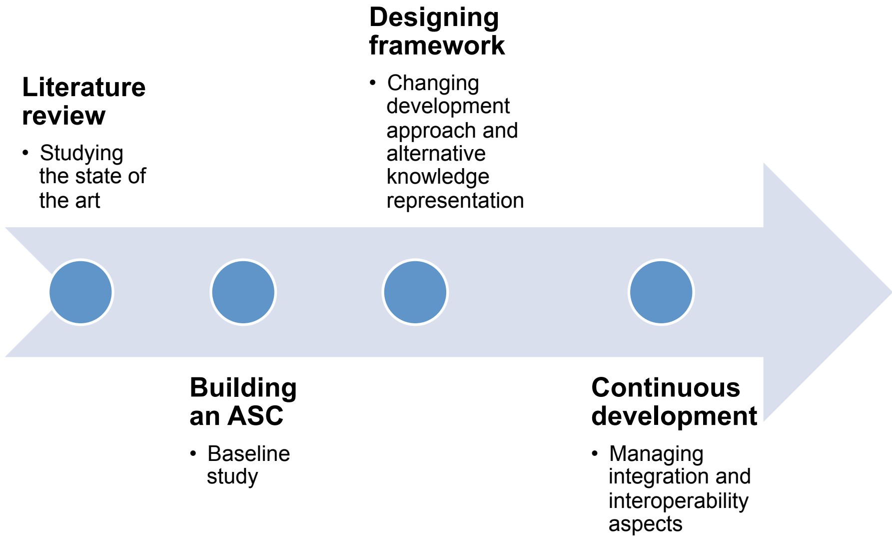


Figure 1.2: Research methodology.

1.6 Thesis outline

The next chapters are organized as follow:

- Chapter 2 introduces the foundations of Interaction Design that are relevant to understand the outcomes of our work. Its main objective is to present the major concerns addressed in developing interactive systems.
- Chapter 3 covers the state of the art related with developing ASCs. This chapter presents a detailed overview on what are ASCs and how they are typically designed and implemented.
- Chapter 4 present the implementation and evaluation of a Virtual ASC, which served as baseline for our study.
- Chapter 5 corresponds to the major contribution of this thesis. It will present the knowledge model and corresponding representation. Mainly, it covers the steps associated with the implementation of the proposed framework, which are guidelines for applying the framework in other relevant scenarios. Additionally, we present an experiment to validate the application of the framework to improve video based person detection in a robotic ASC, which is one functionality typically used when interacting with the user.

- Chapter 6 addresses an experiment that focus on interoperability aspects of sharing knowledge between different ASCs.
- Chapter 7 concludes this study summarizing the most relevant results, conclusions and sets some possible directions for future work.
- In the Annexes we provide technical complimentary information.

Chapter 2

Foundations of Interaction Design

This chapter introduces the foundations of Interaction Design that are relevant to understand the outcomes of our work. Its main objective is to present the major concerns addressed in developing interactive systems. We will start by covering the conceptual understanding of Interaction Design. Following, we will summarize the two main aspects addressed by this topic, specifically what regards to understanding the user and the contexts in which interaction takes place.

2.1 Conceptualizing Interaction

The main concerns for Interaction Design revolve around *understanding the user* and the *context* (e.g. types of activities the user is doing) when interacting with objects, devices or systems. In summary, it is all about optimizing the users' interaction with a system, environment or product considering what people are good and bad at, what might help people with the way they currently do things, what might provide quality user experiences, what people want and getting them involved in the design, using "tried and tested" user-based techniques during the design process.

The principles and methodology proposed in Interaction Design, as described thoroughly by Sharp, Preece and Rogers in [61], allow us to establish, for each system, a conceptual model for interaction that describes the proposed system in terms of a set of ideas and concepts about its intended behaviour, including actions and appearance, from the users' perspective (i.e based on users' needs and other requirements). Consequently, it requires doing iterative testing of the system as it is developed to ensure that it is designed and implemented according users' expectation (i.e. if it is understandable in the intended way).

A key aspect of this design process is initially to decide what the users will be doing when carrying out their tasks. One category of conceptual models is based on activities being carried out by the user. The most common types of activities that users are likely to be engaged in when interacting with systems are instructing, conversing, manipulating and navigating, exploring and browsing. Another category of conceptual models, is based on an object or artifact the user is interacting with, such as a tool, a book, or a vehicle. These tend to be more specific than conceptual models based on activities, focusing on the way a particular object is used in a particular context. They are often based on an analogy with something in the physical world.

Also, interaction design cannot be disassociated from usability principles. These, can be listed as the ten usability heuristics developed by Nielsen et al. in [62–64] to guide designing interactive systems:

1. Visibility of system status: always keep users informed about what is going on, through providing appropriate feedback within reasonable time.
2. Match between system and the real world: speak the users' language, using words, phrases and concepts familiar to the user, rather than system oriented terms.
3. User control and freedom: provide ways of allowing users to easily escape from places they unexpectedly find themselves, by using clearly marked 'emergency exits'.
4. Consistency and standards: avoid making users wonder whether different words, situations, or actions mean the same thing.
5. Help users recognize, diagnose, and recover from errors: use plain language to describe the nature of the problem and suggest a way of solving it.
6. Error prevention: where possible prevent errors occurring in the first place.
7. Recognition rather than recall: make objects, actions, and options visible.
8. Flexibility and efficiency of use: provide accelerators that are invisible to novice users, but allow more experienced users to carry out tasks more quickly.
9. Aesthetic and minimalist design: avoid using information that is irrelevant or rarely needed.
10. Help and documentation: provide information that can be easily searched and provides help in a set of concrete steps that can easily be followed".

2.2 Understanding Users

We focus now on the characteristics of users that must be to taken into account when designing interactive systems.

Since mid 90's, motivated by the massive adoption of personal computers, and more recently in the 00's, with the advent of mobile devices, it became increasingly important to focus the design of interaction on the user.

> "The most important thing to design is the user's conceptual model. Everything else should be subordinated to making that model clear, obvious, and substantial. That is almost exactly the opposite of how most software is designed." (David Liddle in [65])

In spite of considering the eventual physical limitations of the user, interactive systems demand mostly the users' cognitive processes. The specific kinds of processes that have been described as cognitive processes include attention, perception and recognition, memory, reasoning, problem solving, planning and decision making, learning, and language processing (i.e reading, speaking and listening).

2.2.1 Attention

Understanding the user starts by apprehending how he concentrates on objects and/ or tasks. As introduced in [61], *Attention* is the process that allows us to focus our senses on gathering information that is relevant to what we are doing. An example of this capacity is the cocktail party effect, which is the phenomenon of being able to focus one's auditory attention on a particular stimulus while filtering out a range of other stimuli, as when a party-goer can focus on a single conversation in a noisy room. The extent to which this process is easy or difficult depends on whether we have clear goals and whether the information we need is salient in the environment. If we know exactly what we want to find out, we try to match this with the information that is available. When we are not sure exactly what we are looking for we may browse through information, allowing it to guide our attention to interesting or salient items. On the other hand, the way information is displayed (information presentation) can also greatly influence how easy or difficult it is to attend to appropriate pieces of information.

2.2.2 Perception

Understanding user's perception capabilities is an important aspect in interaction design, since it is important to present information in a way that can be readily perceived as expected. For example, considering the design of graphical user interfaces there are many ways to design icons. The key is to make them easily distinguishable from one another and to make it simple to recognise what they are intended to represent. More importantly, in multimodal interaction (i.e. when combining different media to interact with the user) the orchestration between different modalities need to be designed accounting users capability to recognise the composite information represented in them in the way intended. For example, the design of lip-synch applications, where the animation of an agent's face to make it appear to be talking, must be carefully synchronized with the speech that is emitted. In this case, the use of sound and animation together needs to be coordinated so they happen in a logical sequence; a slight delay between the two can make it difficult and disturbing to perceive what is happening.

Perception is a complex process, involving other cognitive processes such as memory, attention, and language makes it highly dependent on context. It refers to how information is acquired from the environment by the senses and transformed into experiences. Dix et al. described in [66] that for the majority of individuals, vision is the most dominant sense followed by hearing and touch. The following sections summarize the most relevant information provided in [66].

Vision

Vision may be considered the primary source of information for perceiving the environment. It is the result of processing and interpreting bio-signals generated by the body's physical receptors (i.e. the eyes) of the stimulus from the outside world.

The physical reception of the stimulus is related with the anatomic characteristics of the eye and the visual system, which act as a natural filter of the useful information capable to be processed by the brain. Despite the limitation associated with the physical formation of images in the eyes, the processing and interpretation of such stimulus allow images to be constructed from incomplete information in the brain. Therefore, it is important to understand how human vision works and what particularities should influence the Interaction Design.

Vision involves to recognize patterns, differentiate colours and disambiguate relative distances. It is crucial, when designing visual interfaces, to understand how size and depth, and brightness and colour, are perceived, in order to obtain effective designs. To that end, first it is needed to consider how the image appears on

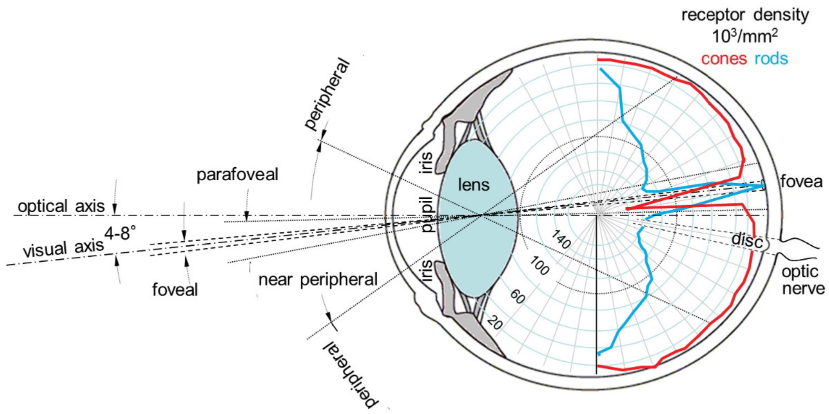


Figure 2.1: Visual angles for an object in the horizon.

the retina. Like in a photographic camera, the reflected light from objects forms an inverted image on the retina, where the corresponding size is related with the visual angle (i.e. the size of the image increases as visual angle increases). This can be illustrated by Figure 2.1[6].

The object's size and its distance to the eye are the two variables that influence the visual angle. Meaning that, if two objects are at the same distance, the larger one will have the larger visual angle. Similarly, if two objects of the same size are placed at different distances from the eye, the furthest one will have the smaller visual angle. Although, the law of size constancy indicates that our perception of size relies on factors other than visual angle. For example, our perception of depth allows us to perceive, when objects overlap, that the object which is partially occluded is further away (i.e. in the background). Other examples include the familiarity with objects, and relative disposition to each other (e.g. if the object is expected to be of a certain size; or when their size and height in our field of view provides a cue to its distance).

Another important factor in visual perception is the perception of brightness, which is a subjective reaction to levels of light affected by the amount of light emitted by an object (i.e. luminance). It is relevant to consider that objects in dim light are not so easily seen when fixated upon, but are rather more visible in peripheral vision, because rods are superimposed to cones. On the other hand, in normal lighting, as cones superimpose to rods, visual acuity is increased. Therefore, for example, using high display luminance is expected to facilitate visual perception of graphical user interfaces.

Finally, the perception of colour is usually regarded as fundamental for discretize objects in the environment. Colour may be described in terms of Hue (i.e. the spectral

[6]https://www.semanticscholar.org/

wavelength of the light), Intensity (i.e. the brightness of the colour) and Saturation (i.e. the amount of whiteness in the colour). By varying Intensity and Saturation, humans perceive in a range of 7 million distinct colours. Because of a higher concentration of cones in the fovea, this is the region of the eye where colour is best perceived. An important characteristic, limiting color acuity in humans, is the fact that only 3 to 4% of cones, in the fovea, are sensitive to blue light radiation wavelengths, resulting in lower acuity to color blue. Also, worth mentioning the fact that colour blindness affect between 1 to 8% of humans, with more prevalence in males. People affected by this condition, is most commonly unable to discriminate between red and green colours.

As mentioned before, visual processing has the ability to interpret and exploit expectations to resolve ambiguities from image formation. For example, consider the image in Figure 2.2a that is present inside Figures 2.2b and 2.2c. The context in which the object appears allows our expectations to clearly disambiguate the interpretation of the object, as either a B or a 13.

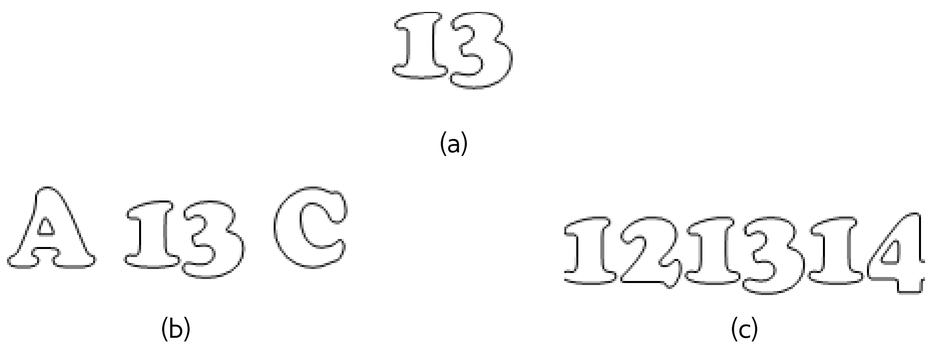


Figure 2.2: Examples of visual ambiguous shapes. Ambiguous shape in 2.2a appears to be a letter when in sequence of letters ABC in 2.2b but appears to be a number when in sequence of numbers 12 13 14 in 2.2c.

Despite this amazing skill, it can also create optical illusions, as the case in the example illustrated in Figure 2.3. This typical optical illusion make the line on the left appear longer than the line on the right, but in fact, the two lines are the same length. This example corresponds to a false application of the law of size constancy; in this case the line on the right appears like a convex edge, the line on the left appears like a concave edge. The former therefore seems closer than the latter and is therefore scaled to appear shorter.

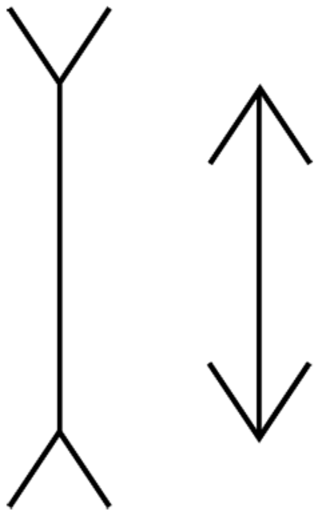


Figure 2.3: The Muller-Lyer illusion

Hearing

Hearing is often considered secondary to sight, but we tend to underestimate the amount of information that we receive through the ears. Through sound an individual not only is able to identify the source but also can identify its distance and its direction, extrapolating a close approximation of its position. These characteristics are of great importance in sensing the one's environment.

Similar to vision processing, hearing also is limited to a range of sound frequencies, which are useful for human perception. Hence, our ears and nervous system act as a filter for the vibrations in air pressure.

Moreover, sound characteristics that affect perception and consequently the ability to process sound are *pitch* (i.e. the frequency of the sound; human can hear frequencies in a range between 20 Hz to 15 kHz), *loudness* (i.e. the amplitude of the sound) and *timbre* (i.e. the type of the sound, related to the source producing it).

Another particularity that humans possess, is the capability of identify a sound's location. This capability is thanks to the binaural structure of human anatomic auditory system (i.e. the two ears receive slightly different sounds, owing to the time difference between the sound reaching the two ears and the reduction in intensity caused by the sound waves reflecting from the head).

In Interaction Design, sound is typically used in a very limited way, commonly used to provide warning sounds and notifications. Some exceptions occur for special interfaces for visually impaired people or other multimedia applications, where voice feedback and sound effects are very present.

Touch

Touch will be the third and last human perception capability we will address (i.e. taste and smell will not be covered given their reduced role in current approaches for implementing interaction modalities).

Touch provides a wide range of information about the physical environment. In the cases of vision and hearing, we were referring to anatomical structures capable of traducing electromagnetic waves into electro-physiological signals. In the case of touch, we are referring to anatomical structures capable of sensing texture and temperature of physical objects.

Touch is possible thanks to a specialized organ that contains three types of sensory receptors. This organ is the *skin*. Skin's thermoreceptors respond to heat and cold, nociceptors respond to intense pressure, heat and pain, and mechanoreceptors respond to pressure.

In Interaction Design, the sense of touch is of a major importance for those whose other senses are impaired, for example, for visual impaired people, interfaces such as Braille are an important source of information in the interaction.

2.2.3 Memory

Understanding how user's recall various kinds of knowledge and solve problems are two characteristics that cannot be ignored when designing an interactive system. These are important aspects to take into consideration in interaction design, since they allow us to perform everyday activities, recognize someone's face, remember someone's name, recall when we last met them and know what we said to them last. It gives us our sense of identity, by preserving knowledge acquired from our past experiences.

It is also important to be aware of the associated limitations. Some well-known memory phenomena are that people are much better at recognizing things than recalling things, and that certain kinds of information are easier to recognize than others. In particular, people are very good at recognizing thousands of pictures, even if they have only seen them briefly before.

On the other hand, it is not possible to remember everything that we perceived, nor it is desirable, as our brains would get completely overloaded. Therefore, our brain is capable to filter what information need to be memorized. This process is tightly connected with attention. The more attention that is paid to something and the more it is processed in terms of thinking about it and comparing it with other knowledge, the more likely it is to be remembered. For example, when learning about a topic it is much better to reflect upon it, carry out exercises, have discussions with others about it, and write notes than just passively read a book or watch a video about it. Additionally, another factor that affects the extent to which information can be subsequently retrieved is the context in which it is encoded. One outcome is that sometimes it can be difficult for people to recall information that was encoded in a different context from the one they currently are in. Consider the following scenario: You are visiting a new place and someone comes up to you and says hello. You don't recognize him for a few moments but then you realize he is one of your neighbors. You are only used to seeing your neighbor in the hallway of your apartment block and seeing him out of context makes it difficult to recognize initially.

Conceptually, still remaining as the most popular model for studying memory is Atkinson-Shiffrin model [67], after Richard Atkinson and Richard Shiffrin who developed it in 1968. It divides memory into three main categories: sensory memory, short-term memory or working memory, and long-term memory. However, it is not clear if these are separated systems or different functions of the same system. In any case, this model serves us to understand the basic arrangement and processes of memory also described in [66], as follows.

Sensory Memory As the name suggests, sensory memory is the memory structure associated with sensing capabilities. For the specific cases of the perception modalities previously addressed we have iconic memory for visual stimuli, echoic memory for aural stimuli and haptic memory for touch. These memories are constantly overwritten by new information. Information is passed from sensory memory to short-term memory by attention. Attention is the concentration of the mind on one particular stimulus when mixed or interfered by a number of competing stimuli or thoughts. This capacity for choice is governed by our arousal, which is our level of interest or need. Information received by sensory memories can therefore be treated in two different ways. Or it is rapidly passed into a more permanent memory stage, or overwritten with new information and lost.

Short-term Memory Short-term memory acts temporary recall of information, which might be needed for short periods of time and that related specifically to the situation the individual is in. This is a rapid access type of memory, in order of 70 milliseconds. However, the information stored in this type of memory usually have a very short duration, in the order of 200 milliseconds. Short-term memory has also a limited capacity. According to Miller in [68], typically a person can remember 7 ±

2 objects (i.e. this is frequently referred to as Miller's Law). This limited capacity of short-term memory produces a subconscious desire to create symbols that can be grouped into a larger whole, and so optimize the use of the memory.

Long-term Memory Long-term memory is our main resource of storing information. Here we store factual information, experiential knowledge, and procedural rules of behaviour. It differs from short-term memory in a number of significant ways. First, it has a much larger capacity. Secondly, it has a relatively slow access time of approximately 100 milliseconds. Thirdly, forgetting occurs more slowly in long-term memory. These distinctions suggest a memory structure with several parts. Episodic memory represents our memory of events and experiences in a serial form. It is from this memory that we can reconstruct the actual events that took place at a given point in our lives. Semantic memory, on the other hand, is a structured record of facts, concepts and skills that we have acquired. The information in semantic memory is derived from that in our episodic memory, such that we can learn new facts or concepts from our experiences.

Moreover, there are three main activities related to long-term memory: storage or remembering of information, forgetting and information retrieval. The process of remembering information happens when the short-memory content is stored in long-term memory by rehearsal. The repeated exposure to a stimulus or the rehearsal of a piece of information transfers it to the long-term memory. For the process of forgetting, the *decay* theory suggests that the information stored in long-term memory may eventually be forgotten. On the other hand, the *interference* theory defends that information is lost from memory by acquiring new information that causes the loss of old information. Information retrieval can be distinguished between two types: recall and recognition. In recall the information is reproduced from memory. In recognition, the presentation of the information provides the knowledge that the information has been seen before. Recognition is less complex cognitive activity than recall, since the information is provided as a cue. However, recall can be assisted by the provision of retrieval cues, which enable the subject quickly to access the information in memory.

2.2.4 Reasoning

Humans can use information to reason and solve problems, in a manner that no other artificial or natural being can do. According to Johnson and Laird in [69] "Thirty years ago psychologists believed that human reasoning depended on formal rules of inference akin to those of a logical calculus. This hypothesis ran into difficulties, which led to an alternative view: reasoning depends on envisaging the possibilities consistent with the starting point—a perception of the world, a set of assertions, a memory, or some mixture of them. We construct mental models of each distinct possibility and derive a conclusion from them. The theory predicts systematic errors in our reasoning, and the evidence corroborates this prediction. Yet, our ability to use counterexamples to refute invalid inferences provides a foundation for rationality. On this account, reasoning is a simulation of the world fleshed out with our knowledge, not a formal rearrangement of the logical skeletons of sentences."

Indeed humans can use information even in cases when the information is partial or is not available. Human reasoning is conscious and self-aware (i.e. while we may not always be able to identify the processes we use, we can identify the products of these processes, our thoughts). Furthermore, can think about abstract things, concepts, and solve problems which have never been seen before. Thinking activities can occur with different levels of complexity and may be much directed and the knowledge required is constrained, or require vast amounts of knowledge from different domains.

There are a number of different types of reasoning: *deductive*, *inductive* and *abductive*.

In *deductive* reasoning we reach to the logical conclusion from a given set of premises, which might not necessarily correspond to our notion of truth. For example,

> If it is sunny then the ground is wet.
> It is sunny.
> Therefore the ground is wet.

Is a perfectly valid deduction, even though it conflicts with our knowledge of what is true in the world.

Inductive reasoning implies generalizing from cases we have seen to infer information about cases we have not seen. For example, if every cow we have ever seen was brown, we infer that all cows are brown. Of course, this inference is unreliable and cannot be proved to be true; it can only be proved to be false. We can disprove the inference simply by finding a cow of different colour. Disregarding its unreliability, induction is a useful process, which we use constantly in learning about our environment.

Abduction is the method we use to derive explanations for the events we observe. It reason based on understanding what caused a certain effect. People usually infer explanations this way, and hold onto them until they have evidence that support an alternative theory or explanation. This can lead to problems in using interactive systems. If an event always follows an action, the user will infer that the event is caused by the action unless evidence to the contrary is made available.

Problem-solving, planning, reasoning and decision-making

Dix et al. in [66] described *problem solving*, in a complimentary way to reasoning, as the process of finding a solution to an unfamiliar task, using the knowledge we have. Human problem solving is characterized by the ability to adapt the information we have to deal with new situations. Problem-solving, planning, reasoning and decision-making (i.e. reflective cognition) often involve conscious actions, dialogue with others (or introspective thoughts), and operate various kinds of artifacts, (e.g., maps, books, and pen and paper). These processes allow us to plan and decide about what actions to take, given a range of options and their corresponding consequences. The extent to which people engage in the various forms of reflective cognition depends on their level of experience with a domain, application, or skill. For example, novices tend to act by trial and error, exploring and experimenting with ways of doing things, progressing slowly and with more errors, given they have limited knowledge and will look often for previous knowledge about similar situations. In opposition, experts are able to select optimal strategies for carrying out their tasks, given their decision are made on much more self-acquired knowledge and experience. This, makes them more likely to be able to think ahead about the consequences of deciding for a particular action.

Learning

Learning is another aspect that is considered in Interaction Design. It refers to the cognitive process associated with acquiring and storing new knowledge. In terms of interaction design, Learning can be considered in terms of (i) using a new interactive system or (ii) using a computer-based application to understand a new given topic. Carroll et al. in [70] wrote extensively about how to design interfaces to help users to learn new computer-based skills. A main observation is that people find it very hard to learn by following sets of instructions in a manual. Instead, they much prefer to "learn through doing". Hence, GUIs and direct manipulation interfaces are good environments for supporting this kind of learning, because they support exploratory interaction. It is also extremely important to allow users to roll-back their actions (i.e return to a previous state if they make a mistake). For the particular case when users are learning how to use a new computer-based application, another way of helping users with this process is by using a "training-wheels" approach. This involves limiting the possible functions to the basics and then extending these as the user becomes more experienced in using the system. The underlying rationale is to guide users' attention to perform more complex operations as he familiarises with more simple ones helping him along the "learning curve".

Language processing

Also, associated with reasoning and learning, language processing appears as a concern in Interaction Design. It includes three forms with both similarities and differences: reading, speaking and listening. They are similar in terms of semantics. This means the meaning of sentences or phrases is independent of the mode in which they are delivered. For example, the meaning of the sentence “Artificial social companions are getting smarter” does not change regardless one reads it, speaks it, or hears it.

On the other hand, we can find specific differences between the three modes. For example, written language is permanent while listening is transient. It is possible to reread information if not understood the first time round. This is not possible with spoken information that is being broadcast. Moreover, reading can be quicker than speaking or listening, as written text can be rapidly scanned in ways not possible when listening to speech. However, listening need less cognitive effort when compared to reading or speaking. Children, often prefer listening to narratives than to read the equivalent text.

There are marked differences between people in their ability to use language. The ease with which people can read, listen, or speak differs depending on the person, task, and context. For example, some people prefer reading to listening, while others prefer listening. Likewise, some people prefer speaking to writing and vice versa. Dyslexics have difficulties understanding and recognizing written words, making it hard for them to write grammatical sentences and spell correctly. People who are hard of hearing or hard of seeing are also restricted in the way they can process language.

These similarities and differences led the development of interaction design in many applications either to capitalise on people’s reading, writing and listening skills, or to support or replace them where they lack or have difficulty with them. Some examples include speech-recognition systems that allow users to provide instructions via spoken commands or speech-output systems that use artificially generated speech (e.g., written text-to-speech systems for the blind).

2.2.5 Emotional Influences and Individual Differences

Additionally to what was seen before, we must not forget that humans tend to be constrained by their emotional state when reacting to a given stimulus or when trying to solve a problem. Therefore, in order to design adequate interfaces, and interactive systems, the designer must take into account these psychological characteristics.

Emotion Our emotional response to situations affects how we perform. For example, positive emotions enable us to think more creatively, to solve complex problems, whereas negative emotion pushes us into narrow, focused thinking. A problem that may be easy to solve when we are relaxed, will become difficult if we are frustrated or afraid. Psychologists have studied emotional response for decades and there are many theories as to what is happening when we feel an emotion and why such a response occurs.

More than a century ago, William James proposed the James-Lange theory in [71] (Carl Lange was a contemporary of James whose theories were similar) that emotion was the interpretation of a physiological response, rather than the other way around. So while we may feel that we respond to an emotion, James contended that we respond physiologically to a stimulus and interpret that as emotion.

Cannon in [72], for example, argued that "our physiological processes are in fact too slow to account for our emotional reactions", and that "physiological responses for some emotional states are too similar (e.g. anger and fear), yet they can be easily distinguished". Experience in studies with the use of drugs that simulate broadly the same physiological responses as anger or fear seems to support this as "participants reported physical symptoms but not the emotion, which suggests that emotional response is more than recognition of physiological changes".

Schachter and Singer in [73] proposed a third interpretation that "emotion results from a person evaluating physical responses in the light of the whole situation. So whereas the same physiological response can result from a range of different situations, the emotion that is felt is based on a cognitive evaluation of the circumstances and will depend on what the person attributes this to. So the same physiological response of a pounding heart will be interpreted as excitement if we are in a competition and fear if we find ourselves under attack".

Whatever the exact process, what is clear is that our body responds biologically to an external stimulus and we interpret that in some way as a particular emotion. The implications of this in the system's design are that, depending on the situation the user can react differently towards the system. In situations of psychological agitation, people will be less able to cope with complex problem solving or managing difficult interfaces; whereas if people are relaxed they will be more forgiving of limitations in the design.

Individual Differences Despite the psychological principles and properties that were discussed before apply to the majority of people, we should be aware that humans, and therefore users, are not all the same. Hence, individual differences must be taken into account as much as possible within the designing of interactive systems. These differences may be long term, including gender, physical capabilities and intellectual capabilities. Others may be shorter term, including the effect of stress or fatigue on the user. Still others change through time, such as age. It is useful to consider, for any design decision, if there are likely to be users within the target group who will be adversely affected by our decision. At the extremes a decision may exclude a subsection of the user population. For example, the current emphasis on visual interfaces excludes those who are visually impaired, unless the design also makes use of the other sensory channels. On a more mundane level, designs should allow for users who are under pressure, feeling ill or distracted by other concerns; they should not push users to their perceptual or cognitive limits.

2.2.6 Context-dependent development

Understanding the user would not be complete without regarding the manner he develops. Troadec and Martinot in [74] summarize the last two decades of cognitive development theories. The overall conclusion is that the study of the mind suffered a shift from the classical conception that the mind is a rational, abstract, universal, central, non-biological, a-historic, emotionless, a-social to a new conception that the mind is indeed positioned, framed by the real time, guided by daily routines and culture dependent. In summary, cognition is now thought to be context-dependent and strictly related with biologic principles. Continuing, in their book, we found three main models for context-dependent cognitive development: *ecological model* from Urie Bronfenbrenner, *the developmental niche*, from Charles Super and Sara Harkness and *ecocultural theory* from John Berry. From these three models, the ecological model is the most centred in the organization of the different context levels in mind development. The other two models are more focused on systemic approaches, focusing deeper in the integration of cultural and societal aspects (Figure 2.4). Therefore, we will assume that the ecological model fit better in designing of interactive agents because it allow us to conceptualize the Individual (i.e. our User) and its relationship with different contexts.

2.3 Understanding Context

In the previous section, we presented the human factors that influence designing interactive systems. In this section, we look into the second aspect addressed by Interaction Design concerning the context in which interaction occurs.

Context has been studied extensively in language use, usually with “context,” meaning the history of prior utterances (e.g., Ferstl [75]), but also including other

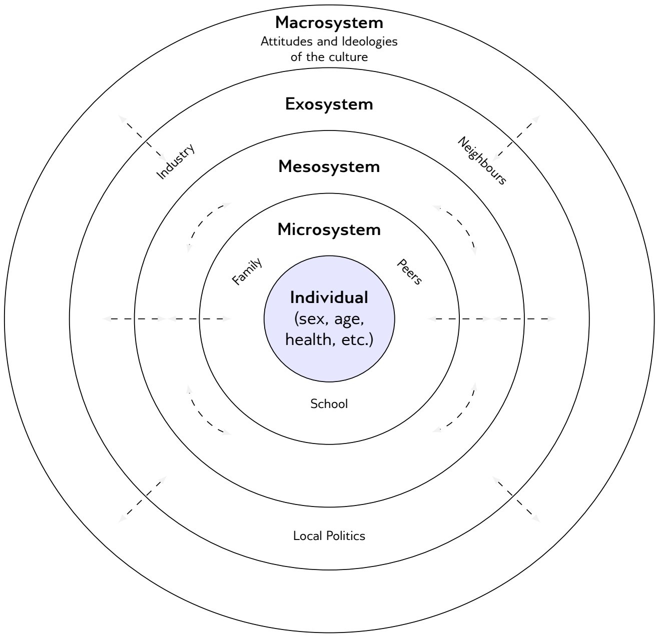


Figure 2.4: Ecological model of development

kinds of context. Holtgraves [76] has found that the status of the speaker relative to the hearer affects whether the literal meaning of an indirect request is activated.

These and other examples have been motivating different scientists to design and implement context-aware systems in a variety of fields of application.

2.3.1 Context definition

Schilit et al. in [41] are recognized as the first to introduce the concept of context-aware systems in ubiquitous computing. Context was described as "*any information that can be used to characterise the situation of an entity*", which include the information about where an entity is, who an entity is with and what resources are nearby. An entity can be a person, a place or an object that is considered relevant to the interaction between a user and a cognitive system, including the user and the system themselves. Context-aware systems adapt according to the location of use, the collection of nearby people, hosts, and available devices, as well as to changes in such things over time. Context-aware applications are presented in four categories: *proximate selection, automatic contextual reconfiguration, contextual information and commands* and *context-triggered actions*. The context is not identified automatically, thus for the different applications, a set of information is considered to be contextual information. For *proximate selection* applications, the context is the location and locate-objects that are nearby. The located-objects can be physical or non-physical.

For *automatic contextual reconfiguration* applications, the context is the location, the task that is being performed and the people involved in the task. For *contextual information and commands* applications, the context is the location, the located-objects nearby, the tasks that are typically associated with the location and the located-objects (contextual information). For *context-triggered actions* applications, the context is similar to that of the contextual information and commands applications. The contextual information was used to adjust the appearance and information displayed on the user graphical interface of each application prototype.

Jason Pascoe in [42] addressed the deployment of context-awareness in wearable computers. Four generic contextual capabilities were presented for context-aware systems that addressed sensing, adaptation, resource discovery, and augmentation. A brief overview has been given of the work in designing and implementing a Contextual Information Service (CIS) to remedy the complexity of obtaining and working with contextual data that prevents the utilization of context-awareness in many applications. The CIS encourages developers to add context-aware features to their software by providing them with an extensible shared model of context that is transparent of any underlying complexities. Pascoe definition for context extend that introduced by Schilit [41] and presented it as a "*...subjective concept that is defined by the entity that perceives it*".

Turner in [77] , defined context and situation as: "*The term context means any identifiable configuration of environmental, mission-related and agent-related features that has predictive power for an agent's behaviour. The term situation is used to refer to the entire set of circumstances surrounding an agent, including the agent's own internal state. Context is thus the elements of the situation that should impact behavior*". A suitable knowledge representation strategy is required in such a way that can be easily created by a human and understandable by the robot. Turner proposed the representation of contextual information using contextual schemas (c-schema), where each schema is organized in a conceptual, content-addressable memory. In his approach a c-schema is an auto-contained description of a specific context, created according a specified format and stored in text file.

Anind Dey et al. in [43] is a reference in terms of defining Context and Context-Awareness Systems. They extended previous definitions given by Schilit [41] and Pascoe [42] to a more general and objective conceptualization. Dey et al. aimed to define context beyond the regular use of synonyms and comparison expressions, which often result in subjective interpretations and difficult the identification of which features should be considered context. According to Dey et al., context is "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.". This means that "... if a piece of information can be used to characterize the situation of a participant in an interaction, then that information is context...". The definition of a context-awareness system is given then as a system that "... uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task...".

Boytsov et al. [78], in reference to Bazire and Brézillon [79], "analyzed 150 definitions of context for different subject areas". The conclusions were that although there is no absolute consensus regarding some aspects of context and its definition, the common understanding is that "*context is the set of circumstances that frames an event or an object*" and "*context acts like a set of constraints that influence the behavior of a system (a user or a computer) embedded in a given task*".

2.3.2 Context models

The numerous definitions for Context motivated the formulation of multiple models for this concept. Schmid and Beig in [80] structured the concept of context using the following model:

- A context describes a situation and the environment a device or user is in.
- A context is identified by a unique name.
- For each context, there is a set of relevant features.
- For each relevant feature a range of values is determined (implicit or explicit) by the context.

In terms of this model, they proposed to develop a hierarchically organized feature space for context. At the top level they proposed to distinguish context related to *human factors* in the widest sense, and context related to the *physical environment*. For both general categories further classification into three categories each. The six categories at this level provide a general structure for context. Within each category, relevant features can be identified, again hierarchically, whose values determine context. Additional context is provided by *history*, that is by changes in the feature space over time.

Human factors-related context is structured into three categories: *information on the user* (knowledge of habits, emotional state, bio-physiological conditions, etc.), the users *social environment* (co-location of others, social interaction, group dynamics, etc.), and the users' *tasks* (spontaneous activity, engaged tasks, general goals, etc.). Likewise, context related to physical environment is structured into three categories: *location* (absolute position, relative position, co-location, etc.), *infrastructure* (surrounding resources for computation, communication, task performance, etc.), and *physical conditions* (noise, light, pressure, etc.).

The model provided some structure for consideration of context. *For pragmatic use of context, the general challenge is to identify the set of relevant features in terms of which a situation or environment can be captured sufficiently. Situations and environments are generally characterized by a large degree of continuity over time, so that context history itself becomes an important feature for approximation of a given situation or environment*.

The types of context according to Abowd and Dey [81] are *location*, *identity*, *activity* and *time*. These were considered *primary* context types for characterizing the situation of a particular entity. The primary pieces of context for one entity can be used as indices to find the secondary context for that same entity as well as the primary context for other related entities. In this categorization, they proposed a simple two-tiered system. The four primary pieces of context are on the first level. All other types of context are on the second level. The secondary pieces of context share a common characteristic: they can be indexed by primary context because they are attributes of the entity with primary context.

Bisgaard et al. [82] analysed previous definitions and models up until the mid 2000s, and produced a survey that summarizes articles representing the general body of literature on context awareness in Human-Computer Interaction (HCI). Their conclusions resulted in four major context features: *Location*, *Time*, *Identity* and *Environment*; a complimentary group of features include: *Social setting*, *Network*, *Season*, *History*, *Task/Activity* and *Device*. However, when mapping their application in context-aware systems, they concluded that a smaller context space is used, typically defined by three context features. The resulting "workable" context features would be fused into a group of five: *Location*, *Identity*, *Time*, *Environment* and *Activity*.

Bazire and Brézillon [79] concluded that "...context occurs like what is lacking in a given object for a user to construct a correct representation.". Following their definition, some determining factors that define a context model are as follows: the *entity/subject* concerned by the context, *focus of attention*, *activity*, *situation*, *environment* and eventually, an *observer*.

Bettini et al. [83] described the state-of-the-art in context modeling and reasoning that supports gathering, evaluation and dissemination of context information in pervasive computing. Most prominent approaches to context modeling and reasoning are rooted in database modeling techniques and in ontology-based frameworks for knowledge representation. They also presented state-of-the-art techniques to deal with high-level context abstractions and uncertainty of context information. The survey finally introduced hybrid approaches as an attempt to combine different formalisms and techniques to better fulfill the identified requirements.

2.4 Evaluating Interaction Design

2.4.1 Usability assessment and evaluation

The literature describes a large number of methods and instruments used to ensure the quality of the usability of a product or service [84, 85]. There are usability evaluation methods for all design and development phases, from initial definition to final modifications of a product or service [86]. Furthermore, some of these methods are only suitable for a specific stage of the development process.

Within the usability evaluation methods, questionnaires assume a significant importance for qualitative self-reported data collection related to the characteristics, thoughts, feelings, perceptions, behaviors or attitudes of the users. Questionnaires have the advantage of being low budget techniques, that do not require measurement equipment, and their results reflect the users' opinions. They also provide useful information about what are the strengths and weaknesses of a product or service.

Since the 80s of the last century, researchers felt the need to develop and evaluate products and services in a systematic and methodical way, considering psychometric properties of usability questionnaires. Consequently, several questionnaires were developed and validated, and have long been used in the usability evaluation of products and services.

System Usability Scale

System Usability Scale (SUS) [87] is a 10 item questionnaire (5 point Likert scale) giving a global view of subjective assessments of usability in terms of effectiveness (e.g., can users successfully achieve their objectives?), efficiency (e.g., how much effort and resource is expended in achieving those objectives?), satisfaction (e.g., was the experience satisfactory?).

Post-Study System Usability Questionnaire

The Post-Study System Usability Questionnaire (PSSUQ) [88] is a usability evaluation questionnaire developed by IBM. It is composed by 19 items aimed at addressing five usability characteristics of a product or service: i) rapid completion of the task; ii) ease of learning; iii) documentation quality and online information; iv) functional adequacy; and v) rapid acquisition of productivity.

Usefulness, Satisfaction and Ease of use

The Usefulness, Satisfaction and Ease of use (USE) was originally developed by Arnold Lund in 2001 [89]. It is a self-perceived usability questionnaire with 30 items, and each item with a seven-point *Likert* rating scale. Users are asked to rate agreement with the statements, ranging from strongly disagree to strongly agree.

2.5 Summary discussion

In this chapter, we reviewed the foundations of Interaction Design that are relevant to understand the outcomes of our work. We noted that

- The fundamental concern for Interaction Design is establishing principles and a methodology that revolves around *understanding the user* and the *context* (e.g. types of activities the user is doing) when interacting with the product.
- In spite of considering the eventual physical limitations of the user, interactive systems demand mostly from the users' cognitive processes. The specific kinds of processes that have been described as cognitive processes include attention, perception and recognition, memory, reasoning, problem solving, planning and decision making, learning, and language processing (i.e reading, speaking and listening).
- Context was defined as the set of information which constrain the performance of an agent while attempting to execute a desired behaviour. In spite of the characteristics of that agent, any set of information will be only considered to be context if it anticipates how the agent should behave when that information is present. We concluded from our survey that the most significant context entities are *Location*, *Identity*, *Time*, *Environment* and *Activity*. We will refer to these context entities later when we describe our framework in chapter 5.

Chapter 3

Designing Artificial Social Companions

In the previous chapter we introduced the foundations of Interaction Design allowing us to understand the main aspects that we must address when designing interactive systems. Specifically, what regards understanding the user and context in which interaction takes place. Recalling our goal of developing artificial social companions capable of proactively supporting people in their everyday life within their environment, in this chapter we will cover what they are and how they are typically designed and implemented.

3.1 Artificial Social Companions

In the next two points, we will present an overview of the two types of artificial social companions we addressed in our work: Social and Service Robots and Virtual Companions.

3.1.1 Social and Service Robots

The state of the art for social and service robots is difficult to review, namely since the boundaries of this category of robots is, in itself, generally acknowledged by the scientific community as being difficult to pinpoint. In fact, the only consensus seems to lie on the premise that this classification applies to any robot that performs useful services to humans, with varying degrees of autonomy, with the only exception being manufacturing operations, which are relegated to traditional industrial robotics (see, for example, the provisional definition of service robotics by the International Federation of Robotics[7].

Although there have been attempts since the 1980s to produce convincing solutions in service robotics, this field has become increasingly relevant in the past decade. During this time, major technological hurdles have been conquered, and robots have become more and more affordable. Consequently, service robots have become a major part of assisted living technology [90], even in its broader sense of giving assistance to promote the welfare of any human being, and not just elderly or people with physical disabilities. On the other hand, service robots have also been finding space in entertainment and assisting or replacing humans in tedious or difficult tasks [2], namely in industrial applications [91]. The fuzzy distinction between this type of robots and all other possible categories implies that service robots can range from devices with very limited cognitive skills, such as **iRobot's™ Roomba** [92], to the so-called "socially interactive robots", in other words, robots with varying degrees of social skills [1].

Thus, we analyse below a set of relevant research projects, commercial products and trending consumer devices that aimed to address the topic of service and social robotics. This survey will help us to understand better the variety and amount of efforts that have been carried out over the last decade related to the development of this type of ASCs.

Research projects

The **ROBOTS@HOME**[8] (2007-2010) FP6 project aimed to provide an affordable and efficient open mobile platform for the introduction of robots into the home. Autonomous navigation in realistic home environments, for instance with typical furniture configurations, is one of the key problems to be addressed in ROBOTS@HOME.

The **DEXMART**[9] (2008-2012) FP7 project aimed to bridge the gap between the use of robots in industrial environments and the future use of robots in everyday home environments, focusing on personal and service robotics where dexterous and autonomous dual-hand manipulation capabilities are required, like the depicted in Figure 3.1a.

EL-E[10] (2008-2009) in Figure 3.1b, developed by GeorgiaTech, likewise investigate object manipulation in home environments with robots in order to retrieve

[7] `http://www.ifr.org/service-robots/`
[8] `https://cordis.europa.eu/project/rcn/80548_en.html`
[9] `https://cordis.europa.eu/project/rcn/85328_en.html`
[10] `http://pwp.gatech.edu/hrl/el-e-an-assistive-robot/`

un-modelled everyday objects for people with motor impairments.

Other projects focus on problems related to the execution of general household tasks. The **ARMAR**[11] (2000) project developed a humanoid robot, in Figure 3.1c, with two arms suitable for basic household tasks such as loading and unloading a dishwasher.

(a) (b) (c)

Figure 3.1: DEXMART project robot(3.1a), EL-E(3.1b) by GeorgiaTech, and ARMAR project robot(3.1c).

Other research and development projects focus on advanced human-robot interaction technologies for assistance. The CMU Pittsburg **NurseBot**[12] (2000-2002) project, in Figure 3.2d, was one of the first research endeavours that dealt with robotic assistance for the elderly. The robotic platform was used to evaluate concepts for such a companion, including intelligent reminding functions, tele-presence applications, surveillance, basic social interaction and help for physically impaired persons. The robot learned typical movement patterns from the people it cares for by observing and tracking them and inferring behaviours.

The goal of the **RoboCare**[13] (2002-2006) project was to build a multi-agent system, in Figure 3.2a, that generates user services for human assistance, for instance in healthcare institutions and domestic environments. The project focused on a specific technical aspect of multiple robot organisation and coordination, modelled as a multi-agent system, and the development of a hardware and software framework to support the system.

CareBot[14] (since 1997) from GeckoSystems, was developed for the scenario of assisting elderly people. It uses fuzzy logic to navigate in domestic environments

[11] `http://h2t.anthropomatik.kit.edu/english/397.php`
[12] `http://www.cs.cmu.edu/~flo/`
[13] `http://robocare.istc.cnr.it/`
[14] `http://www.geckosystems.com/`

using a hybrid control architecture that requires access to an external Personal Computer for computation. The system, in Figure 3.2b, performs simple duties such as vacuum cleaning, patrol and errand running. The main focus of development is again on the robust navigation in real-world home environments.

COGNIRON[15] (2004-2007) was an FP6 project focusing on the development of a robot, in Figure 3.2c, whose ultimate task is to serve humans in general, not primarily elderly or cognitively impaired persons, as a companion in their daily life. To this end, COGNIRON studies the perceptual, representational, reasoning and learning capabilities of embodied robots in human-centred environments. The project aimed at developing methods and technologies for the construction of cognitive robots that can evolve and grow their capacities in close interaction with humans in an open-ended manner. Next to required functionalities for sensing, moving and acting, such a robot will exhibit the cognitive capacities enabling it to focus its attention, to understand the spatial and dynamic structure of its environment and to interact with it, to exhibit a social behaviour and communicate with other agents and with humans at the appropriate level of abstraction according to context.

SRS[16] (2010-2013) FP7 project focused on the integration of a robotic system that "shadows" (i.e. acts as a physical representative), for instance of children or caregiver, for a care recipient. To achieve this, the robotic system can be tele-operated robustly via real-world communication infrastructure. A further aim is to adaptively, autonomously and effectively execute remotely controlled service tasks, using robotic self-learning mechanisms.

(a) (b) (c) (d)

Figure 3.2: LEA from RoboCare project (3.2a), CareBot (3.2b), Cogniron robot (3.2c) and NurseBot (3.2d).

DOMEO[17] (2009-2012) AAL-JP/FP7 project aimed to develop an assistive robotic system that supports cognitive and physical stimulations, helping elderly and disabled person to remain autonomous for as long as possible and to stay at home

[15] http://www.cogniron.org/final/Home.php
[16] http://srs-project.eu/
[17] http://www.aal-europe.eu/projects/domeo/

longer and more safely. DOMEO also included remote presence functionalities such as monitoring and alerting care-givers and relatives to better assist the care-receiver. The project objectives include the development of a robotic walking assistant and of a RobuMate, which is a robot capable of verbal and non-verbal interaction with the user. The project used the Kompai R&D robot, in Figure 3.6a, developed by the French robotics company Robosoft. **Kompai** provides functions including speech recognition for understanding basic commands, localisation and navigation, and basic communication and monitoring functions.

The **KSERA**[18] (2010-2013) FP7 project addressed the research question of how to obtain a successful and effective interaction between a human and a mobile robot in order to guarantee acceptance and adoption of service robots technology, and in order to offer added value to ubiquitous monitoring services. KSERA adopted the NAO robot, in Figure 3.7a, as development platform to address the specific use case of care-receivers suffering from Chronic Obstructive Pulmonary Disease, which makes them susceptible to physical conditions such as environmental pollution, excessive humidity and low air pressure. The envisioned socially assistive robot should monitor the care-receiver's condition and the environmental quality in order to warn, to advise and support care-receivers.

ALIAS[19] (2010-2013) AAL-JP/FP7 project intends to develop a mobile robot system solution that interacts with elderly users provides them with assistance in their daily lives and promotes social inclusion by connecting elderly person to people and events in the wider world. ALIAS envisioned to realize a mobile robot platform that can monitor, interact with and access on-line services in order to realize the integration goal of the project, but does not interact with the physical environment of a care-receiver. This project ended by using the Scitos G5 robotic platform depicted in Figure 3.6b.

Moreover, the current trend in state of the art of platforms is that of extending the robotic agent features with those integrated with the ambient intelligence of smart space environments.

The FP7 project **MOBISERV**[20] (2009-2013) aimed to develop and integrate state of the art technology in a coordinated, intelligent and easy to use way in order to support independent living of older persons in their private home or various degrees of institutionalization for as long as possible. To reach this objective, MOBISERV developed a personal intelligent platform that consists of various middleware and devices and an integrated autonomous robot unit. The developed system supported health status monitoring, secure tele-alarm and health reporting and nutrition ad-

[18] https://cordis.europa.eu/project/rcn/93796_en.html
[19] http://www.aal-europe.eu/projects/alias/
[20] https://cordis.europa.eu/project/rcn/93537_en.html

vice. The communication platform and system to integrate the different components of the MOBISERV platform were designed to be project-specific customized implementations that connected the individual components via wireless local are network communication.

The **CompanionAble**[21] (2008-2012) FP7 project was a key research and development project that explicitly realized synergies of robotics and ambient intelligent technologies to provide for care-givers assistive environment in the AAL space. CompanionAble aimed to support cognitive stimulation and therapy management of care-receivers, mediated by a mobile robotic companion working collaboratively with the stationary smart home environment. The distinguishing advantages of the CompanionAble Framework Architecture arisen from the objective of graceful, scalable and cost-effective integration. CompanionAble addressed issues of social inclusion and homecare of persons suffering from chronic cognitive disabilities prevalent among the increasing European older population. A participative and inclusive co-design and scenario validation approach drove the RTD efforts in CompanionAble; involving care recipients and their close carers as well as the wider stakeholders. The project aimed to ensure end-to-end systemic viability, flexibility, modularity and affordability as well as considerations of the overall care support governance and integration with quality of experience issues such as dignity-privacy-security preserving responsibilities. CompanionAble was evaluated at a number of test beds representing a diverse European user-base as the proving ground for its socio-technical-ethical validation.

The **FLORENCE**[22] (2010-2013) FP7 project developed a robotic solution that uses a robot as the central point of access for care receivers. The focus of FLORENCE was to provide a multipurpose mobile robot platform for AAL scenarios that could be accepted by care-receivers and cost-effective for society and care-givers. FLORENCE envisioned using the developed robot as the central connecting and coordinating element between several stand-alone AAL services within a home environment. The project intended to support the delivery of care and coaching services so that elderly care-receivers will be able to remain independent for a much longer period of time.

The **SocialRobot**[23] (2011-2015) FP7 project, also aimed to provide assistive services and companionship Robots for the elderly people. The major challenges addressed in the project included navigating indoors and unstructured environments and provide affective and empathetic user-robotic interaction, taking into account the capabilities of and acceptance by elderly users. The project made an important contribution in terms of identifying individual needs and requirements related to ageing (e.g. physical mobility limitations or/and cognitive decline), and provision of

[21] https://cordis.europa.eu/project/rcn/85553_en.html
[22] https://cordis.europa.eu/project/rcn/93917_en.html
[23] http://mrl.isr.uc.pt/projects/socialrobot/

support through timely involvement of care teams, consisting of different groups of people (family members, neighbours, friends) that collaborate dynamically and virtually, means independently of time and their physical locations, behaviour analysis to adapt social relationships and contexts of the elderly people as they age.

GrowMeUp[24] (2015-2018) H2020 project, built on top of SocialRobot results and extended them with a novel concept and features including context-awareness, as it able to learn the older persons needs and habits over time and adapt its functionality, compensating for the older persons capabilities degradation over time; and cloud robotics to enable shared and distributed knowledge with multiple robots performing a similar job.

(a) (b) (c)

Figure 3.3: CompanionAble (3.3a), FLORENCE (3.3b) and SocialRobot/GrowMeUp (3.3c).

Robotic platforms, market ready and consumer products

Currently we can find some robotic platforms with a high technology readiness level, which were adopted in some of the previously mentioned research project and are being used currently in some undergoing works. This robotic platforms include therapeutic tools, tele-presence devices and personal assistants, and also toys.

PARO in Figure 3.4a, is probably the most notorious example in terms of application of social robotics as a therapeutic tool[25]. This advanced interactive robot allows the documented benefits of animal therapy to be administered to patients in environments such as hospitals and extended care facilities where live animals present treatment or logistical difficulties. It is commonly associated to stimulate interaction and socialization in people suffering from dementia. PARO has been found to reduce

[24] `http://www.growmeup.eu/`

[25] it holds a World's Most Therapeutic Robot certified by Guinness World Records

patient stress and their caregivers, showing to have a benefic psychological effect on patients, improving their relaxation and motivation.

AIBO in Figure 3.4b, was developed by Sony and was first introduced in 1999 as the first consumer robot of its kind to be offered to the public. This little robot's name comes from **A**rtificial **I**ntelligence ro**BO**t and is also the Japanese word for 'Companion' or 'Friend'. AIBO's appearance resembles a small dog. Its "personality" can develop from a newborn puppy to an adult depending by the interaction with their owners and surroundings.

ASIMO in Figure 3.4c, is most recent remarkable result of two decades of humanoid robotics research by Honda engineers. In the future, Honda engineers are expecting ASIMO might help with important tasks like assisting the elderly or a person confined to a bed or a wheelchair.

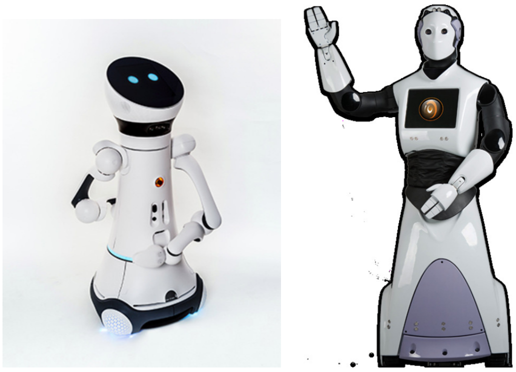
(a) (b) (c)

Figure 3.4: Paro (3.4a), AIBO robot by Sony (3.4b), ASIMO robot by HONDA (3.4c).

Care-O-Bot in Figure 3.5a, is a mobile service robot which has the capability to interact with and assist humans in typical housekeeping tasks. The hardware of the robot is based on two completely autonomous systems: a mobile platform with adjustable walking supports and a top level system with a 6-degrees-of-freedom manipulator and gripper designed for handling objects in home environments and a tilting sensor head for 3D scanning and vision.

REEM in Figure 3.5b, is a humanoid robot series from PAL Robotics combines home navigation, user and assistance and advanced functionalities such as entertaining users by playing chess with them.

Hector in Figure 3.3a, is a robot designed to assist elderly people who suffer from mild cognitive impairment. Developed as part of the CompanionAble project by researchers at the Smart Homes foundation in the Netherlands. According to Herjan van den Heuvel of Smart Homes this robot *"... acts as a coach and companion,*



(a) (b)

Figure 3.5: Care-O-Bot4 by Fraunhofer IPA (3.5a) and REEM by Pal Robotics (3.5b).

and supports the user by means of suggestions, encouragements and reminders on a physical, cognitive and social level. ...".

Kompai in Figure 3.6a, first designed in 2009, this robotics platform was used to generate dozens of robots that were used to conduct proofs of concept over 1,000+ days of experimentations with different users. Currently in its second version, KOMPAI-2 offers a new and improved design, including all the IoT and Big Data technologies, in addition to state-of-the-art navigation and Human-Machine Interface. It has been on the market since early 2016.

Scitos G5 in Figure 3.6b, is an all-purpose mobile base by MetraLabs. It brings along all the basic functionality that enables rapid scenario implementation and testing. For the launch of your research projects, this robotic platform offers a highly innovative system that combines the latest technological developments in robotics and established industrial technology.

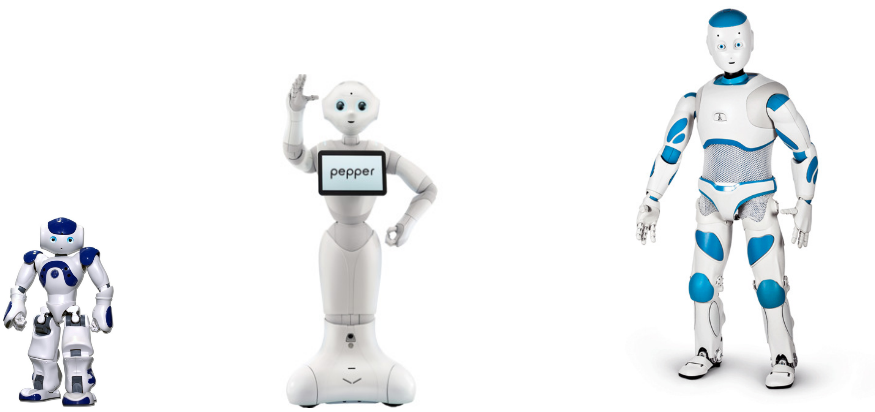
Giraff in Figure 3.6c, is a tele-presence robot that allows one person to virtually enter or visit another remote place. This is of course facilitated by controlling the robot from a personal computer, via the Internet, using an accessible user graphical interface. The user can move freely about the remote place simply by moving the computer mouse, and interact with the residents there via videoconferencing.

SoftBank Robotics promotes three of the most well known robotics platforms used to develop social robots: **NAO**, **Pepper** and **Romeo**. **NAO** in Figure 3.7a is the first humanoid robot designed by SoftBank Robotics (i.e. originally a French company known as Aldebaran). It associates a matured programming tool and an appealing and friendly look and feel, making it currently a reference in the sectors of education and research. **Pepper** in Figure 3.7b is the second platform developed by this company. It is being claimed as the first emotional humanoid robot, which is capable

(a) (b) (c)

Figure 3.6: Kompai used in DOMEO project (3.6a) and Scitos G5 used in ALIAS project (3.6b) and Giraff/Giraff+ (3.6c).

of recognizing main emotions, welcoming and greeting, informing and entertaining visitors at numerous public places. **Romeo** in Figure 3.7c is the third robotic platform to be developed by SoftBank Robotics, which it is still in a research phase. Romeo is currently being used for developing and testing new technologies and solutions to help people with limited autonomy, foreseeing its application on assisting people in their day-to-day lives.



(a) (b) (c)

Figure 3.7: Social robot platforms from SoftBank Robotics: NAO used in KSERA project (3.7a), Pepper (3.7b), Romeo (3.7c).

We summarize, in Table 3.1, some of the relevant characteristics of the most adopted robotic platforms in European social robotics research projects.

Table 3.1: Comparison between different robotic platforms adopted by the most relevant European projects developing social robots.

Feature	**Hector**	**GrowMu**	**NAO**	**Scitos G5**	**Kompai**	**Giraff**	**Care-o-Bot**
Facial expression recognition	No	Yes	No	No	Yes	No	No
Facial expression synthesis	No	Yes	No	No	No	No	No
Speech recognition	Yes	Yes	Yes	Yes	Yes	No	Yes
Speech synthesis	Yes	Yes	Yes	Yes	Yes	No	Yes
Conversation (Dialogue Management)	Yes	Yes	Yes	Yes	Yes	No	Yes
Reminder or agenda service	Yes	Yes	Yes	Yes	Yes	No	Yes
Guidance service	Yes	Yes	Yes	No	Yes	No	Yes
Armed robot	No	No	Yes	No	No	No	Yes
Touch screen	Yes	Yes	No	Yes	Yes	Yes	Yes
Integration with smart environment	Yes	Yes	No	No	Yes	No	Yes
Reached technology demonstrated in relevant environment (TRL6)	No	No	Yes	No	No	No	Yes

Market ready and Consumer Social Robots

In a somewhat complimentary approach, some of the big players operating in the market of personal assistants, like Amazon or Google, launched into the market some products that aim to deliver partially the services expected from social robots.

Amazon Echo, in Figure 3.8a, devices connect to the voice-controlled intelligent personal assistant service Alexa, with the capability of voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, playing audiobooks, and providing weather, traffic and other real-time information. It can also control several smart devices acting as a home automation hub. Amazon Echo became widely available in the United States on mid-June 2015, but since February 2018 it is available in 36 countries (including 21 European countries).

Google Home, in Figure 3.8b, was developed by Google and released in the United States in November 2016, with subsequent releases globally throughout 2017. Google Home speakers enable users to speak voice commands to interact with services through Google Assistant. It integrates in-house and third-party services, allowing users to listen to music, control playback of videos or photos, or receive news updates entirely by voice. Google Home devices also support integration with home automation devices, letting users control smart home appliances with their voice.

Sphero, in Figure 3.8c, is a spherical robot designed by Sphero. It is a white orb wrapped in polycarbonate, capable of rolling around, and controlled by a smartphone or tablet. Mechanically the inventors compare the inside mechanics of the ball to a two-wheel electric vehicle such as the Segway PT. Sphero began the connected play revolution in 2010 by creating something unlike anything. The fusion between robotics and digital technology into immersive entertainment experiences changing the way the world thinks about play.

(a) (b) (c)

Figure 3.8: Amazon Echo (3.8a), Google Home (3.8b) and Sphero (3.8c).

Crowd funded and Trending Social Robots

More recently, beginning in mid-2010', we are observing to the increasing popularity of crowd funding platforms, like Kickstarter and Indiegogo. This trend is leading to the democratization of funding access supporting the development of new devices, including social robot platforms. Some examples of Crowd funded and Trending Social Robots are depicted in Figure 3.9.

Jibo, in Figure 3.9a, Jibo made headlines back in 2014, raising more than $3 million from excited backers on Indiegogo. Developed by a MIT professor named Cynthia Breazeal, it was pitched as the “world’s first social robot.”. Jibo followed a minimalist approach. It has no legs, it has a shiny white plastic body and overall is a curvy cylinder with a flat screen for a head on top that can move and provide the interface for the user.

Buddy, in Figure 3.9b, is an open source and easy to use robotic platform developed by Blue Frog Robotics. Another example of a product resulting from successful Indiegogo campaign in 2015. Buddy services include home security, connect to smart home appliances, promote social interaction, play multimedia contents, entertainment and some services aimed for elderly care (e.g. fall detection safety and medication reminders).

Mykie, in Figure 3.9c, which stands for "my kitchen elf", is one example of the benefits of connected kitchens and an innovative Home Connect concept from BSH Hausgeräte GmbH. Mykie listens, answers questions, and projects information and recipes. It keeps track of the household at all times and entertains users. Mykie can be operated through voice controls or a touchscreen, and it assists users with recipes through step-by-step videos.

Kuri, in Figure 3.9d, developed by Mayfield Robotics, a Bosh-backed startup, introduced in the early 2017. Kuri has a camera, microphone array, speakers, and touch sensors, and a laser-based sensor array that it uses for obstacle detection, localization, and navigation. Besides mobility, what makes Kuri unique in its category is the fact that it has no display (besides a color-changing light on its chest), and that it does not even try to talk to you, as Pepper and Jibo do.

LG Hub, in Figure 3.9e, was launched by LG in early 2017 and it also belongs to the same category of the previous examples. LG Hub features a circular "face" with a screen displaying eyes – perched atop a softly conical white body. The robot can re-orient itself to face the user and bob along with music it is playing, but it is otherwise stationary. The screen can also display images and videos using some sort of basic web browser.

3.1.2 Virtual Companions

Virtual Companions are computer-animated characters exhibiting a certain level of intelligence and autonomy as well as social skills to simulate human face-to-face conversation, and the ability to sense and respond to user affect. A number of Virtual

(a) (b) (c) (d) (e)

Figure 3.9: Jibo (3.9a), Buddy (3.9b), Mykie by BOSH (3.9c), Kuri (3.9d) and Hub by LG (3.9e).

Companion systems have been successfully developed for various target applications to monitor, encourage, and assist older adults. Based on recent research findings [93–96], it is anticipated that this is a promising technology that can play an important role in maintaining the health, wellness, and independence of older adults in the future, either by complementing human care or acting as an alternative for those who cannot receive it due to high cost or low availability of care personnel.

A number of researchers have explored Virtual Companions that interact with users over multiple conversations, ranging from a handful of interactions to hundreds of interactions spanning long-term periods [94, 97, 98]. Nonetheless, most of the developed Virtual Companion systems are designed for specific controlled environments and have rarely made the step out of the laboratory as autonomous applications in real-world settings. As a consequence, there is still little information available about how autonomous Virtual Companions perform and which factors influence their acceptance and success in contexts such as private households. To achieve useful and successful virtual agents, that maintain their users engaged in beneficial long-term relationships, we need to integrate these systems seamlessly in real-world environments and make them capable of interacting with humans autonomously, in an intuitive, natural and trouble-free way in everyday situations.

Virtual Companions are typically represented in the form of human or animal bodies that are specifically lifelike and believable in the way they behave. They simulate human-like properties in face-to-face conversation, including the abilities to recognize and respond to verbal and non-verbal input, generate verbal and non-verbal output, such as mouth, eye and head movements, hand gestures, facial expressions, and body posture and can deal with conversational functions such as turn taking, feedback and repair mechanisms [99]. Due to these characteristics, Virtual Companions provide familiar and non-threatening interfaces, especially useful for building systems that are easy to use, engaging and gain the trust of their users, even those suffering from age-related or other cognitive impairments [100,101]. We briefly review prior work on Virtual Companions and agent-based systems designed to address the needs of older adults.

Studies suggest that Virtual Companions dealing with emotion and affect are particularly capable of capturing the user's attention, engaging them in active tasks and entertaining them [102,103], leading to the development of affinity relationships with their human partners [104]. Wizard of Oz [105] experiments showed high acceptance and positive attitude towards Virtual Companions as companions for older adults [101], [106]. A variety of Virtual Companions has been developed aiming to provide social support to isolated older adults [106–108] and to address daily needs for an autonomous living [101]. Agents have also been used as coaches and wellness counselors in health behaviour change interventions for older adults [109,110].

Virtual agents that will be used for extended periods of time require special design considerations compared to systems that are either only used for brief interactions or do not engage the user in social interaction [110]. Relational agents, a term introduced and explored by Bickmore [98], are Virtual Companions designed to form long-term relationships with their users. They are distinct from other types of social agents in their ability to imitate the way people incrementally get to know and trust each other through conversations [110]. They often maintain computational models of affect and relationship and memory of specific interactions, with the intention of recalling and referring to them later so as to evolve relationships with their users [111].

Kasap et al. [112,113], discuss a virtual agent designed for repeated interaction. The agent maintains a relationship model of the user which is updated based on the emotional content of events during a session. This model biases the agent's mood and indirectly influences its behaviour.

A handful of studies were conducted in which autonomous Virtual Companions were installed for prolonged periods in the daily living environments of older adults. In an exploratory pilot study by Ring et al. [114], an Virtual Companion designed to provide longitudinal social support to isolated older adults using empathetic feedback was placed in the homes of 14 older adults for a week. Results demonstrated significant reductions in loneliness based on self-reported mood.

In a randomized controlled trial by Bickmore et al. [109], a virtual exercise coach designed to encourage sedentary older adults to walk more was installed in homes for two months, followed by another ten months where participants had the opportunity to continue the interaction in a kiosk in their clinic waiting room. Participants in the intervention group walked significantly more on average than participants from the control group.

Next, we refer to some examples of ECAs that are commonly known in the Active and Assisted Living community, and overview the common set of features that have been integrated into these systems.

CaMeLi[26] project designed and implemented a Virtual Partner (ViP) able to show a wide variety of human-like understanding and responding and solicit the appropriate services to answer the user's needs/requests offering real time complimentary feedback through voice and a wide spectrum of animated facial expressions. The ultimate goal of the project was to create an innovative virtual assistant that interact and collaborate with the elderly, helping them in performing activities of daily living at home and expressing emotions that will stimulate the act.

Rea is a virtual reality embodied conversational agent whose verbal and nonverbal behaviors are generated from underlying conversational functions and representations of the world and information. Rea implements social, linguistic and psychological concepts that is part of any conversation. Rea has a human-like body appearance, which it is used to add body language during conversation. Conversational skills and comprehension is speech based, thus, each user expression is interpreted and the responses are generated according to which function needs to be achieved.

Greta[27] is a embodied conversational agent that uses a virtual three-dimensional model of a female character. Greta uses verbal and nonverbal communication to interact with the user. Greta can be used with different external Text-to-Speech software. Currently available languages include English, Italian, French, German, Swedish and Polish. Facial Animation Parameters are required to animate faces of different sizes and proportions, the FAP values are defined in Face Animation Parameter Units (FAPU). Greta is used in various European projects: CALLAS, SEMAINE, HUMAINE, and national French projects: ISCC Apogeste, ANR MyBlog3D, ANR IMMEMO.

Virtask[28] has developed **Anne**, a virtual assistant that provides organisations and individual users with novel opportunities in regards to the execution of their

[26] http://www.cameli.eu/
[27] https://perso.telecom-paristech.fr/~pelachau/Greta/
[28] http://www.virtask.nl/wordpress/en/

tasks. Currently, Anne is being successfully used by several companies and organisations, for example, this agent was adopted in DALIA and MyLifeMyWay AAL projects, where it is being employed as a virtual caregiver of elderly and disabled persons, acting as an assistant for daily life activities at home. It was designed to run on Android based consumer devices commonly available in households (TV, phone, tablet). Older adults can use speech to interact with an avatar, which will answer based on data collected through a set of sensors deployed in the household. This ambient intelligence promote a more independent living for older adults. Moreover, informal carers have access to the same avatar, which can tell them what they should do in different situations or just to talk with the person cared for.

V2me – Virtual coach reaches out to me combines virtual and real life social networks to prevent and overcome loneliness of older adults. V2me supports active ageing by improving integration into society through the provision of advanced social connectedness and social network services and activities. V2me was inspired in the **A2E2** project, which is also a financed AAL project. Initially, this system was designed for 7 inches tablets. The key of this choice is portability and low costs; thus, a good graphic processor is needed and the system was extended to All-in-one computers. Professional caregivers and elderly family can monitor user's activity with the system by a designed web interface. By this way is also possible to adjust and configure the way the assistant interacts with the old users. One-on-one test sessions had shown that a lot of elderly are not familiar with new technologies, this includes touch interface of the device, thus preferring human support.

The graphical appearances, for the ECAs referred above, are depicted in Figure 3.10 and summary of the relevant features of each system is provided in Table 3.2.

3.2 Architecture Design for Artificial Social Companions

In the previous section, we presented what are Artificial Social Companions. In this and following sections, we will look into the architecture aspects involved in the technical design of such systems.

Alami et al. [115] presented an integrated architecture allowing a mobile robot to plan its tasks, taking into account temporal and domain constraints, to perform corresponding actions and to control their execution in real time, while being reactive to possible events. The general architecture is composed of three levels: a decision level, an execution level and a functional level. The latter is composed of modules that contain the functions performing sensor data processing and actuation control. The

Figure 3.10: Examples of Virtual Companions: CaMeLi (3.10a), Rea (3.10b), Greta (3.10c), Anne by Virtask (3.10d) and V2me (3.10e).

decision level is goal and event driven, and it may have several layers. According to the application, their basic structure is a planner/supervisor pair that enables users to integrate deliberation and reaction. Alami et al. [116] discussed a decisional framework for human-robot interactive task achievement that aimed to allow the robot to produce behaviors that support its engagement vis-a-vis its human partner and to interpret human behaviors and intentions. The architecture used for controlling the robot followed a similar three-layered architecture from their previous work [115], but highlighted some aspects as situation assessment and context management, goals and plans management, action refinement, execution and monitoring.

This generic architecture model has been adopted in state-of-the-art robotic applications; refer, for example, to the SPENCER project by Triebel et al. in [117].

On the other hand, state-of-the-art interaction models similar to that proposed by Sili et al. [32] typically refer to some degree of adaptation, but explicit models must be provided to rule out the behavior of the system. Their proposed interaction model is depicted in Figure 3.11. In this model, the authors consider that control and decision mechanisms are included within the *Dialogue Manager* component.

In recent works, for example, the study of Devin et al. [118], the authors summarize the essential building blocks to design an architecture for cognitive and interactive robots. The concepts presented may be generalized for human-machine systems overall.

Table 3.2: Comparison between different ECAs implementations.
* - (Without avatar on portable devices)

Feature	CaMeLi	Anne	Rea	V2Me *	Greta
LipSync	Yes	Yes	Yes	Yes	Yes
Platform	Windows	Android, Windows	Android	Android	Windows
Speech Recognition	Windows SDK	Built-in	N/A	Third-party services	Built-in
Conversation (Dialogue Management)	Yes	Yes	No	No	Yes
Emotional State (**D**etect/**S**ynthesize)	D/S	D	D/S	D/S	S
Multi Language	Yes	Yes	No	No	Yes
Multi Avatar	No	No	No	No	No
Social Networks	Built-in	No	No	Yes	No
Camera	Yes, Kinect	Yes	Yes	Yes, Kinect	No
Video Conferencing	Yes	Yes	N/A	Yes	No
Portable	Yes	Yes	No	Yes	No
Emergency Mechanisms	Yes	Yes	N/A	N/A	N/A

Overall, using a top-down approach (i.e. from what is more in contact with the user to what is supporting agent's operation), we can divide the architecture into four main layers: user interfaces, interaction modalities, decision making and knowledge representation.

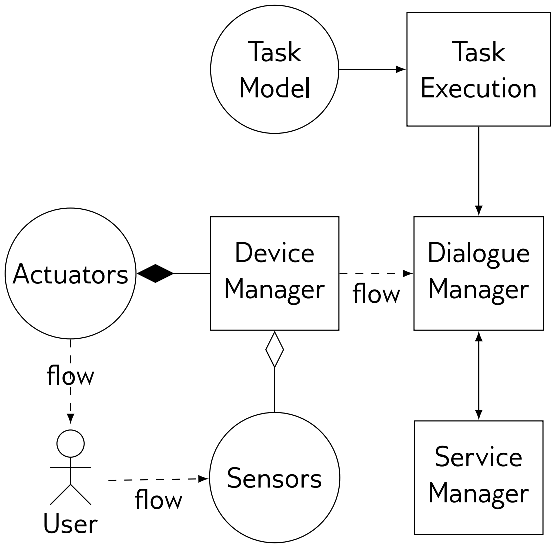


Figure 3.11: Interaction model.

3.3 User Interfaces

The current state of the art in the domain of user interaction with Artificial Social Companions is still mainly based on graphical user interfaces implemented on stationary (often wall-mounted) touch panels (or terminals), portable devices (like touch tablets or mobile phones) or by using TV-sets (often in combination with set-top-boxes) and their remote controls as front end for the elderly. Graphical user interfaces are often text or icon based. Speech input is becoming more and more stable, with some industrial players leading the adoption of such approach (e.g. Siri (Apple), Cortana (Microsoft) and Google Assistant (Google)), nevertheless its application is still limited to simple commands and queries, since surrounding noise and sounds, speaker localisation and optimum input with ambient microphones (by avoiding wearable microphones) are still a challenge and are addressed by several research projects (e.g. CompanionAble etc.).

Recent approaches as well apply avatar technology to enhance (increase acceptance and entertainment value) and personalize user interaction [119–121].

Integrating user interfaces (e.g. touch panel) on a mobile robot platform gives the advantage that terminals do not have to be replicated in all rooms and makes the user interface mobile itself (brought to the user on demand). Very often additional features like moving eyes or head/face emulation are implemented on the robot. Gesture and emotion recognition and input to interact with robot platforms are slowly maturing but are mostly tested under very controlled settings still.

Additionally, multimodal socially-apt robotic cognitive systems would allow for both verbal and nonverbal (i.e. emotion- and body language-based) human-like dialogue enactment. This way, the desires and needs of end-users would be recognised and consequently addressed much more efficiently. Service robots, whether capable of social interaction or not, must address many common design problems.

The analysis of this issue presented by Fong et. al. [1] is still relevant today; according to these authors, these problems include cognition (planning, decision making), navigation, action, human-robot interaction (perception, environment sensing, interfacing with the end-user) and architecture development and middleware. Socially interactive robots, however, as these authors point out, must also extend on human-robot interaction capabilities in order to address issues imposed by social interaction - see also [2] or [3].

3.4 Interaction Modalities

Thanks to the large variety of sensors that are available in the market, it is possible to build sophisticated systems that try to optimize the interaction capacity. The most popular approaches use vision and audio inputs to assure an interaction experience with enough quality. These systems are typically cheap to implement, what constitutes an advantageous factor for those approaches. More sophisticated systems use a multimodal approach, where the data fusion of different sensors could be used to enhance the interactivity, by building more robust perception mechanisms. The latter appeared because human beings usually interact using different ways, frequently making use of multiple channels of communication. Therefore, this approach can use video, audio, range, pressure or chemical sensors altogether to build a robust system. Multimodal approaches can be more reliable than others, what is a plus on its behalf, but they can raise technical issues that may be difficult to solve, and they can be easily more expensive than simpler methods. In the next points we will present definitions for different interaction modalities.

Vision-Based Interaction

Vision-based interaction attempt to provide a broader and more expressive set of input capabilities by using computer vision techniques to process data from visual sensors (e.g. video cameras, lasers, depth sensors), in order to reliably estimate relevant visual information about the user as a passive, non-intrusive, non-contact input modality for human-machine interaction. Consequently the interaction must be made based on visual information presented on the image. The vision system of an artificial agent is responsible to solve tasks like identifying faces, detecting and tracking head and hands poses, capturing human motion, recognising gestures, eye tracking and reading facial expressions. According to Porta in [122], the scope of application of vision-based interaction will be on prototypes intended for office and home use, as we are mainly interested in vision technology applied to ordinary computing environments. Thus, the four main areas in which vision-based interfaces find their maximum expression, namely head tracking, face/facial expression recognition, eye tracking and gesture recognition. Also, vision-based interaction may be used in certain tasks related with navigation on cases where the artificial agent can move (e.g. mobile robot). In such cases, it is common to use vision-based approaches to perform visual mapping and localisation, object recognition and obstacle avoidance. Some works using this approach can be exemplified by [123] and [124], and common applications can be found in Assistive robotics, Human-guided learning or Visual attention mechanism.

Audio-Based Interaction

In this modality, the interaction between human and an artificial agent is made using sounds. Audio-based interaction in HMI typically refers to speech recognition, speech synthesis and non speech audio. Some interesting works done in this field can be found in [125] and [126], where this interaction modality is used for guiding the agent's attention to a specific spot. In [127], the authors developed a solution for mobile workers that need seamless access to communication and information services while on the move. Emphasis was placed on the auditory modality of Nomadic Radio, as it was designed to be used while performing other tasks in a user's everyday environment; a range of auditory cues provided peripheral awareness of incoming messages. Also, notification was adaptive and context sensitive, as messages were presented as more or less obtrusive based on importance inferred from content filtering, whether the user is engaged in conversation and his or her own recent responses to prior messages. In [128], an interesting application of audio-based interaction was used to implement an Audio Notebook for taking notes and interacting with a speech recording, which combines a digital audio recorder and paper notebook, all in one device. Audio recordings were structured based on user note taking activity, and acoustic structuring based on a talker's changes in pitch, pausing, and energy. Moreover, audio-based interaction approaches are being used on automatic emotion recognition in speech. For example, in [125] they focused on studying fear-type emotions occurring during abnormal situations (here, unplanned events where human life is threatened), and intended to apply this approach on public safety applications.

Haptic-Based Interaction

This interaction modality tries to mimic the touch sensation of humans into artificial agents. Usually, this may occur either as an input modality, for example when using pressure sensors; but the most familiar application is as an output modality or haptic rendering [129], which allows users to “feel” virtual objects in a simulated environment. Rendering refers to the process by which desired sensory stimuli are imposed on the user to convey information about a virtual haptic object. For example, in [130] the authors used force feedback in an application for assessing typical tasks related to assembling two components of a mechanical system. In another example [131], haptic feedback was used on computer-assisted training for first-person maintenance tasks targeted to the aerospace industry. This type of interaction modality could be an appropriate way of interaction for people with some kind of visual handicap, which must rely on touch feeling to interact with the environment (i.e. guiding robot for blind people). An example for this approach can be found in [132]. Additionally, haptic interfaces can be designed for any body part capable of feeling touch or proprioceptive stimuli. For example, in [133] researchers have developed systems for simulating walking in a virtual environment, using foot based interactions, even for users in a sitting position.

Brain-Computer Interfaces

This modality has been explored over the last two decades as a radically new communication option for those with neuromuscular impairments that prevent them from using conventional augmentative communication methods. Wolpaw et al. in [134], presented a review of the first international meeting on Brain-Computer Interface (BCI), summarising the state-of-the-art back in early 2000’s. Current BCIs use Electroencephalographic (EEG) activity recorded at the scalp or single-unit activity recorded from within cortex to control cursor movement, select letters or icons, or operate a neuroprosthesis. The signals of brain activity are measured with the help of electrodes and then the patterns of the brain activity are converted into actions in the real world. Therefore, the central element in this approach is the translation algorithm that converts electrophysiological input from the user into output that controls external devices. For example, in [135], this approach is being studied to help locked-in patients suffering from advanced amyotrophic lateral sclerosis to regain some means of communication. Other examples, of use on other scenarios, include control of video games [136] and flight simulators [137].

Multimodal-Based Interaction

As described by Neti et al. in [138], multimodal-based interaction is inspired on the "*... human perceptual principle of sensory integration (e.g. typically, the joint use of audio and visual information) to improve the recognition of human activity (speech recognition, speech event detection and speaker change), intent (intent to speak) and human identity (speaker recognition), particularly in the presence of acoustic degradation due to noise and channel. In this paper, we present experimental results in a variety of contexts that demonstrate the benefit of joint audio-visual processing.*". Therefore, multimodal interaction approaches typically use different types of sensors and devices with the objective of ensure a better interaction between human and artificial agents. This area of study is being actively developed, and can be exemplified by [139–141]. A relevant scenario of application for these approaches, as emphasized in [142], is for helping people with communication disorders due to hearing or visual impairment to interact with technology (e.g. virtual or robotic ASCs). Nowadays, as traditional interfaces are continuously being replaced by mobile, wearable, or pervasive interfaces, some researchers as Almeida et al. in [143] and Munteanu et al. in [144] were motivated to explore multimodal interaction approaches applied on interactive visualisation scenarios, where interaction must be adapted according the device where information is visualised.

3.5 Decision-Based Approaches

Typical approaches used in decision making include Decision Trees [145], Influence Diagrams [146], Multi-criteria decision making [147], or Markov Chains [148].

Up to date, typical works that address decision processes involved in artificial social companions mainly focus on task planning. Altisen et al. [33] formalized a general intermediate layer approach that allowed automatic generation of property-enforcing layers to be used between an application program and a set of resources for which safety properties are defined and should be respected by the global system (the application, plus the intermediate layer, plus the set of resources). Alami et al. [34] focused on the organization aspects of the agent decisional abilities and on the management of human interaction as an integral part of the agent control architecture. Their proposed framework allowed the agent to accomplish its tasks and produce behaviors that support its engagement vis-a-vis its human partner and interpret similar behaviors from him. The framework was applied in a companion agent scenario by the same authors [35] within the scope of the Cogniron project. Cloudic et al. [36] presented the agent control architecture, SHARY, which is dedicated to agent action in the presence of or in interaction with humans. This architecture focused more on task planning but provided support to implement a supervision system adapted to human-machine interaction.

3.6 Knowledge Representation

As ASCs are knowledge based systems, we must note that information models for these type of systems are often represented by means of ontologies. In this scope of application, an *Ontology* is "a representation of the shared background knowledge for a community". This means, it is an implementable model of the intended meaning of a formal vocabulary used to describe a certain conceptualization of objects in a domain of interest. Additionally, although interaction with the world takes place at the level of the individual objects, much of the reasoning process takes place at the level of categories [149]. The organization of objects into categories is a vital part of knowledge representation. In reasoning systems, categories play the important role of being the building blocks of large-scale knowledge representation schemes. Categories help to organize the knowledge base through inheritance. This inheritance allows making predictions about the objects once they are categorized. Additionally, inheritance makes possible the reasoning process of inferring the presence of certain objects from perceptual input or category membership from the perceived properties of the objects and then uses category information to make predictions about the objects. For example, if we describe that a given interaction with the user requires executing a face detection algorithm, and we describe this algorithm requires a video camera; then we can omit to describe explicitly that to execute the interaction with the user requires a video camera, because this relationship will be inferred from the other two relationships.

Tom Gruber in [150] has the most famous definition of ontology in the computer science sense and established the popularity of the word within the domain, though conceptual models of various types have been built within computer science for decades. Gruber's definition is:

"In the context of knowledge sharing, I use the term ontology to mean a specification of a conceptualization. That is, an ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents. This definition is consistent with the usage of ontology as set-of-concept-definitions, but more general. And it is certainly a different sense of the word than its use in philosophy."

3.6.1 Applications of ontological approaches

The representation of knowledge using this approach and associated frameworks and tools can be integrated into a tool chain methodology that ease the development of knowledge based systems. Different authors have studied the application and integration of this type of representation in different domains of application.

Maedche and Staab [151] created MAFRA. MAFRA is part of a multi-ontology system, and aimed to automatically detect similarities of entities contained in two different department ontologies. An interesting contribution of the MAFRA framework is the definition of a semantic bridge. This is a module that establishes correspondences between entities from the source and target ontology based on similarities found between them.

Calvanese et al. [152] proposed a formal framework for Ontology Integration Systems - OISs. The framework provided the basis for ontology integration, which was the main focus of their work.

Kiryakov et al. [153] developed a framework, OntoMapO, for accessing and integrating upper-level ontologies. They provided a service that allowed a user to import linguistic ontologies onto a Web server, which could then be mapped onto other ontologies.

Fernández-Breis and Martínez-Béjar [154] described a cooperative framework for integrating ontologies. In particular, they presented a system that could serve as a framework for cooperatively built, integration-derived (i.e., global) ontologies. Their system aimed towards ontology integration and was intended for use by normal and expert users.

Madhavan et al. [155] developed a framework and proposed a language for ontology mapping. Their framework enabled mapping between models in different representation languages without first translating the models into a common language, the authors claim.

3.6.2 Knowledge representation in artificial social companions

In the particular case of robotic systems some authors also presented some works dedicated to represented relevant information models using ontologies.

Schlenoff et al. in [156] proposed a neutral knowledge representation for capturing relevant information about robots and their capabilities within search and rescue robot systems. They chose an ontological approach for its flexibility of adaptation as the robot requirements evolve. The Robot Ontology was presented containing 230 classes, 245 attributes (properties), and 180 instances. The authors identified as future work: further specialization of the ontology to provide greater level of detail in the areas that have already been addressed; explore other standards efforts and existing ontologies that can be leveraged, such as ontologies for Sensors, Power Source, Materials, Environment; continue to incorporate the requirements from the requirements workshops into the robot ontology structure; Explore the use of reasoning engines to suggest robots as well as configurations (e.g., sensors to be mounted) for different situations.

Weihong et al. in [157] applied an ontological approach into solving the problems of semantic heterogeneity of information in decision-making systems for maritime search and rescue applications. In this paper, maritime search and rescue decision-making ontology reference model was proposed. They designed an ontology for classification on maritime perils as an example to illustrate the process of ontology description based on Protégé. In contrast to traditional knowledge-based approaches, e.g. formal specification languages, ontology seems to be well suited for an evolutionary approach to the specification of requirements and domain knowledge.

Moritz Tenorth and Michael Beetz in [158,159] presented KNOWROB, which combines static encyclopedic knowledge, common-sense knowledge, task descriptions, environment models, object information and information about observed actions that has been acquired from various sources (manually axiomatized, derived from observations, or imported from the web). It supports different deterministic and probabilistic reasoning mechanisms, clustering, classification and segmentation methods, and includes query interfaces as well as visualization tools.

Keshavdas et al. in [160] demonstrated a method for the interaction of a robot with 3D landmarks in a search and rescue environment, based upon ontological knowledge, both pre-existing and additionally computed, as an aid to collaborative efforts by human-robot rescue teams. They performed experiments on some car models and robot configurations and found that poses thus generated by the functional mapping workflow perform far better than those by a naive algorithm of the ontological knowledge. In the future, they plan to perform experiments with a navigating robot, with a camera on a movable arm and plan trajectories around several crashed cars that optimize the amount of visualization inside these cars. Further, they plan to extend the notion of openings and containers to other use cases e.g., entering a hole into a room of known dimensions, climbing a known stairway and so on.

Prestes et al. in [161] proposed the Core Ontology for Robotics and Automation (CORA ontology), which was published in 2015 as the standard ontology for autonomous robots IEEE 1872. This ontology has been developing by the IEEE Autonomous Robotics (AuR) working group[29] for this standard and is still actively preparing the second version of the standard IEEE 1872.2 to be published by 2020, which aims to extend and complement the current description.

3.6.3 Context Ontologies

As presented in the previous chapter, one of the main concerns in Interaction Design is to address the context in which the system is being used. Therefore, we must understand how this information can be represented in an artificial social companion.

The common approach to represent context information is using explicit descriptions about the environment conditions and behaviours to be executed in the present of those conditions. Complex domains of knowledge require more general and flexible representation besides explicit representations (e.g. first-order logic). For example, abstract and general concepts - such as Event, Time, Physical Objects, and Beliefs - that occur in many different domains are not well suited for first-order logic representation [149].

However, ontological representation is getting more popular, as more concrete examples appear and are available for integration into context-aware systems.

Strang and Linnhof-Poppien in [162] (2004) conducted a survey of the the most relevant approaches to modeling context for ubiquitous computing. They compared six approaches (key-value models, markup scheme models, graphical models, object oriented models, logic based models and ontology based models) regarding the fulfilment of six requirements arising from ubiquitous computing (distributed composition, partial validation, richness and quality of information, incompleteness and ambiguity, level of formality, applicability to existing environments). They concluded that the most promising approach for context modeling for ubiquitous computing environments with respect to the requirements listed can be found in the ontology category. A list of Context models ontologies is represented in 3.3.

Table 3.3: Ontologies that represent Context models

Name	Details	
ASC/CoOL	**Author (Year)**	Thomas Strang (2003)
	Publication	Applications of a Context Ontology Language

[29] https://standards.ieee.org/develop/wg/Autonomous_Robotics.html

	Description	Aspect-Scale-Context Information model and Context Ontology Language, which supplemented ASC with extensions for Web Services and others. Context referred to the provision of services and a mean to ensure interoperability.
	URI	`http://context-aware.org/schema/cool.owl`
	Status	inactive and not available.
ConOnto	**Author (Year)**	Mohamed Khedr (2004)
	Publication	PhD Thesis
	Description	ConOnto describes the different aspects of context-aware systems. ConOnto includes location, time, activities, software and hardware profiles. ConOnto also includes meta-information that describes negotiation and fuzzy ontologies to be used in systems that will negotiate and infer about context information.
	URI	`http://www.site.uottawa.ca/~mkhedr/contexto.html`
	Status	inactive and not available.
CONON	**Author (Year)**	Wang, X.H (et. al) (2004)
	Publication	Ontology based context modelling and reasoning using OWL
	Description	Initially based on the same idea of ASC/CoOL approach; an upper ontology captures general features of basic contextual entities and the associated subdomains capture a collection of domain specific ontologies and their features. Variation CONON+CYC Upper Ontology.
	URI	Not available.
	Status	inactive and not available.
SOUPA	**Author (Year)**	H. Chen, T. Finin, A. Joshi (2005)
	Publication	The SOUPA Ontology for Pervasive Computing

	Description	The SOUPA ontology is expressed using the Web Ontology Language OWL and includes modular component vocabularies to represent intelligent agents with associated beliefs, desire, and intentions, time, space, events, user pro- files, actions, and policies for security and privacy.
	URI	`http://cobra.umbc.edu/ont/soupa-ont.tar.gz`
	Status	active and available.
MySAM ontology	**Author (Year)**	Oana Bucur, Philippe Beaune, Olivier Boissier (2009)
	Publication	Defining and Modeling Context in a Multi-Agent Systems Architecture.
	Description	MySAM context ontology propose a concept called *ContextAttribute*) that will always contain the same kind of information: the name of the attribute, the type of needed parameters (entities) for the instantiation, the Va (values domain).Starting from this class, each context attribute is characterized by these properties, with different restrictions: "RoleOfPersonInGroup" will need a *Person* and a *Group* as parameters and will give a *Role* when instantiated. The class "Entity" is the super class of all concepts. *Person*, *Group*, *Room*, *Activity*, etc. are subclasses of *Entity*. The class *Entity* is the super class of all concepts that characterize the domain, in MySAM, such concepts are: *Person*, *Activity*, *Agenda*, *Group*, *Role*, etc.).
	URI	Not available.
	Status	inactive and not available.

3.7 Summary discussion

As we have seen in this chapter, the design and implementation of artificial social companions are still based on closed architectures, thus being operational only in specific contexts of usage, equipment and data. Furthermore, most of current perceptual models neglect the context and the spatial relation during the perception process. Associated to these, current systems use static and implicit representations for context preventing dynamic system adaptation through information sharing and learning.

Summarizing, based on the information from our survey and also in-line with our works in [163–166], we found that implementing realistic interaction workflows on artificial social companions (e.g. virtual assistants, social robots) is extremely challenging due to the limitations imposed by the selected system architecture, availability of perception features that operate correctly in multiple environments (i.e. contexts). On the other hand, the approach used to capture the user's perspective regarding the expected behaviour of the agent is by describing different situations in the form of user scenarios.

Furthermore, recalling the words of Oliver Brdiczka et. al in [59, 60] "... *computerized spaces and their devices require situational information, to respond correctly to human activity. In order to become context aware, computer systems must thus maintain a model describing the environment, its occupants, and their activities.* ... ". However, the typical approach to provide contextual information to the application is by manually defining the context models according to their end goal and taking into account particular user needs. This approach does not render the "real world" in the sense that it fails if the user needs evolution over time is taken into account. Brdiczka [59, 60] continued and considered that "... *New activities and scenarios emerge in a smart environment, and others disappear. New services must be integrated into the environment, whereas obsolete services should be deleted. Thus, a fixed context model is not sufficient.* ...". Moreover, long-term maintenance, required by common approaches, have a negative economical impact to the user, i.e. having an expert periodically adjusting the system according user needs would be expensive. Thus, the research for more intelligent, self-learning and self-adaptable systems is justified facing the inefficiency of the common approaches.

Taking into consideration with the above, we identified the development of a context-based human-interaction framework as an open challenge. Furthermore, this challenge cannot be left unanswered as it address a key aspect to improve user acceptance regarding social robots and smart assistants.

Chapter 4

Building a Virtual Artificial Social Companion

This chapter describes the implementation and evaluation of a Virtual ASC, which served as baseline for our study. The semi-autonomous agent was developed to simulate a daily life companion for older adults and its assessment and evaluation was conducted in relevant environments (i.e. home environment of older adults). The agent operates on a all-in-one device, which provides the user interfaces and interaction modalities needed to realize face-to-face interactions with older adults, in a number of daily life scenarios. For example, it simplifies visualization and query of information related to the user's daily schedule, reminders and social events and helps older adults to connect with others by exchanging messages. We start from commonly agreed requirements, which resulted from following co-design methods. These requirements include some relevant functional aspects for designing an agent that must maintain a long-lasting interaction with the user. For example, the agent should be as natural and believable as possible, it should simulate a real-life human companion; it must support natural speech interaction and it must be capable of recognizing the user's emotional state and delivering appropriately tailored empathetic feedback, including emotional facial expressions. However, the reporting of these types of needs and wishes may result in users getting frustrated and disbelief technology if the agent cannot fully met their initial expectations. Therefore, it becomes very important to have a close and careful follow up by a human actor (e.g. caregiver) that can oversee the introduction of this type of technology for the particular target group of older adults.

4.1 The CaMeLi Daily Life Companion for Older Adults

In the scope of the European project CaMeLi[30] we investigated the use of Virtual Artificial Social Companions (ASC) as long-term assistive companions for older adults (aged 65 and above). Specifically, we developed a semi-autonomous agent to motivate older adults self-manage their care and remain socially and physically active [5]. The agent is seamlessly integrated into the living environment and at any point in time, a user can request assistance. The agent delivers appropriately tailored empathetic feedback, in a similar way a real-life human companion would do.

4.1.1 Development methodology

The development methodology adopted in CaMeLi followed an iterative process, very much aligned with Agile software development methods. This process started with the identification of user needs, during interviews from caregivers and researchers together with potential end-users of the system. The user needs were translated into technical requirements using use cases description and Unified Modeling Language (UML) diagrams to express the expected features for the ASC. The following step was the implementation of the features according the technical specifications and system integration. Afterwards, the system was tested and validated by the end-users, who could provide feedback and request changes for the implemented features. The whole process is depicted in Figure 4.1.

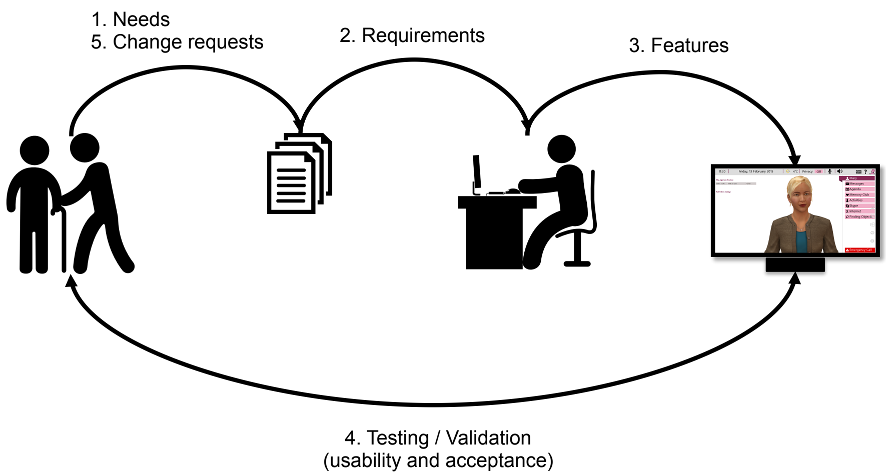


Figure 4.1: CaMeLi iterative development methodology.

[30] www.cameli.eu

4.1.2 Involvement of end-users

To understand how to design the agent, we conducted two user-centered design studies in collaboration with two care organizations which provide daily assistance to elderly in the Netherlands and Switzerland. The studies were based on focus groups, individual interviews and paper based surveys with older adults with age above 65 (N= 20), professional caregivers (N= 12) and psychologists specialized in the aging process (N=2). The study was performed on three different user environments: 1) Independent elderly living at home in an urban setting, 2) elderly home and 3) care apartments. The detailed methodology and results on how we carried out these initial part of the studies were described and published in [167].

4.1.3 Example Scenarios

Based on the design studies, we defined several scenarios that illustrate how the CaMeLi agent can interact with older adults in a daily life context. In all the scenarios, the agent expresses coherent behavioural and emotional responses that can be interpreted by the users as indicative of a natural social personality.

- Scenario 1 - Self-management of daily activities: To help elderly manage their daily activities the companion maintains a digital version of their personal agenda, assists to keep it up to date (e.g., enter or modify social activities, medical appointments) and issues appropriate reminders during the day.
- Scenario 2 - Safety: The agent monitors the elderly and recognizes abrupt human body motions that indicate instability or a sudden fall or a call for help, based on a predefined voice command. In the case of an emergency, the agent initiates a dialog to acknowledge the detection and reassure the user and if necessary, automatically dispatches a phone call to a designated caregiver.
- Scenario 3 - Guidance for household activities: Since elderly are prone to forgetfulness, the agent offers assistance for locating objects around the house employing real-time vision-based detection of previously learned objects, whenever possible (i.e., when an object is in the field of view of the camera).
- Scenario 4 - Wellness and Leisure: The agent knows the user's personal interests and proposes leisure activities or guides users through brain wellness activities (i.e., guided meditation and relaxation exercises) and a program designed to teach concrete strategies to improve the performance of prospective memory to apply these skills in everyday life.

- Scenario 5 - Communication and socialization: To facilitate communication with friends and caregivers the agent guides the elderly through the use of a private message exchange system as well as Skype communication and Internet navigation.

These studies resulted in a list of needs and requirements as expressed by older adults. The complete list of the expected features to include in an ASC is provided in table 4.1. For example, reminder functions, brain training or other games, daily agenda, fall detection and call for help were identified as needs in all three user settings, whereas other services appear only in one or two user settings.

Table 4.1: User surveys and list of services

Services	Setting1	Setting2	Setting3
Agenda for daily living activities (wake up)	-	+	+
Breakfast reminder	-	+	-
Medication reminder	+	+	-
Daily schedule reminder	+	+	+
Reminder prior to appointments or activities	+	+	+
Add agenda items themselves	+	+	+
Program of activities nearby or in the city	+	-	+
Notify carer or family member when running late or cancelling a planned event	-	+	+
Messages from the system	-	+	-
Do not forget things when going out	+	-	-
Fall detection and call for help	+	+	+
Call for help in case of an emergency	+	+	+
Noise detection (for safety purposes) and call for help	+	+	+
Safety alarm follow-up	+	+	+
Behaviour analysis and motivation functionality	+	+	+
Emotion recognition by means of facial expression and speech analysis	+	+	+
Brain training or other games	+	+	+
Playing recorded relaxations, meditations	+	-	-
Stimulation for physical exercise (Yoga or going out)	+	+	+
Finding things detected by the camera	-	+	+
Object storage memory	-	+	-
Dinner menu of the restaurant (read out)	-	+	-

Shopping list	+	-	-
Skype functionality	+	+	+
Communication with friends via CaMeLi system messages	+	+	+
Communication with friends and family via text messages (sms)	+	+	-
Retrieve information online	+	-	-
Bus/public transport time tables and routes	+	-	-

4.1.4 The Interaction Process

In CaMeLi, users interact with the ASC using a multi-modal interface that includes a touch-based Graphical User Interface (GUI) and Automatic Speech Recognition (ASR). In spite of both modalities can be used to complement each other, we mapped all graphical artefacts associated with actions to a voice command. In this way, we simplified the learning process and allow a broader group of users to use the system. This means that, by providing a redundant interaction mechanism we allow people with different capabilities to still use the system. For example, visually impaired people or illiterate can use speech to activate the same functionalities as another user that might not be able to express himself in word can, by touching virtual buttons. On the other hand, although our agent does not perform at a human level conversational skills, the natural speech interaction interface was based on human-to-human dialogues from scenarios of daily life interaction. While the dialogues are highly situation and task dependent, we engaged in the co-design process involving both older adults and caregivers to define dialogue scripts that would cover the assistive care services mentioned above, as illustrated in Figure 4.2.

4.1.5 Implementation

The CaMeLi agent's architecture resembles what has been proposed as a reference architecture for Embodied Conversation Agent (ECA) [168]. As outlined in figure 4.3, the architecture consists of three key components responsible for perception, decision making, and synthesis. The well-orchestrated interplay of these components gives rise to the human-like cognitive capabilities of the agent. figure 4.3 outlines the components in more detail.

Figure 4.2: Example of an interaction scenario. Interaction is started by the user, who initiates the conversation with the agent, following a question-answer sequence.

```
User: Avatar, show me the agenda for today.

Agent: On [activityDate], from [activityStartTime] to
[activityEndTime], the activity [activityName] will take place
in the location [activityLocation].

User:  Ok, please invite my friend [friendName] for [activityName].

Agent:  I sent an invitation message of [activityName] for your
friend [friendName].

User:  Thank you!

Agent (Facial expression [Joy]): You are welcome. I will notify you
receive a message from your friend.
```

4.1.6 Hardware Setup

The hardware requirements for the agent to operate properly include a all-in-one stationary computer (e.g. we selected Lenovo ThinkCentre Edge 93z All-in-One), that can be mounted on some support at a height ranging from 80 to 120 cm, allowing the user to interact from a standing or sitting position. Moreover, the preference for an all-in-one system was also given by built-in integration of a set of hardware components that are required to provide multi-modal interaction. In particular, a touch-screen and graphical display, a built-in microphone, a built-in video camera and speakers. Additionally, we used a Microsoft Kinect device to acquire depth (RGB-D) visual data. On the particular case of using this device for Human Behaviour and Environment Analysis, we realized that the appropriate mounting height should be around 130 cm. Given that this device has a viewing angle approximately of 60° vertical by 70° horizontal, we ensured a minimal field of view in which we could detect standing and laying down persons and detecting objects at heights of up to 90 cm (e.g. small room table).

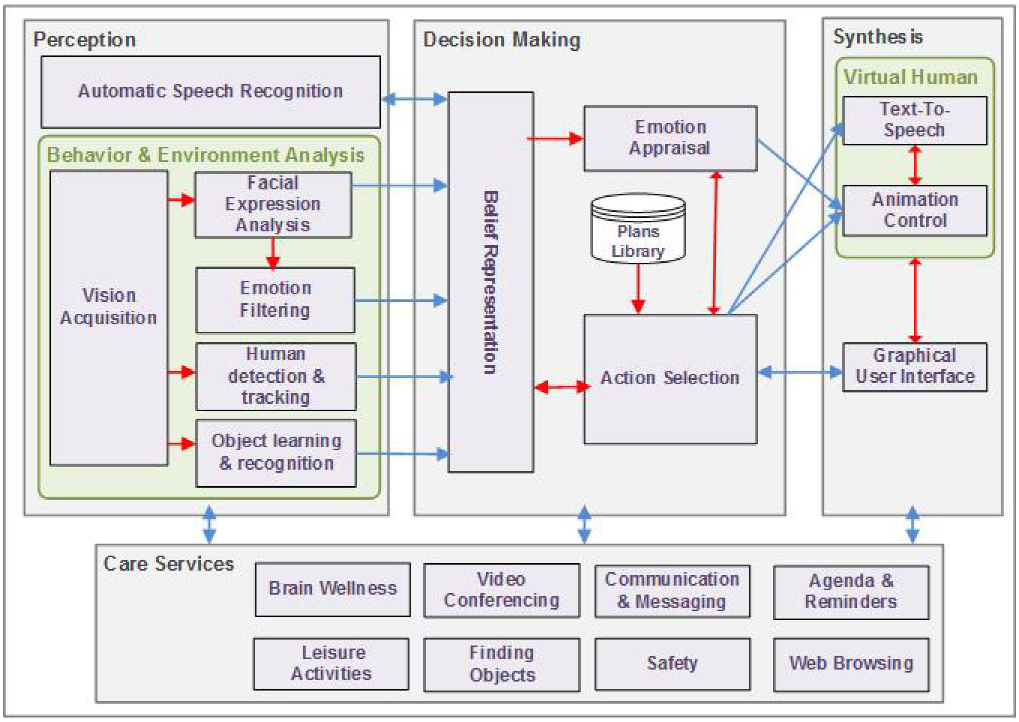


Figure 4.3: The system architecture of the CaMeLi system

4.1.7 Perception Components

The Perception components collect sensory information from different sources e.g., speech recognizer (Automatic Speech Recognition), RGB cameras tracking the user and depth cameras that recognize the environment (Vision Acquisition). The perceived information, such as the user's speech commands and perceived emotion, directly triggers reactive responses from the agent or is send to the Decision Making components for further processing.

Automatic Speech Recognition The ASC developed in the scope of CaMeLi can support four languages: English, Portuguese, French and Dutch. In order to activate the different functionalities of the system, we created custom grammar files (i.e., language models) for each language. In the grammar files, we indicate what utterances are recognized. Also, it includes the probability distributions associated with sequences of words. This was done to constrain (i.e. defining rules) the search among alternative word hypotheses during speech recognition.

Human Behaviour and Environment Analysis The agent includes human-behaviour and environment analysis features, built on top of vision-based approaches for human body detection and facial expression analysis, and objects recognition. Regarding human detection and tracking we followed the approach presented in [169]. This uses a probabilistic approach to classify human behaviour based on movement features like the height variations, velocity, and acceleration of limbs. We acquired visual data (RGB-D) using Microsoft Kinect and the classifiers were implemented as a Bayesian Network (BN). The main interest of using this component was to recognize abrupt body motions that could relate to a sudden fall. The Facial Expression Analysis module extends the FaceReader [170]. FaceReader is a commercial product from Noldus. It was developed with the objective of emotion recognition in laboratory settings; to be used in behaviour analysis studies. Using this component, our ECA can detect the six basic emotions described by Ekman [171] (happy, sad, angry, surprised, scared, disgusted) and a neutral state. The approach used for environment analysis focus mostly on Object Recognition. We implemented our approach based on the Global Hypothesis Verification, proposed in [172], using shape descriptors obtained from different views of the object, represented in Point Cloud Data (PCD) [173]. This approach is especially useful on a household, which we considered to be an unstructured and uncontrolled environment; given it has the inherent ability to detect significantly occluded objects without increasing the number of false positives.

4.1.8 Behaviour Synthesis Components

The behaviour synthesis components are in charge of generating intelligible verbal, gestural and facial expressions and combining them into a continuous flow of human-like multi-modal behaviours. These components are built on top of SmartBody [174]. SmartBody is responsible for generating the gestures and facial expressions of the virtual companion and is controlled by the CaMeLi's framework through the Behavioural Markup Language (BML). Consequently, it enables easy control of each Action Unit (AU) to achieve particular emotions by describing facial expressions of the virtual human using the Facial Action Coding System (FACS). This simplifies the process of describing the physical realization of behaviours such as speech and facial expressions and the synchronization constraints between these behaviours. A set of pre-set emotional expression configurations is pre-programmed in the CaMeLi framework. Furthermore, SmartBody implements lip resynchronisation based on the visemes generated by the speech synthesis engine. In addition to generating gestures and facial expressions, the animation module also generates secondary behaviours to make the agent appear more lifelike (e.g. eye blink and breathe). In our implementation Microsoft's native text-to-speech engine is employed for French and English, while CereProc was utilised for the Dutch language. Figure 4.4 illustrates the 3D virtual human, which closely simulates human conversational behaviour through the use of synthesized voice and synchronized non-verbal behaviour such as head nods and facial expressions.

4.1.9 Graphical User Interface

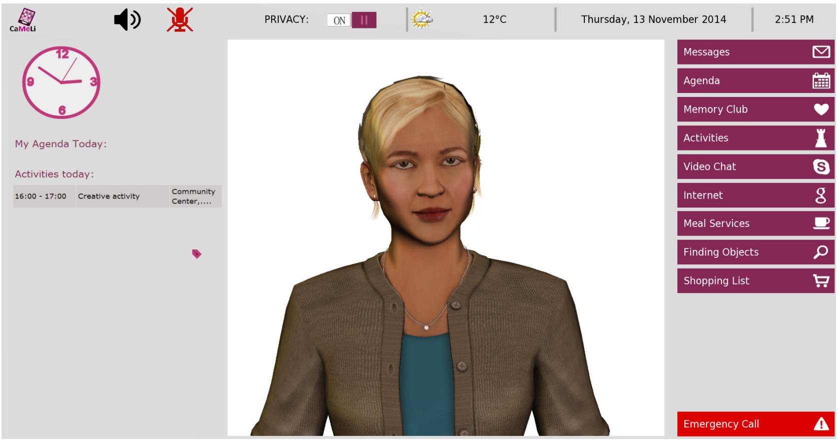


Figure 4.4: The agent and the Graphical User Interface.

In addition to the natural speech interaction, the users can issue commands to the agent using buttons and elements on the GUI. The design of the GUI (Figure 4.4) took into consideration relevant standards of User Interface Design (e.g. such as defined by W3C) and the main input was results of the iterative process of the co-design with the end-users. The later, allowed us to identify clear requirements and to collect relevant feedback from older adults. The result was a GUI with three main areas: (1) on the right-hand side there is a menu that provides access to the care services (note the big and clearly spaced buttons to facilitate their activation); (2) on the top, we put a status bar that provides information about current time and date, the weather forecast, and notifications. Also, on the right-hand side of the status bar, the visible icons inform the user about the agent's status (e.g. enabled/disabled, currently talking/listening for commands). Additional buttons are provided to enable the on-screen keyboard and the help menu and to exit the system; (3) the central area of the screen is reserved to display content. In the home screen, the avatar will appear maximized. When accessing to a service, the graphical interface for that service appears in the center and the avatar appears in a smaller version at the bottom of the menu column. This design ensures that the users can interact with the touch-based interface intuitively after a short introduction.

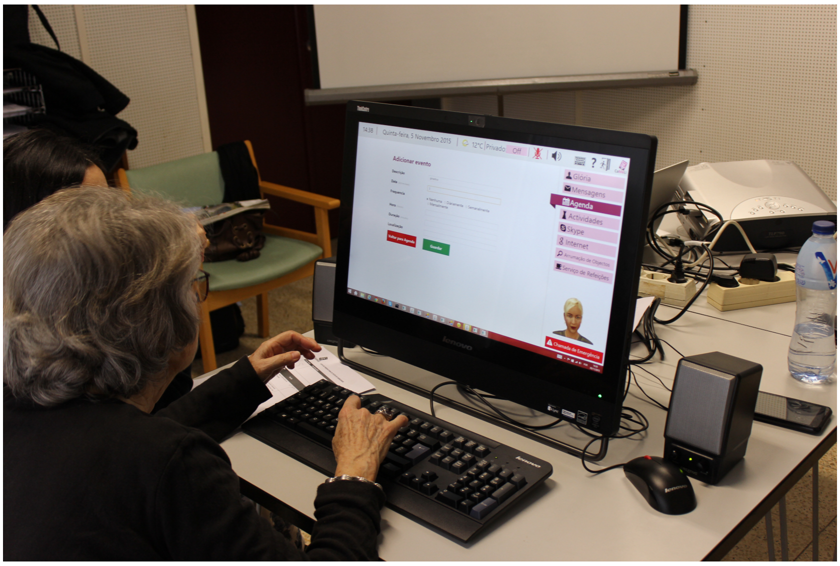


Figure 4.5: User Evaluation is one of the pilot settings.

4.2 Agent Evaluation Study

One important matter in designing and implementing ASCs in general, is to understand the impact they have in the daily life of the users for who they are designed for. In this sense, system evaluation and assessment is vital to understand if the system (i.e. agent) answers the user needs and if its behaviour met user expectations. Consequently, we conducted a longitudinal evaluation study to assess the effects of the agent in the daily life of older adults. Our main goal was to verify empirically how our target user group would interact with the Artificial Social Companion Avatar agent, when installed in their private space. Also, we wanted to understand the subjective opinions users had in terms of acceptance, perceived usability and usefulness while operating a relevant environment. This means, operating in a real scenario where various types of interaction and tasks could occur during daily life activities.

4.2.1 Setting

The study took place in three distinct test-beds from three European countries. In Switzerland, where elderly were living alone in apartments of an assisted living complex; in the Netherlands where participants were living alone in independent apartments; and in Portugal, where elderly living independently stay in a home care during the day period (e.g. Figure 4.5).

4.2.2 Participants

We considered the target group to be healthy older adults that might have light physical or light mental health problems, who live alone at home and want to be helped or stimulated to carry out their daily activities. The inclusion criteria for older adults to be included in the study were:

- be able to put glasses off when necessary for the system;
- should not use a wheelchair inside;
- living alone.

We estimate the approximate number of population with this characteristics of the inclusion criteria is between 5 and 7 Million people, for the sum of the three settings. The total number of users involved of 36 participants, which correspond for a confidence level of 95% and confidence interval of 16.33%. The distribution per respective sites is: 13 (Netherlands), 11 (Switzerland), 12 (Portugal).

The characteristics of the users involved in the evaluation and assessment of the system is shown in table 4.2.

4.2.3 Evaluation phases

The total duration of the this evaluation study was 12 weeks. We divided evaluation into three parts (see Figure 4.6). First, in the Introduction phase (T0), we carried out the baseline measurements, system installation and users' training. Regarding the baseline measurements, we adopted the World Health Organization Quality of Life (WHOQOL) questionnaire and conducted semi-structured interviews. Second, after 4 weeks, in the intermediate evaluation (T1), we used the System Usability Scale (SUS) questionnaire to gather user's impressions about the system. This phase was important to collect data during the period were the users might have experience the novelty effect, which would be useful to compare with an equivalent assessment after a longer period of use. Third, in the final evaluation (T2), we applied both of the previous questionnaires and conducted semi-structured interviews. This last phase was performed after at the end of the 12 week period of daily basis interaction with the system.

We conducted semi-structured interviews with open questions about the participants' level of autonomy and quality of life before (T0) and after using the system (T1, T2). Also, we collected their expectations (T0) and their conclusions (T2) about the system's impact on their daily life, regarding autonomy, daily life organization, activity,

Table 4.2: User baseline measurements

Topic	Netherlands	Switzerland	Portugal
Participants	N=13	N=7 (4 drop-outs)	N=10 (2 drop-outs)
Average age	79,4	76,4	79,9
Experience computers (1-5)	3,1	4	1
Experience tablets (1-5)	3,8	2,6	1
Experience avatars (1-5)	1,4	1,7	1
Satisfaction daily life	100%	87,5%	NA
Memory (1-10)	6,9	Qualitative	NA
Quality of life (1-10)	7,5	Qualitative	NA
Expectation of becoming more autonomous with the system (1-5)	2,6	2,9	NA
Expectation of becoming more organized with the system (1-5)	2,3	3,4	NA
Expectation of becoming more active with the system (1-5)	1,8	2,5	NA
Expectation of improving memory with the system (1-5)	3,0	4,1	NA

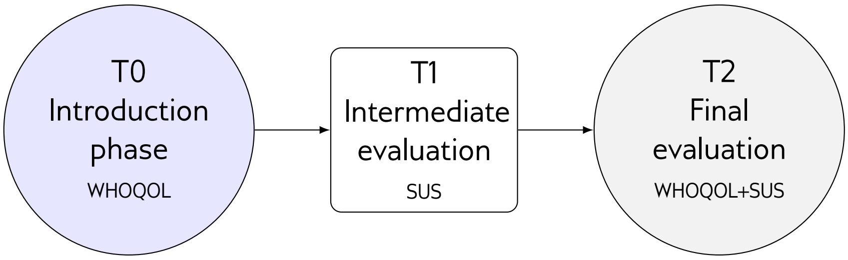


Figure 4.6: The three phases of the evaluation study

and memory. Based on the thinking aloud protocol [88], we invited participants to say what came into their mind, what they were looking at, thinking, doing, and feeling as they performed tasks with the agent (T0, T1, T2). Finally, participants were invited to keep a diary where they reported remarks, questions, and ideas throughout the study.

4.2.4 User Acceptance

Overall, users were positive regards accepting the agent in their households. However, after sometime interacting with the system, passed the novelty effect, we observed, in the final evaluation (T2), a decrease in acceptance. The most reported reason was that the agent did not fully reach users' original expectations. Among the main reasons for this we note the mismatched expectations related with verbal communication capabilities. In fact, the majority of the users in the target group, on facing a human-like character expected a more natural interaction in terms of speech dialogues. Although, we could not find an ASR system that could fulfill such expectation. Participants easily got frustrated after a few unexpected verbal behaviour by the agent. On the other hand, we noticed that users from the target groups face some challenges regarding training the interaction with the agent. This led to a higher number of repetitions that desirable. From these consideration, we draw two main conclusions.

1. *An ideal solution would be to have more flexibility and variety in the speech commands.*

2. *it is of utmost importance that all the interaction components run as robustly as possible, are fault-tolerant, and support repair mechanisms.*

4.3 Assessing Usability

Usability was assessed using the System Usability Scale (SUS). This instrument establishes that a score between 50-70 is positive, where above 68 is considered as “above average”. Besides, a total SUS score below 70 is problematic since this value seems to be the threshold for a “good” usability. The obtained results are summarized in Figure 4.7.

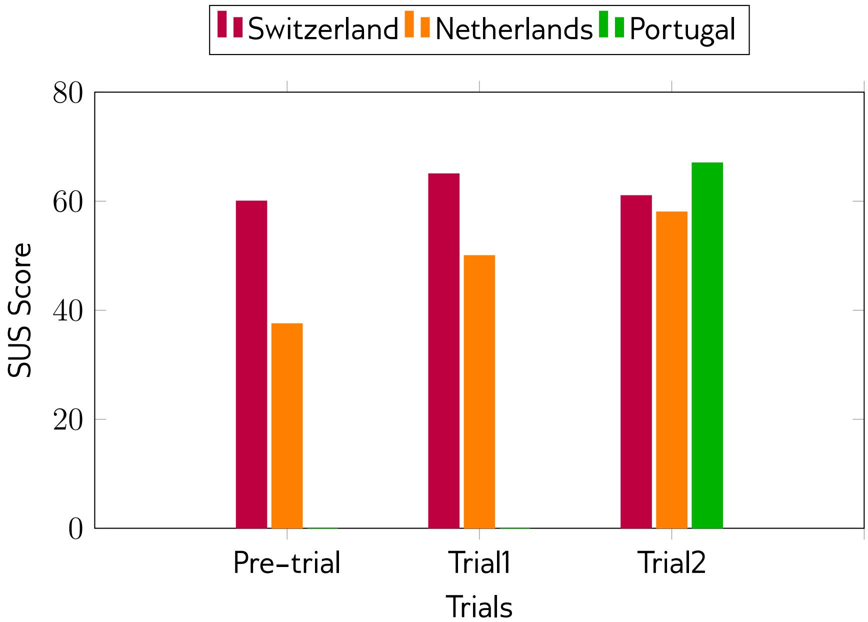


Figure 4.7: SUS results for the three settings.

The obtained results refer to the answers for the SUS questionnaire in the three distinct periods mentioned previously in section 4.2.3. Hence, data was collected at the beginning of the study, at the fourth week after starting using the system and finally at twelfth week. Worth to note that for the Portuguese site, the system was tested only during the phase between the eighth and twelfth week of operation of the system. This fact was motivated not only to compensate the dropouts from the other two trial sites, but also to include an external group of users, who were not directly involved during development of the ASC, and allowed us to understand the perspective from users unfamiliar with the system.

The results show average SUS scores of each setting were: 52 for the Netherlands, 62.2 for Switzerland and 67 for Portugal. The combined average score was 60.4 (range from 37.5-80). Thus, overall we conclude that the interaction with the system was perceived as “average" positive. However, when considering these results individually we observe a high variation between settings. While the system was perceived as positive in all settings, the users in Portugal were more positive towards the system than the users from Switzerland and Netherlands (i.e. the less positive towards the system). At this point, we must argue that cultural differences may have influenced this result. Recall from the user characteristics, in table 4.2 that users in the Netherlands were more experienced with technology than the rest. This fact may be enough reason for them to be more demanding and have higher expectation towards novel technology.

4.3.1 Perceived Usefulness

We asked all participants to assess the usefulness of the digital services provided with the agent. Based on the qualitative and quantitative results, we observed a significant variability, between the three settings and between users. In Switzerland, the users favoured more services for memory training, agenda, simplified Internet browsing and Skype integration. In the Netherlands, the most popular services were agenda, social activities, and message exchange between friends. Finally in Portugal, the more positive services were the agenda, skype connection and internet browsing. Summarizing, these findings revealed that usefulness is tightly coupled with the specific context of each older adults. This means, it depend highly on each personal care needs and lifestyle choices. Therefore, to remain its usefulness in the long-run, the system should be capable to learn and adapt to its users' context (i.e. personal needs, and social and physical situation).

4.3.2 Ethical considerations

During the development of CaMeLi, we looked into several ethical issues that cannot be disregarded when using socially intelligent ASCs. Given the specificities associated with our target group, first and foremost, *any artificial agent should not be obtrusive or stigmatizing for the users, nor restrict their privacy*. In fact, during the study, we registered some of the worries that cross older adults minds, and any other person, regarding the installation of cameras and the associated uncertainty about whether or not their interactions with the agent would be recorded. Additionally, it was uncomfortable for the user when, during the installation of the agent, we found any constrain in the household layout that required to rearrange the space for the system to work properly. For example, insufficient space or inappropriate location for mounting the computer and additional devices at the optimal positions (functional for the system vs. comfortable for the user). From this study, we learned that *introducing hardware in the household is inevitable, but ideally, no recorded data should be stored, and all the devices should be integrated into the existing furnishings*. Additionally, following the good practices of interaction design, *it is essential that older adults are not deceived into thinking they are interacting with a agent that is capable of doing things that only humans can do in an interaction*. This means, it must be clearly explained to the user that the agent is a computer-driven character with limited capabilities.

4.4 Summary discussion

In this chapter, we described the implementation and evaluation of a Virtual ASC. Such type of agents are seen as promising solutions to help facing the challenges imposed by the ageing population. The purpose of this experiment was to conduct a 12-week evaluation where we could observe how older adults interact with an artificial social companion, which would serve as baseline for our consequent work. The study was conducted in three distinct settings in three European countries (Netherlands, Switzerland and Portugal) and involved 36 participants. This sample size was calculated based on the approximate population that met the inclusion criteria for the study and had a confidence level of 95% and a confidence interval of 16.6%. We followed a qualitative and quantitative methodology, using standardized assessment scales WHOQOL and SUS and looked at the users interacting in daily-life scenarios with the agent.

Results confirmed that overall users are positive regarding accepting novel technology, in particular regarding interactive systems. Usability and perceived usefulness was also positive. However, unmet expectations in terms of level and maturity of interaction resulted in borderline average results for usability, as shown in the results from SUS score. Nevertheless, we noticed some differences on how users from different settings responded to the same technology. Hence, cultural, socio-economical and personal preferences (i.e. context) can influence the opinions towards this type of technological approaches. The results suggest that older adults with lower digital literacy are more optimistic regarding ASC, when compared with older adults more familiar with Information and Communication Technologies (ICT).

Additionally, the evidences from this study revealed that the (cognitively impaired) older adults found it difficult to gain insights into the agent's possibilities and cope with its limitations. Age related cognitive degradation, specifically age related memory changes and their effects on learning, make older adults feel mentally stressed when interacting with new technologies. Human-machine interactions are in principal cognitive tasks that require recalling, recognizing processing and storing information, as well as problem solving abilities. Unexpected behaviour (e.g. Speech recognition errors) or technical faults (e.g. component malfunction) during the interaction with the agent were proven to be highly demoralizing for the older adults.

They rapidly increased their stress levels, and generated feelings of reduced self-confidence and insecurity. After realizing and experiencing the systems fallibility, the users tended to establish a distinguished mistrust concerning the companions' competencies and seem reluctant to prolong their interaction. For a system that autonomously works as an assistive companion at home, it is of utmost importance that all the components run as robustly as possible and are fault-tolerant, especially since an elderly user is involved. For example, the dialogue manager component can be enhanced to enable repair mechanisms to support reaction to Automatic Speech Recognition misunderstanding, asking for clarification, etc. At the same time, it is essential that older adults are not deceived into thinking they are interacting with a companion that is capable of doing things that only humans can do in an interaction. It must be clear that the companion is merely a computer-driven character with limited capabilities. By minimizing the discrepancy with the user's expectations about the companion, is possible to minimize the risk that the system is less accepted and not trusted.

Chapter 5

Improving Artificial Social Companions with Context-based Human-Machine Interaction Framework

In the previous chapters, we presented the foundations of Interaction Design and covered the relevant aspects of Artificial Social Companions (ASC). Based on these chapters, we note that for designing ASC we need to understand the user and the context when interacting with the product. However, we found that implementing realistic interaction workflows on such agents is extremely challenging due to the limitations imposed by current system architectures and availability of perception features that operate correctly in multiple environments (i.e. contexts).

Nevertheless, achieving situation, activity and goal awareness are vital to enhance interaction between the user and the ASC. Therefore, these agents must integrate the necessary context and interaction information models. This means, its perception and knowledge representation need to take into account how to represent a context model. To answer the aforementioned challenge, we looked for a strategy that could translate user's descriptions into the knowledge representation; and then use this knowledge during agent's operation in such way it could guarantee a functionality even when context changes. This goal framed our understanding of ASC to be tightly coupled with their capacity of self-developing over time. This means, we consider that an ASC must be capable to "learn new things" during its "life cycle"; at least, when interacting with new users or operating in new context. It should be capable to include new information into its knowledge base and use it later when needed. To a certain point, this thought justified we would look to current cognitive development theories searching for an inspiration on how to design our framework in such a way it could allow the artificial agent to develop similarly to the human mind, or at least to take into consideration some basic factors (e.g. scalability of the knowledge representation, representation of the user model, representation of context model).

In this chapter, we will cover the main steps associated with the implementation of the propose framework, which are depicted in Figure 5.1.

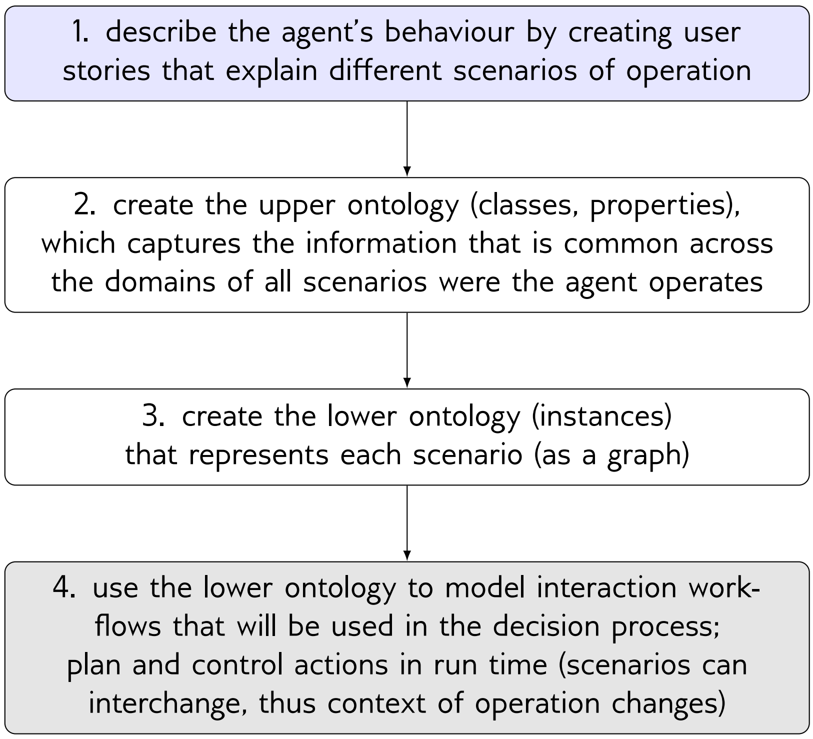


Figure 5.1: Diagram for representing the framework steps

5.1 Adjusting development methodology

Before we explain in more detail each of the implementation steps, we must look first to the development methodology used to build ASCs and understand how it should be adjusted to allow the implementation of the proposing framework. Taking into consideration all lessons learned during from the previous chapter, we felt the typical development cycles took too long to iterate. This situation contributed to user's frustration and uninterested in technology. As an initial step of our framework, we propose an adjustment in the way needs, requirements and features would be described in the system, which would allow a more flexible and adaptable way of operation. The overall idea, as depicted in Figure 5.2, is to collect user's expectations into descriptive scenarios, like it will explained in the next section, and translate these descriptions into a format that can be used by both the developer and the machine. In parallel to the description of the scenarios, developers would describe the features and associate those with the corresponding scenarios where they should be executed. Such description should be followed by features implementation that should be also deployed in the system. The overall goal is to have a complete knowledge representation of scenarios, features, the context of operation, and have a periodic mechanism that supervises what features should be executed when a specific context is observed. Finally, the remaining steps to close the development cycle should be still guaranteed to ensure end-users testing and validation, which allow to gather feedback and request changes.

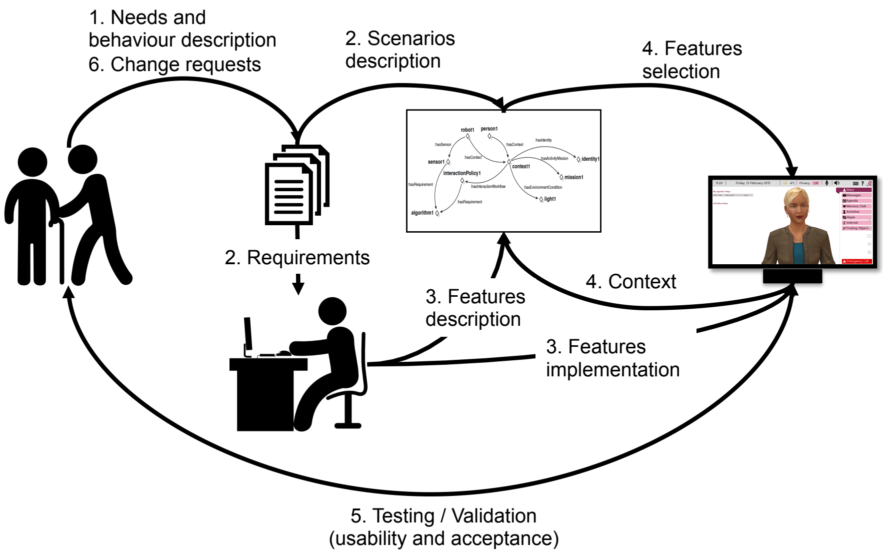


Figure 5.2: Adjustments to iterative development methodology.

5.2 Describing the agent's behaviour

Inspired on the *Behaviour Driven Development* methodology [175], we describe the agent's behaviour by creating user stories that explain different contexts of operation for some relevant application scenario. First, we should start gathering some relevant information by interviewing directly the end-users. After we have collected their initial expectations we should describe this information in generic formats, for example using *Personas* and *User Scenarios*. Having described users' initial needs and requirements, we can refine these into a more suitable format to be used in the agent. At this stage, we will create the user stories. This is done by breaking down Personas and User Scenarios into a more detailed description of the expected behaviour for each feature of the agent, while operating in different contexts. We use *Gherkin Scenarios* (see Figure 5.3) to describe the agent's behaviour with great detail.

Figure 5.3: Gherkin Scenario pattern

```
Feature: <feature title>

  In order to <goal>
  As a <actor role>
  I need to <action>

  Background:
    Given <pre-condition1>
    And <pre-condition2>
    ...
    And <pre-conditionN>

  Scenario: <scenario title - context1>
    Given <context1_feature1>
    And <context1_feature2>
    ...
    And <context1_featureN>
    When <control action1>
    And <control action2>
    ...
    And <control actionN>
    Then <outcome>
```

5.3 Creating the Upper Ontology

In our approach, motivated by the conclusions from the previous chapter, we decided to represent knowledge about concepts and their relationships by using ontologies. This representation format will allow us to capture the types of knowledge required to fully represent the cognitive model for an artificial social companion, which include concepts related with person, environment, physical interaction, social interaction, machine/robot interaction and algorithms. In this section, we focus on the Knowledge model for our framework. For additional details regarding the process involved in Ontology Development, please refer to Appendix A.

5.3.1 Knowledge model

Now that we have considered how to represent knowledge, we can look into the knowledge model (Figure 5.4), which intends to capture the relevant information involved in the Human-Machine Interaction (HMI) process. We define the upper ontology for this framework based on four main entities *Context*, *Human*, *Machine* and *Interaction*. From these, we can define other entities as associated sub-classes and establish a relationship between entities that entails the semantics of their association. This model resulted from several iterations from the initial proposal in [166] and it is available to be incorporated or extended by other representations[31].

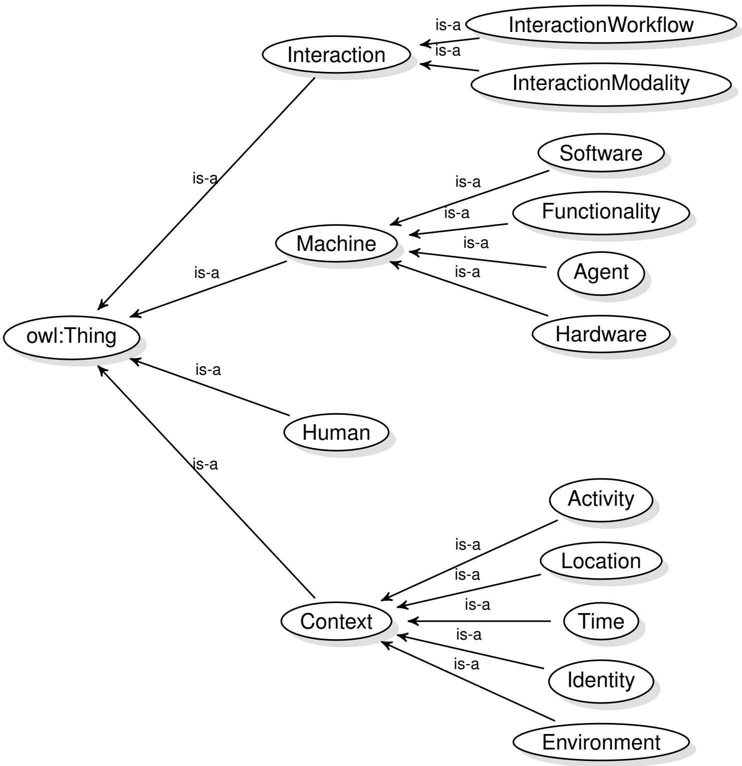


Figure 5.4: Knowledge model used in the Context-Based Human-Robot Interaction Framework.

The entities we defined and their relationships make it possible to represent which components of the system are involved in the interaction process at each time. Besides defining the classes taxonomy, we define Object Properties (OP) and Data Properties (DP) that will establish the relationships between the individuals of each class. Our current model includes the following object properties:

[31] `http://www.contextawarerobotics.org/aurora/kr/im-aurora.owl`

- (OP) hasActivityMission
- (OP) hasActuator (Domain: robot / Range: actuator)
- (OP) hasContext (transitive)
- (OP) hasEnvironmentCondition
- (OP) hasIdentity
- (OP) hasInteraction
- (OP) hasInteractionWorkflow
- (OP) hasRequirement (transitive)
- (OP) hasSensor (Domain: robot / Range: sensor)
- (OP) isActivityMissionOf (inverse of hasActivityMission)
- (OP) isEnvironmentConditionOf (inverse of hasEnvironmentCondition)
- (OP) isInteractionWorkflowOf (inverse of hasInteractionWorkflow)
- (DP) policyGraph (string)

Context Ontology

Looking back to the definitions discussed in Section 2.3.1, we can identify some key concepts that can be associated with some branches of mathematics (e.g., set theory, vector spaces, constraint optimization). This is particularly true when the majority of the definitions refer to context as a *set of circumstances* that act like a *set of constraints* on the system's behavior (i.e., output). On the other hand, the concept of situation appears associated with context when this is defined as *elements of the situation that should impact behavior*. We propose to extend on previous definitions as follows:

> *Context will be defined as the set of information which constrain the performance of an agent while attempting to execute a desired behaviour. In spite of the characteristics of that agent, any set of information will be only considered to be context if it anticipates how the agent should behave when that information is present.*

This definition describes context as a *set* of information that is relevant for decision making, which will result in a specific behavior. After the decision is made, for a behavior to be performed, it is implied that the context is maintained.

In our context model, we took into account the most used context entities by context aware applications. Therefore, in spite of those identified in literature - the four main primary context entities (i.e., location, time, identity, environment) and the six complimentary context entities (i.e., social setting, network, season, history, task/activity, device) - only five may be considered as "workable" context entities.

We concluded from our survey that the most relevant context entities are *Location*, *Identity*, *Time*, *Environment* and *Activity*.

Predicted versus observed contexts The definition of the context that influences the execution of tasks by an autonomous robot may include, for the sake of improved autonomy, observable quantities that could extend the definition of context beyond the typical time, location, task, etc. This would permit to define modes of operation given observable and thus dynamic contexts, besides the static ones. Including a true context awareness in the decision process enables the adaptation of the robot to changing work conditions that may appear at unpredictable times and places.

Following this assumption, we can extend the Context Ontology according to the model depicted in Figure 5.5.

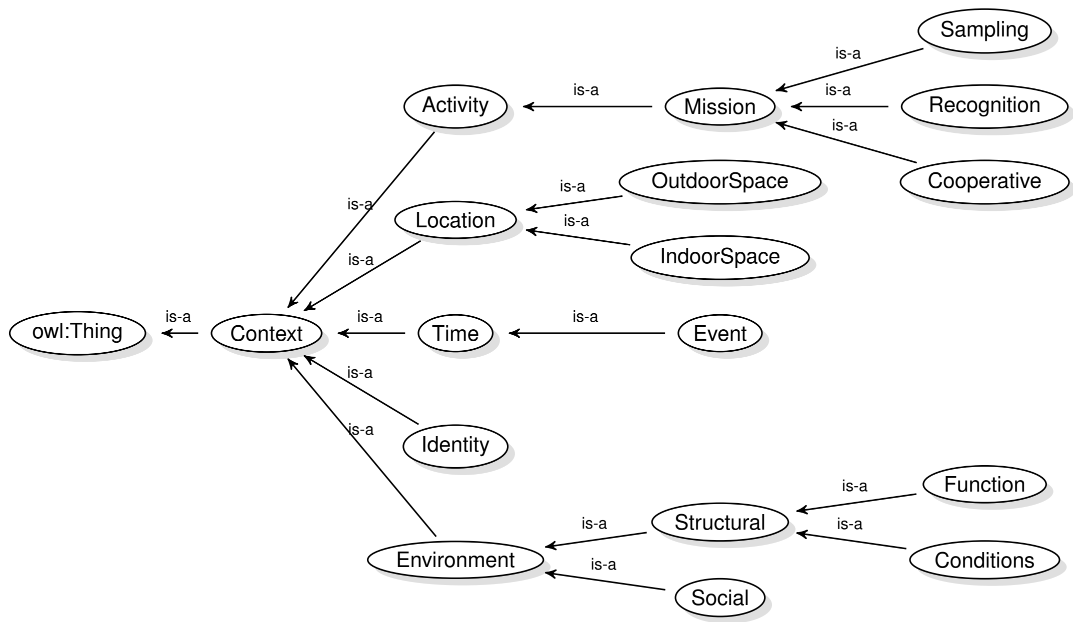


Figure 5.5: Extended Context Ontology.

Human Ontology

The Human Ontology model is based on the "Friend Of A Friend" (FOAF) ontology [176], which is a collaborative effort amongst Semantic Web developers to semantically model user's information.

The FOAF ontology is commonly used for describing persons, their activities and their relations to other people and objects. Moreover, FOAF allows groups of people to describe social networks without the need for a centralised database. The FOAF Vocabulary specification[32] is maintained by the FOAF project[33] that grouped the main FOAF classes into three broad categories: *Core*, *Social Web* and *Linked Data utilities*. For our specific Knowledge Model, we reuse only a sub-set of the FOAF Core, which describe characteristics of people and social groups that are independent of time and technology. We use classes *foaf:Person* and *foaf:Group* to describe basic information about people in present day, historical and cultural heritage contexts. These two classes extend our Knowledge Model properties with new OPs associated with foaf:Person: *foaf:name*, *foaf:familyName*, *foaf:knows*, *foaf:age*; and with foaf:Group: *foaf:member*.

Following this assumption, we can extend the Human Ontology according to the model depicted in Figure 5.6.

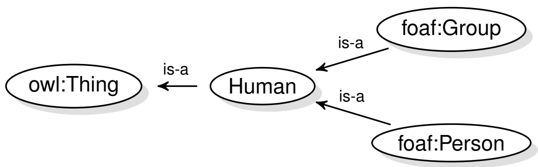


Figure 5.6: Extended Human Ontology.

Machine Ontology

In our approach, to achieve system automatic adaptation to dynamic interaction processes, we must link functionalities and algorithms properties to our information model. It is particularly relevant to establish the relationships between context entities and the interaction entities.

First, we consider functionality description. Each functionality is usually associated to at least one algorithm. For each functionality, the description should include properties *Parameters* and *Type* referring to generic information about the class. The association to the class *Context* relates a given functionality to its requirements in

[32] `http://xmlns.com/foaf/spec/`
[33] `http://www.foaf-project.org`

terms of context entities (i.e., the conditions when it can be executed). At this level, instances of Context may refer to concepts given by sources of information other than sensors (e.g., features obtained from a classification process).

Second, we consider algorithm description. The model for any algorithm includes *Input*, *Output*, and *Parameters*. In our description, we used *Input* and *Output* to define the input and output datatypes, respectively, and the *Parameters* to describe the information related to the algorithm type and variables used. As for functionality, the association to the class *Context* relates the algorithm to the context when it can be applied. At this level, the context instances may refers to concepts given by the outputs of sensors (e.g., range of light intensity where the algorithm is known to work or not). The logical view of this association is depicted in Figure 5.7.

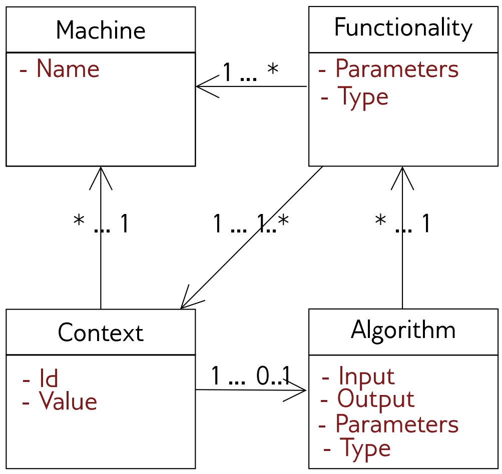


Figure 5.7: Functionality description model.

Hence, we can extend the Machine Ontology according to the model depicted in Figure 5.8.

Interaction Ontology

Similarly to the previous models, we can extend the Interaction Ontology by detailing further the concepts associated with IntetactionModalities. A more detailed model for the Interaction Ontology is depicted in Figure 5.9

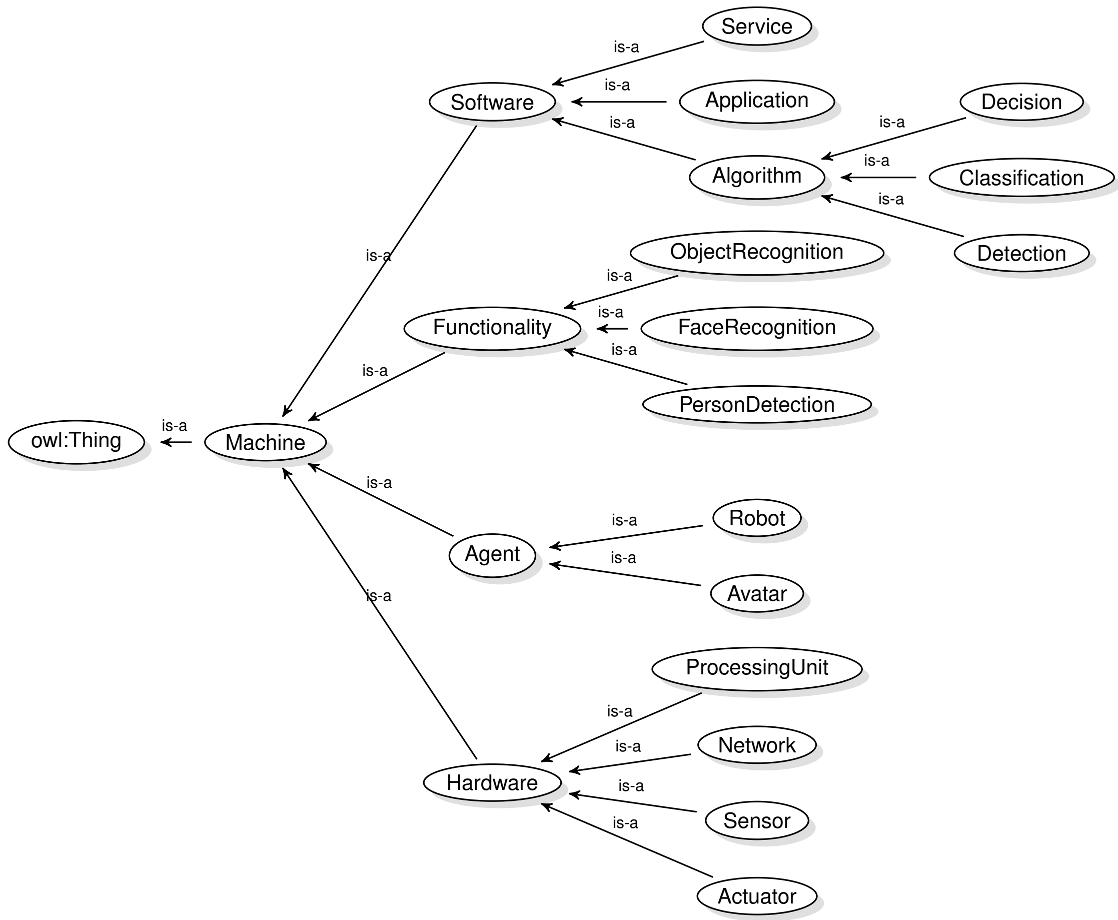


Figure 5.8: Extended Machine Ontology.

5.4 Creating a Scenario Ontology

After we have created the upper ontology and collected enough scenarios describing the agent's behaviour, we can now turn to how creating domain (i.e. context) specific ontologies.

In each Gherkin scenario, following the format of Figure 5.3, we write each sentence as similar as possible to an ontology triple format (i.e. subject-predicate-object). The resulting assertions are represented using OWL or RDF, see listing in Figure 5.10 for a snippet of the representation for the asserted axioms referring to individual *context1*.

Therefore, each domain specific ontology represent the corresponding instances in the scenario and their relationships by means of a graph. For the sake of providing an example, we illustrate in Figure 5.11 one specific case that will be presented later.

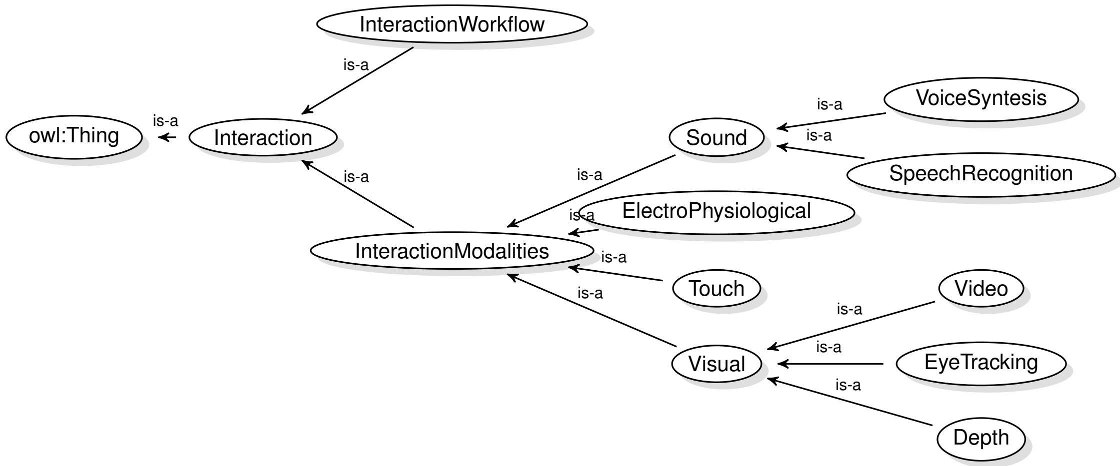


Figure 5.9: Extended Interaction Ontology.

Figure 5.10: Snippet of OWL representation for the Gherkin Scenario illustrated in 5.11.

```
<owl:NamedIndividual  rdf:about="http://www.contextawarerobotics.org/
aurora/kr/feature1-scenario1.owl#context1">
      <rdf:type rdf:resource=
      "http://www.contextawarerobotics.org/aurora/kr/im-aurora.owl#Context"
      />
      <im-aurora:hasActivityMission rdf:resource=
      "http://www.contextawarerobotics.org/aurora/kr/feature1-scenario1.owl
      #mission1"
      />
      <im-aurora:hasEnvironmentCondition rdf:resource=
      "http://www.contextawarerobotics.org/aurora/kr/feature1-scenario1.owl
      #lightCondition1"
      />
      <im-aurora:hasIdentity rdf:resource=
      "http://www.contextawarerobotics.org/aurora/kr/feature1-scenario1.owl
      #identity1"
      />
      <im-aurora:hasInteractionWorkflow rdf:resource=
      "http://www.contextawarerobotics.org/aurora/kr/feature1-scenario1.owl
      #interactionPolicy1"
      />
  </owl:NamedIndividual>
```

5.5 Using knowledge to plan and control interaction

The last step on our framework corresponds to using the lower ontologies that model interaction workflows. This is done by integrating this information while in the plan-

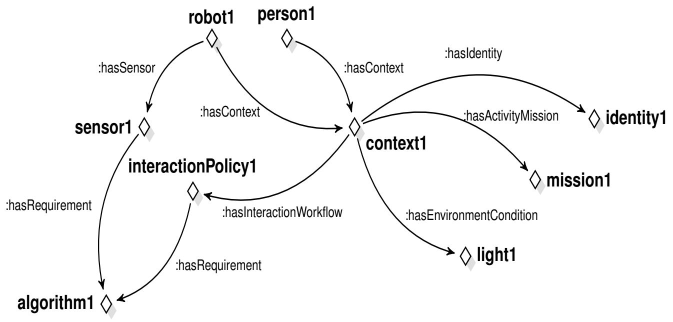


Figure 5.11: Gherkin Scenario represented in terms of an ontology, instantiating classes from the upper ontology defined in Figure 5.4.

ning phase of the decision process. The resulting plans can then be used for controlling the agent's actions (i.e. interaction behaviour).

Following an identical structure as the generic architecture proposed by Alami et al. [115], we designed the architecture for our artificial social companion (i.e. agent) broken down into *four* major blocks that deal with *knowledge representation* (i.e. ontologies), *functional level* (i.e. the collection of features for data acquisition, processing and actuation), *execution level* (i.e. finding connections between features and coordinating their execution) and *decisional level* (i.e. decision process for planning and action control).

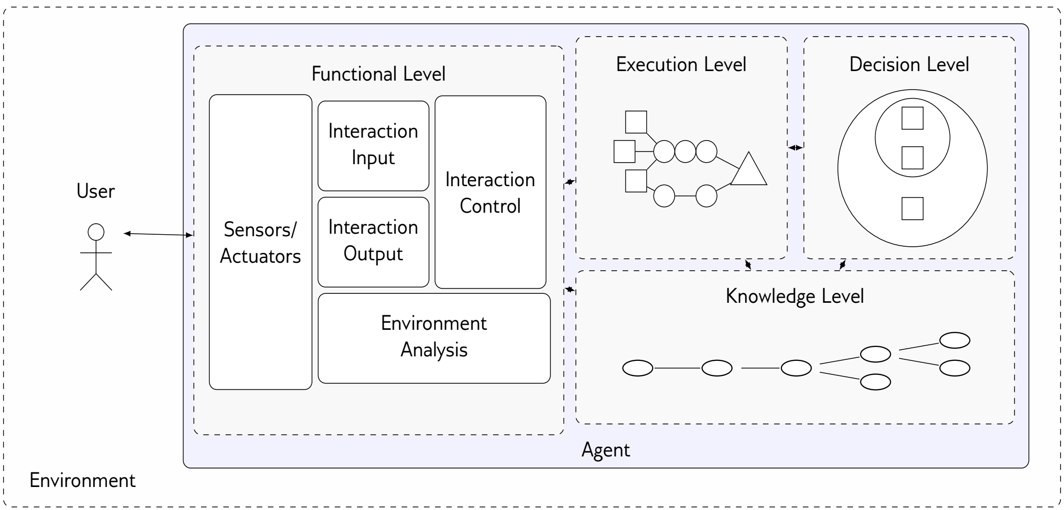


Figure 5.12: Overall system architecture

5.5.1 Knowledge Representation Level

This level supports the operation of all other levels in the architecture. It stores information to be used on situation assessment, context management, goal and plans management, action refinement, execution and monitoring. This level corresponds to the upper and lower level ontologies explained in the two previous sections.

5.5.2 Functional Level

This level include the basic built in agent's perception and action features. For example, in this level we will have the algorithms that are responsible for acquisition and processing of raw data from sensors and drivers for actuators. Also, in the Functional level we have the features responsible for context acquisition and interfacing with the Knowledge level.

The sets of features, which will allow the agent to be able to perceive and act (i.e. performing interaction) are illustrated in figure 5.13.

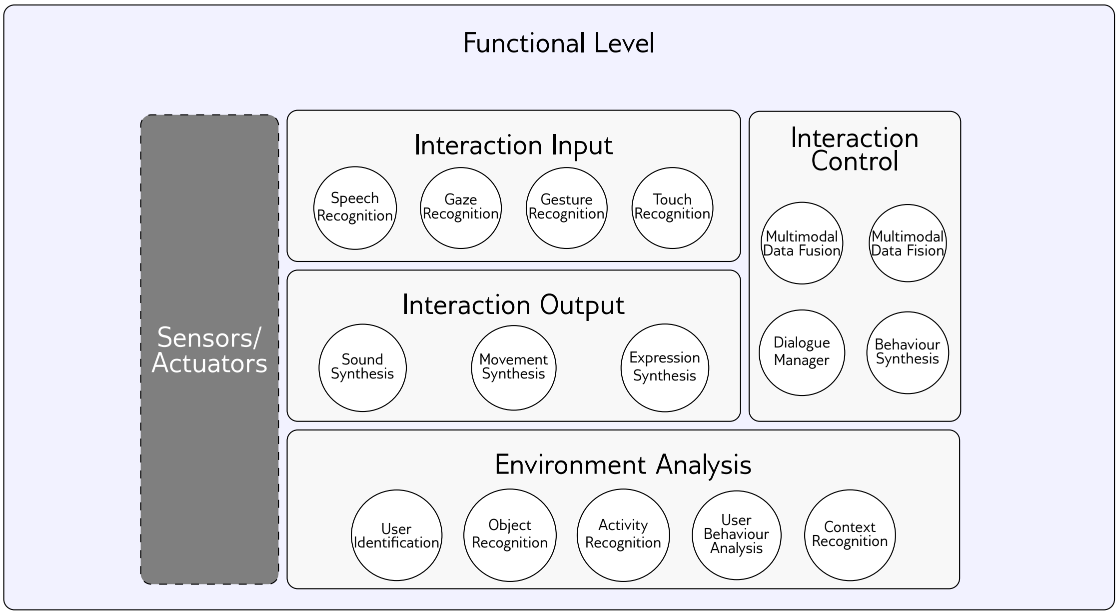


Figure 5.13: Examples of algorithms that might be present in the Functional level

The features are grouped into four main categories. **Interaction inputs** include the features related with user interaction inputs (i.e. Speech recognition, Gaze recognition, Gesture recognition and Touch/Haptics recognition). These modules contribute to the overall system by perceiving the user's explicit state (i.e. intentions and emotions). **Environment analysis** is also related with perception mechanisms, but it contributes with implicit states. It refers mainly to what can be recognized and identified from the environment and from user habits profile (i.e. User identification, Object recognition, Activity recognition, User behaviour analysis, Context recognition). **Interaction control** features receive inputs from the Interaction inputs features and handle the data fusion and fission in order to update the agent's internal state and determine appropriate response (i.e. Multimodal data fusion, Multimodal data fision, Dialogue manager and Behaviour synthesis). **Interaction outputs** are directly linked to the synthesis features generating the appropriate verbal and non-verbal behaviour of the companion and communicate relevant information to the user via the graphical user interface (i.e. Expression synthesis, Movement synthesis) and vocal outputs (i.e. Sound synthesis).

5.5.3 Execution Level

This level is responsible for finding connections between features and coordinating their execution. Here, we consider that each system or machine must offer a set of functionalities that are decomposed into a set of features (e.g. algorithms), which may work under a certain context. For the sake of the following descriptions, from this point on, we will consider a feature is an algorithm.

Discovering connections between algorithms

Lets consider that all systems have a set of algorithms, and each algorithm has restrictions on the input data and output data. Furthermore, the implementation of any functionality results from the sequencing a given set of algorithms (i.e. sub-set of all algorithms implemented in the system).

Also, lets assume that system's adaptation can be modeled as a search problem where we want to find the optimal sequence of algorithms to implement a given functionality. From all possible combinations of algorithms that together implement a functionality, the optimal sequence will depend on the conditions prevailing at each instant (i.e. context). Therefore, we are considering the availability of algorithms that provide redundancy (i.e., same inputs and outputs), but are optimized to address the same problem in different contexts. On the other hand, we are also considering the availability of algorithms that can be sequenced (i.e., first algorithm output information to the second). Following this considerations, we can establish various possible sequences to implement the same functionality, depending on the conditions for a given time (i.e., context). By chaining some of these algorithms, we can expect to progressively transform the input data into the desired information, for example, personal identity and location. Such a chaining of algorithms can be viewed as a set of nodes in a graph connected with arcs that represent the data that flows from one algorithm output to the subsequent input. To formulate the graph that describes the possibilities of the relationships between the algorithms, lets assume we have N algorithms A_i, where i = 1, ..., N, which are characterised by their inputs Ain_i and outputs $Aout_i$. Two algorithms have a dependence if the input Ain_i of A_{i+1} is the same of the output $Aout_i$ of A_i. We can now establish that if we have a finite set of algorithms and their description in terms of their inputs and outputs, we can form the graph that represents the system's architecture, using the algorithm shown in Figure 5.14.

Figure 5.14: Algorithm to establish graph of the system. Compute the adjacency relation of the graph.

1. **for** $i, j = 1$ **to** N **do**
2.   **if** $Ain_i = Aout_j$ **then**
3. $d_{i,j} \leftarrow True$
4.   **else**
5. $d_{i,j} \leftarrow False$
6.   **end if**
7. **end for**

For the algorithm description, consider A_i and A_j as two algorithms; the dependence $d_{i,j}$ exists if Ain_i is the same as $Aout_j$, where i and j are integer numbers between 1 and N, in which N is the number of algorithms available in the system.

The result of this algorithm will give the basic structure for the graph that connects all possible combinations of available algorithms in our system, which will be similar to that represented in Figure 5.15. In this figure, the square nodes represent sensors, the circles represent the algorithms, and the triangle represents the goal we want to achieve.

Introducing dynamic changes

With the previous representation established, we can now focus on how to incorporate the context-based mechanism that will allow us to select the optimal sequences of algorithms in case the system is affected by condition changes, which may result in errors or faults. We want to be capable of switching between algorithms that maximize the chance of achieving a desired goal. This adaptation avoids re-planning. The optimization of algorithm selection depending on changing conditions corresponds to a decision process.

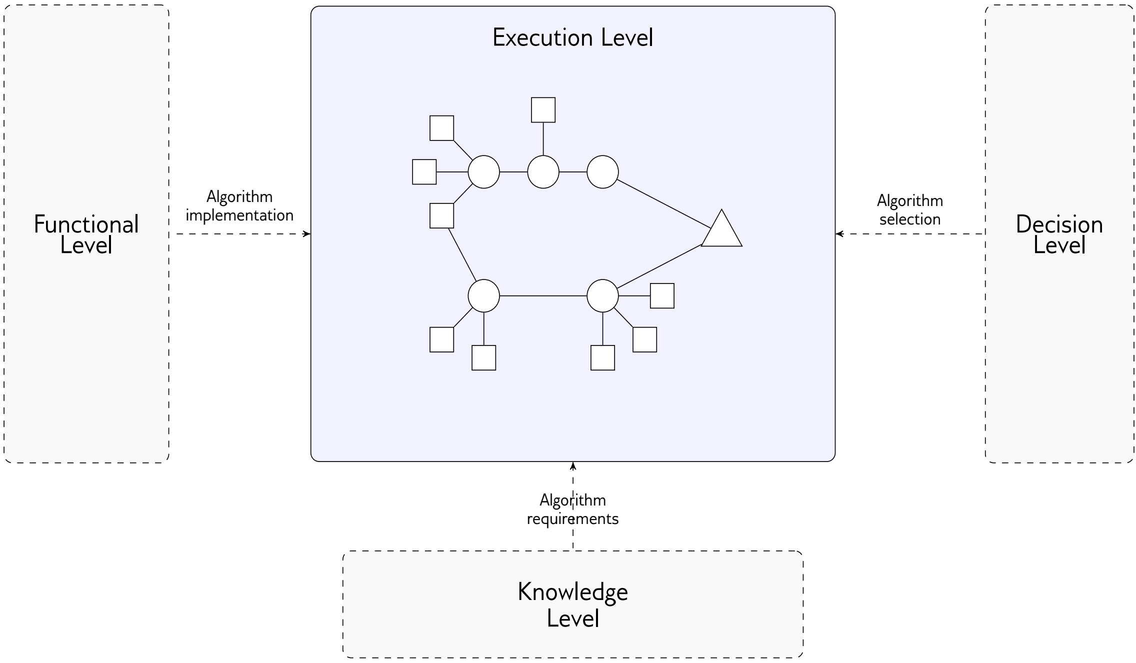


Figure 5.15: Network topology for sequenced algorithms.

Consider that each algorithm that processes direct or transformed sensor data has some requirements for it to work properly. These requirements may be related with the values contained in the input data that must be within some optimum range, or some other specific characteristics. As an example, we can mention a vision based Simultaneous Localization and Mapping (SLAM) process that only may work if there is adequate lighting for extracting the required features from the camera images.

These requirements form a set of constraints that correspond to context. When reaching the limit conditions set by the requirements, we must decide if we should switch to an alternative that either provides similar functionality or resets context, thus putting the decision process as part of system context-based adaptation.

If an algorithm requires conditions a, b, c (e.g., context1) to work, but for some reason one or more of these cannot be verified, the decision process will check if the system can perform an action in such a way that context1 can be present and the algorithm can be applied. This can happen primarily in two ways; first, the system could find some way to influence the environment in such a way that the conditions/context for the algorithm is satisfied; second, the system could find some internal strategy to overcome such a limitation – for example, it could use an auxiliary algorithm that can be used with the current context or it can generate the context for the functionality that must be used (Figure 5.16).

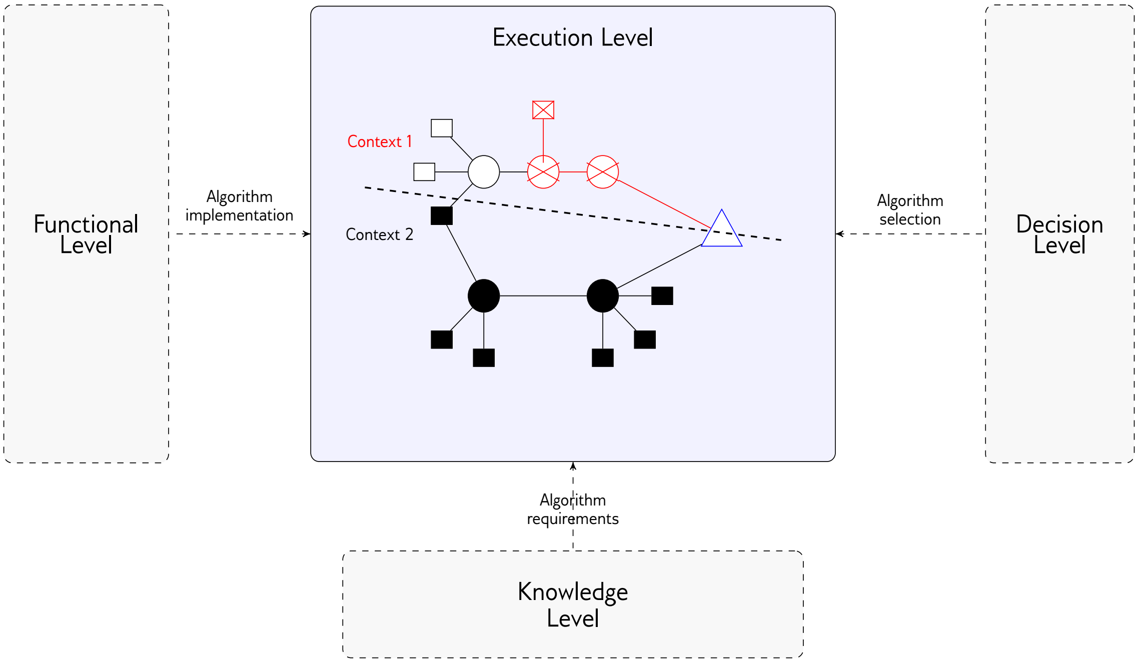


Figure 5.16: Selecting the sequence for the given context.

At this stage, we need an optimization approach that can control the selection of the most promising sequence to achieve a goal. Considering this, we need to establish the costs/weights/rewards of each edge, which will be associated with the context perceived in a given moment. For each context, the set of edges will have a different cost. This information is passed to decision analysis, which finally decides the optimal sequence of algorithms (i.e., path) to achieve a goal (i.e., provide a functionality).

This approach allows us to incorporate redundancy in terms of human-machine interaction by guarantying multiple possible solutions to achieve a similar result (ideally, the same) adapting to different contexts of operation.

Considering Adaptation Through Context Awareness

Obviously the actuation must be adapted to each situation. For example, a wheeled robot for indoors use should have wheels replaced for outdoor operation and is unable to evolve on sandy or muddy terrains. But, even if we consider a task of driving an autonomous vehicle from one point, i.e. a system that was designed to evolve on urban roads, it cannot rely on an initially defined trajectory without continuous adaptation. Other vehicles will be present, a road may be interrupted, an obstacle may force a deviation, and even a solar eclipse may occur changing the lighting conditions with respect to those considered during the preparation of the initial plan.

Coming back to indoors human environments, any kind of robot cannot rely on its initial plan as the conditions may also change during the task execution. Humans tend to move things from place to place, leave objects on the floor, open window blinds, switch lights on and off, in sum introduce larges amounts of changes in the environment that are not predictable.

In both cases, these unpredictable situations are true obstacles to the deployment of traditional robots as their functionalities are dependent on sensing modalities and algorithms that cannot adapt to every possible situation. Apart from the changes that people constantly introduce in their environments, other sources of problems may be found related with the architecture of human spaces. In houses, offices, and other buildings, we have rooms, and corridors that use artificial light, where other have windows or glass panels instead of walls for receiving sunlight and provide spectacular views over the outside spaces. Where the spaces with artificial light can be seen as more stable in terms of illumination, and therefore the consequent acquisition of images via camera sensors, the latter may create more complicated and uncontrollable problems for the sensing tasks of robots. Here the lighting conditions vary constantly and may go from dark at night to excessive light in some situations that may saturate the used cameras. As another possible problem, a glass panel as an outer wall may be totally undetectable for some laser-based sensors. To circumvent the problem of this last example we can opt for some ultrasound sensors, but these are far less precise than Laser Range Finder (LRF), thus a possible rule could be: "use LRF to detect walls except in rooms A, B and C (those with glass panels) where sonar sensors and appropriate algorithms must be used".

The above presented situations suggest the inclusion of a decision process that takes into account the present context for the selection of the sensors and algorithms to apply at each instant. The trivial solution seems to define this decision layer based on a set of rules that define for each context which sensor and/or algorithm to use. This requires a predefinition of those rules together with the relevant contexts where they should be applied.

Rules versus requirements Previously, we have presented the idea of using contextual information to decide the activation of a given set of sensors together with a specified algorithm for accomplishing a task in a predefined mission. If this decision process if rule-based, it becomes clear that the necessary set of rules may grow uncontrolled when applied to a home/office robot, if it is to show a good level of adaptability minimising the necessary human intervention.

Our proposal with respect to this is based in the inclusion of a set of elementary functionalities where each of them includes its own requirements and capabilities.

Table 5.1: Example definitions of simple functionalities

Functionality	Requisites	Output	Classification
Wall follower 1	Place with laser reflective walls LRF available	Move robot from current to goal position	Fast
Wall follower 2	Place with walls Sonar available	Move robot from current to goal position	Slow
Person detector from silhouette	Light Level>=Low	Approximate location of person	Precision=Low
Person detector from thermal images	No Sun Light	Approximate location of person	Precision=Low
Face detector	Light level=Good	Approximate location of person	Precision=Medium
OpenNI Person detector	Light level>=Good No Sun Light	Approximate location of person	Precision=High

Using the examples presented on table 5.1 we can now define a decision process that for each necessary functionality chooses the most promising approach based on the observed conditions (context) and necessary output.

5.5.4 Decision Level

This level is in charge of planning and supervising execution of the features from the other levels.

The main goal of this level is to ensure Interaction is more robust, more flexible and more dynamic. In opposition to command-driven approaches, here we use the knowledge representation from previous levels for planning and control Interaction and user interfaces. Therefore, as knowledge changes (e.g. learning process) new plans for Interaction can be computed that will result in a more robust system overall.

In command-driven approaches we explicitly describe the protocols for interaction, thus we are limiting the interaction with the agent to the execution of previously defined "rigid" plans for interaction. Our approach aims to overcome this limitation by using a knowledge representation approach that allows to represented known interaction plans in the form of asserted graphs, but that allow to infer new relationships in data using a reasoner. We can then use this information in a decision process that uses probabilistic graphical models for a seamless integration of knowledge representation and run time execution. On the other hand, the decision process can add complementary information about the interaction protocols by determining the likelihood of certain interaction workflows (i.e. policies in a Partially Observable Markov Decision Process (POMDP)) to occur. As described in [164–166], the mathematical formalism of POMDP is well-suited to our problem because we require an approach that takes into consideration aspects regarding limitations in a priori planning (i.e., we cannot plan every possible course of actions a priori) and the limited capability of measuring the state of the world (i.e., limited perception capability), and if we assume that interaction workflows follow a markovian process. This means that the interaction workflow to be selected only depends on the preceding state of the system (i.e. context). These aspects introduce uncertainty into the decision process; such uncertainty is not fully considered by other approaches commonly used in decision making (e.g, Decision Trees, Influence Diagrams, Multicriteria decision making, or Markov Chains). On the one hand, we assume interaction workflows follow a Markovian process (i.e., an interaction workflow depends solely on the preceding state of the system (i.e., context)). On the other hand, our problem addresses decision making (i.e., choose the right actions); thus, it addresses planning and control, not exclusively addressing perception or actuation, thus we are dealing with decision making and a higher level of information. Furthermore, we can incorporate, in the decision process model, a sensing action to reduce uncertainty as a mean to achieve our main goal, reliable and robust interaction workflow, whilst considering uncertainty in action effect and uncertainty in perception.

Therefore, in our approach, we define a POMDP (i.e. that can be computed offline) model for each context and represent the resulting policies (i.e., interaction workflows) in the scenario ontology, as depicted in Figure 5.17. During execution, an ASC will adapt its decision process to different contexts by querying its knowledge model for the most suitable scenario ontology.

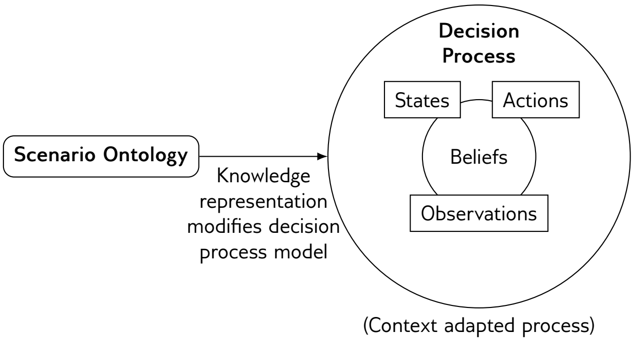


Figure 5.17: Context representation in the scenario ontology modifies the components of the decision process at each moment in time.

Partially Observable Markov Decision Process

A Partially Observable Markov Decision Process (POMDP) is described as the tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{O}, \Omega, T, R\}$ that can be specified as follows:

- $\mathcal{S}$ - State is the way the world currently exists. This set represents all possible information about the agent and its context (e.g., location, environment conditions).

- $\mathcal{A}$ - Actions form the set of possible alternative choices you can choose to make, which include algorithms that can be executed to provide a certain functionality.

- $\mathcal{O}$ - Finite set of observations of the state of the world, which correspond to measurable parameters (e.g., sensor readings). In our model, context is included in the decision model as variables in the set of observations.

- $\Omega(a, s, o) : \mathcal{O} \times \mathcal{S} \times \mathcal{A}$ - This captures the relationship between the state and the observations (and can be action dependent). $\Omega(a, s, o)$ tells the agent the probability that it will perceive observation o when in state s, after performing action a. To define the observation function, we consider a set of conditional probabilities
$$Pr(o|s', a) \tag{5.1}$$

- $T(s, a, s') : \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ - The transition function, or the likelihood of transition from state s with action a to new state s'. To define the transition function, we consider a set of conditional probabilities
$$Pr(s'|s, a) \tag{5.2}$$

- $R(s,a) : \mathcal{S} \times \mathcal{A}$ - The reward function; this refers to the reward received for transitioning to state s with action a. We specify some immediate value for performing each action in each state. The reward or payoff function in POMDPs is defined as

$$r(b,a) = \sum_{s} b(s)r(s,a) \tag{5.3}$$

where here $b(s)$ is the value of b for s.

Belief update The agent can then use the observations o it receives to update its current belief b. Specifically, if the agent's current belief for state s is b, it takes action a and gets an observation o, then its new belief vector b' can be determined using

$$b'(s') = \frac{Pr(o|s',a)\sum_{s\in\mathcal{S}} Pr(s'|s,a)b(s)}{\sum_{s'\in\mathcal{S}} Pr(o|s',a)\sum_{s\in\mathcal{S}} Pr(s'|s,a)b(s)}, \tag{5.4}$$

where here $b(s)$ is the value of b for s.

Policy The solution to a POMDP is called a policy, and it simply specifies the best action to take for each of the states. We will use π to denote the agent's policy. The optimal policy can thus be defined as

$$\pi^*(b) = \arg\max_{a\in\mathcal{A}} \Big[r(b,a) + \sum_{o\in\mathcal{O}} P(o|b,a)V^*(b')\Big], \tag{5.5}$$

where $\pi^*(b)$ yields the highest expected reward value for each belief state b, represented by optimal value function V^*, where b' is the next belief state of the agent.

Value Function The final goal of the POMDP algorithm is to find a Value Function (VF) $V(b)$ that represent the optimal policies over the belief distribution b, where b is defined with parameters $p_1, p_2, ..., p_N$, the beliefs of corresponding state, with N the number of states. The Value Iteration Algorithm (VIA) is the most common choice to compute the VF. Moreover, $V(b)$ is defined as,

$$V(b) = \sum_{i=1}^{N} v_i p_i \tag{5.6}$$

where $v_1, v_2, ..., v_N$ are the coefficients of a linear function. For a finite horizon T, equation 5.6 is a piecewise linear and convex value function $V_T(b)$ and can be represented by the maximum of a finite set of linear functions,

$$V_T(b) = \max_k \sum_{i=1}^{N} v_i^k p_i \tag{5.7}$$

where $v_1^k, v_2^k, ..., v_N^k$ denotes parameters of the k-th linear function.

Moreover, it is worth to notice that $V_T(b)$ is calculated recursively with a nested loop, which calculates $V_t(b)$ by updating $V_{t-1}(b)$ taking into account the beliefs of observations, actions and states for that horizon. We should notice that, the two main steps for calculating the value function, requires to compute one linear function for each combination of action, observation for each linear constraint of the previous value function. This means that, in the first step, we must compute,

$$v_{a,o,s}^k = \sum_{i=1}^{N} v_i^k p(o|s_i) p(s_i|a, s_j) \tag{5.8}$$

for all s states, all o observations, all a actions for all k linear functions from previous $V_{t-1}(b)$.

Additionally, the second step involves a similar type of calculations, which will compute the expectation for each combination of k linear functions with O observations. This operation is defined as,

$$v_i^{\cdot} = \gamma \left[r(s_i, a) + \sum_o v_{a,o,i}^{k(o)} \right] \tag{5.9}$$

with γ a normalizing factor, $i = 1, ..., N$ number of states and $r(s_i, a)$ corresponding to the reward of transitioning to state s_i with action a.

Context adaptation loop

In our approach we consider context recognition to be the process that leads to auto-adaption of the system. We call this process the *Context adaptation loop*. *Context adaptation loop* works as an orchestration/supervision process at a higher level of the decision level.

The system want to achieve a certain goal. Taking into account that they are multiple alternative tasks that can be executed to achieve the goal, this supervision mechanism will select the best approach to be used at a given time. It looks first if all the pre-conditions/requirements to execute the tasks are in place. Second, it selects the tasks that should be performed minimizing a cost function (i.e. weighted sum of energy consumption, computation time, priority, QoS etc).

As depicted in figure 5.18, this is a periodic process that operates in the background of the system, while the user is interacting with it. It plays the role of detecting changes in the context and provide this information to the agent's main execution loop (e.g. perception-action). The detailed *Context adaptation loop* process is depicted in Figure 5.19.

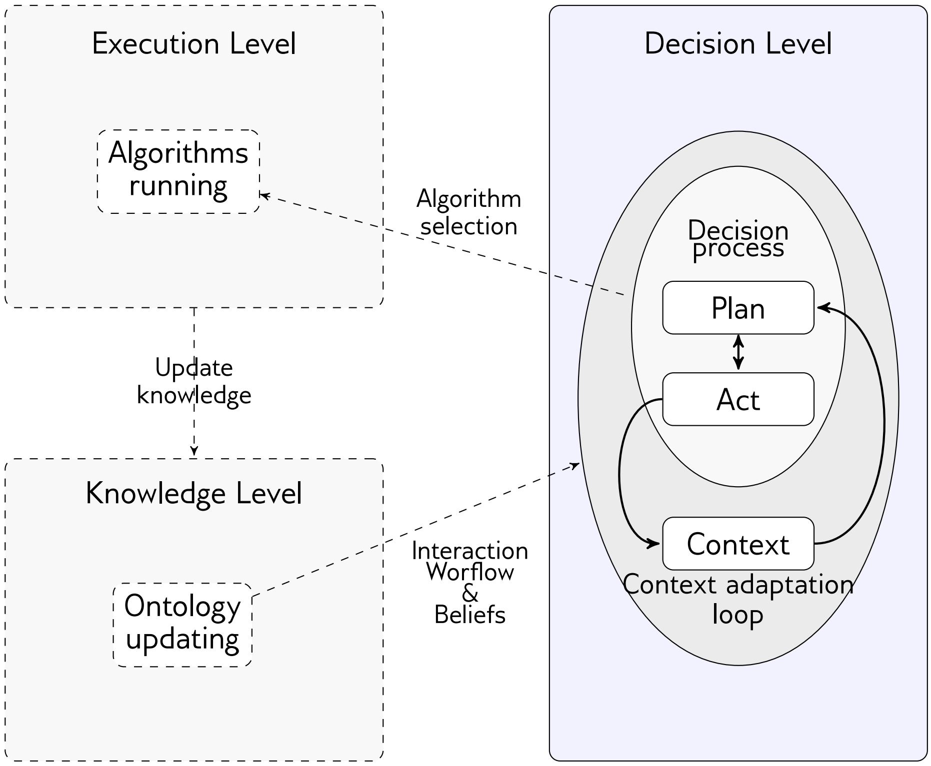


Figure 5.18: Context verification process

5.6 Experimenting with visual modalities on a Robotic Artificial Social Companion

The objective of this experiment was to understand if our approach would result in a more effective and efficient strategy to detect a human. Given we choose between different algorithms to cover different working conditions, we expected we would detect a human in more situations, using simple algorithms instead of focusing on

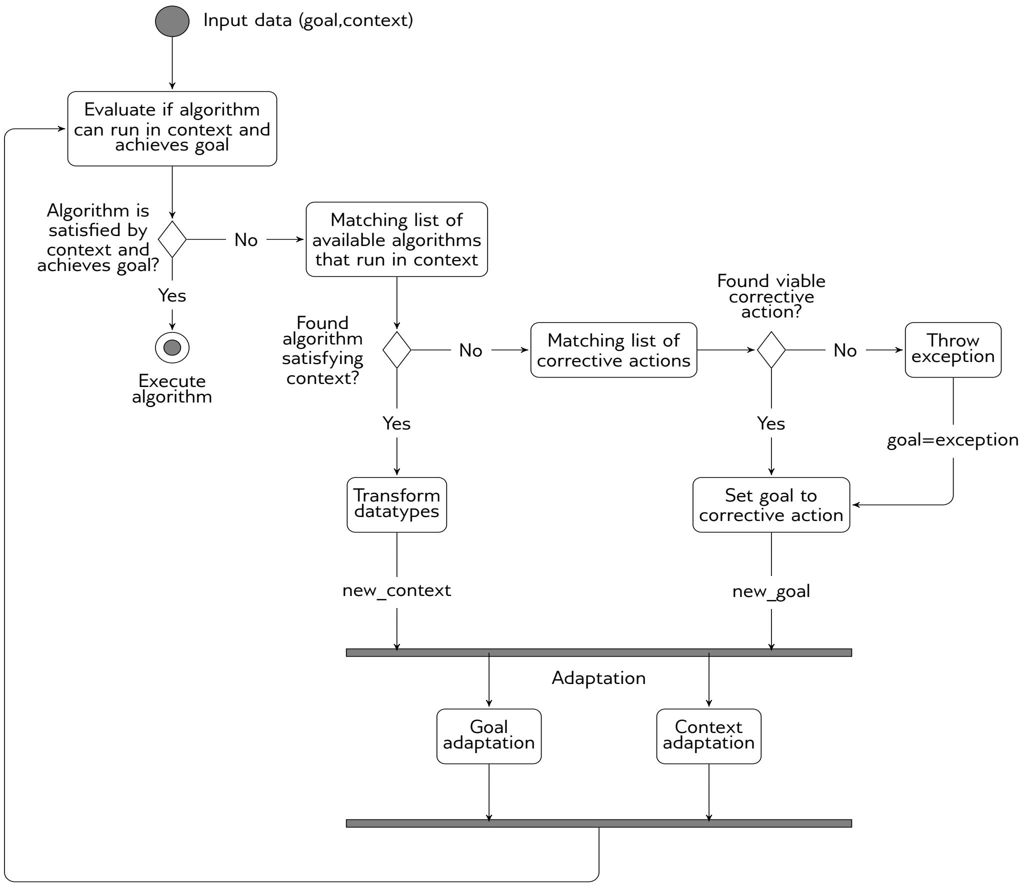


Figure 5.19: Context Adaptation Loop workflow

complex implementation of data fusion and customized adaptation. In this case, we were interested in finding the number of correct detections and number of iterations (i.e. computational time) elapsed until a detection occur. In Figure 5.20, we summarize the overall process to implement our framework. The whole process goes as we first identified the need and corresponding requirement for such the person detection feature using the available documentation from referenced research projects developing ASCs. Second, we described the scenario using the ontological representation. Following, in terms of feature implementation and description, we listed a set of possible algorithms for performing person detection and selected two of the most promising candidates, which could work under the specific context of our scenario. Then, we evaluated the performance of the two algorithms to allow us modeled the decision process for planning and, at a later stage, controlling the agent's interaction policies during run time.

In the next sections, we will detail each of the steps represented in Figure 5.20.

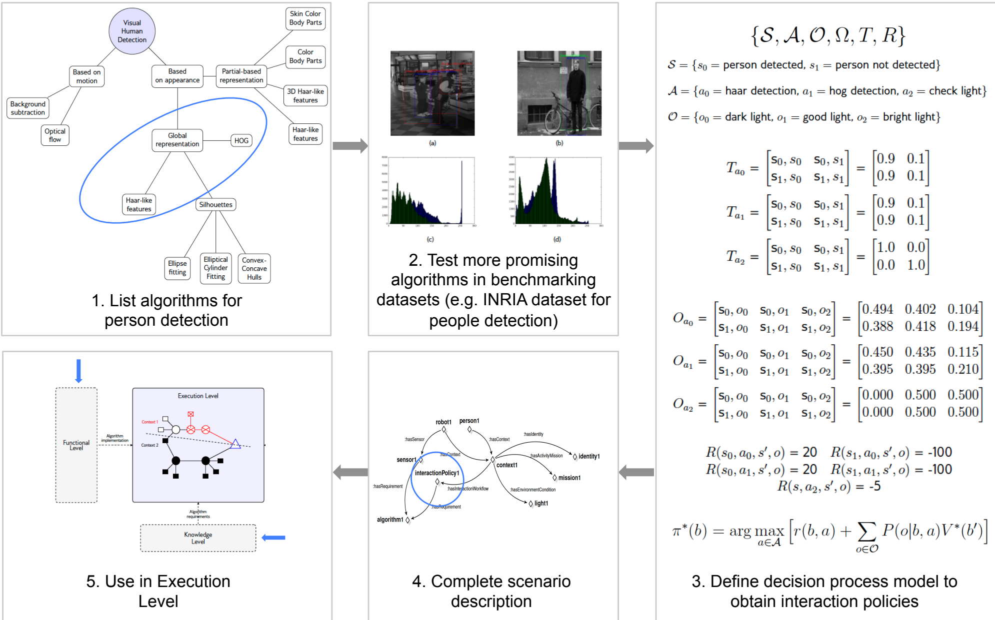


Figure 5.20: Application scenario for implementing the proposed CAHMI framework.

5.6.1 Describing agent’s behaviour

We describe the agent’s desired behavior by creating user stories that explain different operation scenarios for some relevant application scenario. In our case, we refer to the conclusions from the SocialRobot and GrowMeUp projects in [177], which identified natural Robot-User communication (e.g., via facial recognition, voice commands, and audio/video conferencing) as *user needs*. The interaction design of both systems was framed in well-identified requirements for the robotic platform that was developed. Of particular interest, in terms of human-robot interaction, we highlight the following:

- The robot should be able to guide or follow someone in the environment;
- The robot should be able to track a person (Haar-like features);
- The robot should be able to perceive person poses; and
- The robot should be able to identify a person (via face recognition or reading some RFID tag identification).

These requirements were later extended into use case scenarios in [178]. Specifically, we consider “Scenario 4: Face Recognition, Navigation and Tracking - Elderly

Care Centre Use", illustrated by Figure 5.21, and that can be summary described as following:

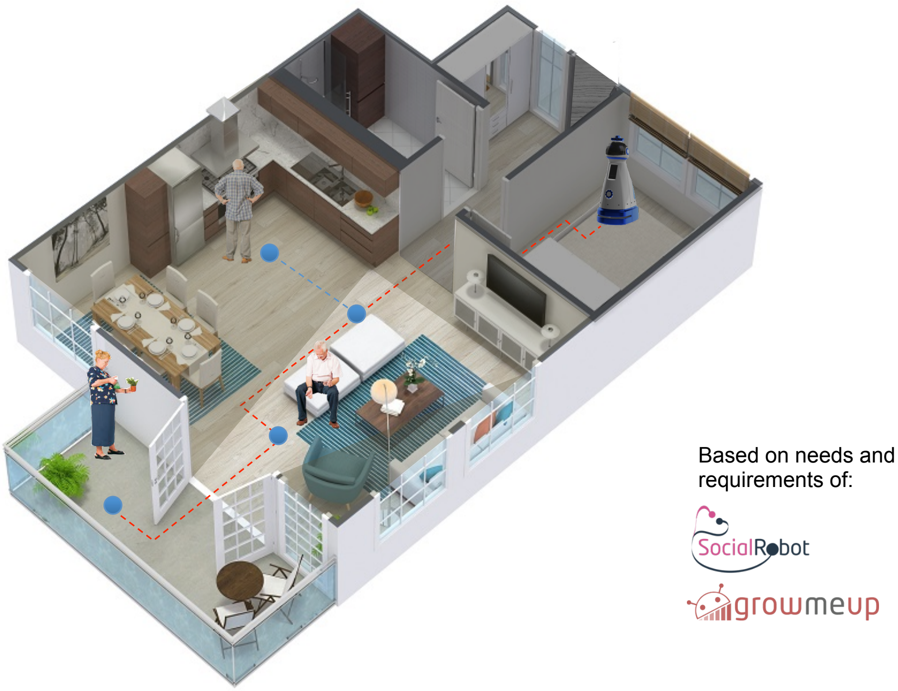


Figure 5.21: Application scenario representation.

Persona: *George is an 81 years old man having some light memory problems and also some difficulties in balancing by walking used to stay alone at home. After a fall, during the night, George decided that it was better for him to stay in an elderly care centre since the only person who could take care of him was his daughter, who lives far away in another city, and he is not a very communicative person to ask for support from his neighbours.*

User Scenario: *In the elderly house one morning George decided to walk to the small, sunny and warmer living room instead of going to the big and colder one at the main entrance. SocialRobot identified him sitting there alone, and ask him if he would like to tell his friend Kostas to join him. George responded that yes, he would like to have his friend Kostas around. SocialRobot went around the elderly centre and found his friend Kostas, a 78 years old man who has similar disabilities and behaviour ways as George. Both became friends in the elderly care centre. SocialRobot asked Kostas if he wants to join George in the small living room because he is sitting there alone. Kostas answered yes and SocialRobot accompanied him in the small sunny living room. George and Kostas were happy to be together discussing and enjoying the sun. SocialRobot recorded that they both like this room and next time it will inform them again, if it finds one of them* *sitting there alone.*

This initial formulation for a user scenario provides an extended background for the feature description using the Gherkin Scenario pattern, as described in [175] and depicted in Figure 5.22).

It is worth to note we assume that the robot does not know a-priory about persons' location. Therefore, it has to be detecting persons constantly along the way, until it finds the correct user. In this case, we can assume that the robot will navigate through the apartment where the environment conditions will vary depending on the room it is passing by. For simplification, lets consider the changes will only happen in lighting conditions (i.e. this will be our context). The challenge in this scenario is that it is difficult to develop a robust algorithm that can perform person detection equally well under diversified contexts. Here, our objective would be combining simple algorithms that are known to perform well under specific contexts. Hence, we would be interested in the robot could select the algorithm that performs the best on the context it is in and consequently reduce the errors for person detection.

Figure 5.22: Elderly Care Centre User Scenario - Gherkin Scenario example

```
Feature: Person Detection and Face Recognition

  In order to identify the different people around the elderly center
  As a SocialRobot
  I need to perform face recognition, while moving around

  Background:
    Given SocialRobot moves around the elderly center
    And the light conditions will be different in distinct divisions of
    the building

  Scenario: Person Detection and Face recognition in dimmed light
    Given the robot is moving around detecting people using the haar like
    features algorithm
    And the robot is crossing a division with ambient light below
    200-350 luxes
    When the robot selects hog like features algorithm to detect people
    in its way
    And the robot selects haar like features algorithm to detect faces
    And the robot selects eigenfaces algorithm to identify the person
    Then the robot should identify the person
```

5.6.2 Creating scenario specific ontologies

In the Gherkin scenario of Figure 5.22, we write each sentence as similarly as possible to an ontology triple format (i.e., subject-predicate-object). The domain-specific ontology related to the previous example is illustrated in Figure 5.23. This ontology represents the corresponding instances in the scenario and their relationships by means of a graph.

The resulting assertions are represented using OWL; see the listing in Figure 5.24 for a snippet of the representation for the asserted axioms referring to individual *context1*.

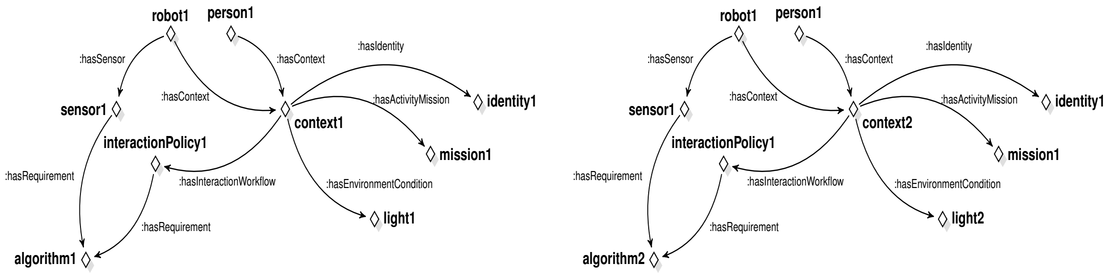


Figure 5.23: Gherkin Scenario (Feature1 - Scenario 1) represented in terms of an ontology, instantiating classes from the upper ontology defined in figure 5.4.

5.6.3 Using knowledge during runtime operation

The last step on our framework consists of using the lower ontology to model the plan for interaction workflows.

To this end, we modeled the decision process model (i.e., POMDP) for the specific scenario. Assuming that our model converges for an infinite horizon, it is possible to define a *policy graph* that can be used latter at run time (i.e., planning phase). This result will be stored as the value for the instances of class *InteractionWorkflow* in Figure 5.4 that correspond to *interactionPolicy1* in the example illustrated above in Figure 5.23.

The great advantage of using an ontology representation is that after asserting a set of axioms, we can use a *Reasoner* to infer new knowledge from the relationships between instances (i.e., this process is also known as classifying the ontology). In other words, we only need to define explicitly that the *interactionPolicy1* requires *algorithm1* as the Reasoner would infer that *interactionPolicy1* also requires *sensor1*, given that *hasRequirement* is a transitive object property. Another advantage is the scalability and flexibility of merging different ontologies into a "unique" knowledge base.

Figure 5.24: Snippet of OWL representation for the Gherkin Scenario illustrated in 5.22 and 5.23

```
<owl:NamedIndividual
rdf:about="http://www.contextawarerobotics.org/
aurora/kr/feature1-scenario1.owl#context1">
     <rdf:type rdf:resource=
     "http://www.contextawarerobotics.org/aurora/kr/im-aurora.owl#Context"
     />

     <im-aurora:hasActivityMission rdf:resource=
     "http://www.contextawarerobotics.org/aurora/kr/feature1-scenario1.owl
     #mission1"
     />

     <im-aurora:hasEnvironmentCondition
     rdf:resource=
     "http://www.contextawarerobotics.org/aurora/kr/feature1-scenario1.owl
     #lightCondition1"
     />

     <im-aurora:hasIdentity rdf:resource=
     "http://www.contextawarerobotics.org/aurora/kr/feature1-scenario1.owl
     #identity1"
     />

     <im-aurora:hasInteractionWorkflow
     rdf:resource=
     "http://www.contextawarerobotics.org/aurora/kr/feature1-scenario1.owl
     #interactionPolicy1"
     />
 </owl:NamedIndividual>
```

The result of inferring new knowledge by the Reasoner can be made permanent by adding the inferred axioms to initially asserted ones. This result is particularly useful for applications where a Reasoner is not available or for improved searching because inference can become impractical for large ontologies.

In runtime, we may use any programming library that can manipulate RDF/RDF-S/OWL (e.g., rdflib, Sesame for Python, or Java implementations respectively) to query our knowledge representation (also known as a triple store), using SPAQRL language, to obtain useful information from our asserted axioms and conduct the interaction workflow (i.e., following the policy graph stored previously). An example of these types of queries is illustrated in Figure 5.25.

Figure 5.25: SPARQL query to get all contexts for all robots defined in the knowledge representation using rdflib for a Python implementation.

```

import rdflib
import rdfextras

def getRobotContext(graph):

    qres = graph.query("""
    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX owl: <http://www.w3.org/2002/07/owl#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
    PREFIX im-aurora: <http://www.contextawarerobotics.org/aurora/kr/im-aurora.owl#>
    PREFIX : <http://www.contextawarerobotics.org/aurora/kr/feature1-scenario1.owl#>

    SELECT DISTINCT ?robot ?context
    WHERE {
    ?robot rdf:type im-aurora:robot .
    ?robot im-aurora:hasContext ?context .
    }""")

    for row in qres:
        print("%s hasContext %s" % row)

    return
```

5.6.4 Experimental validation

The experimental validation of our work replicated the conditions of the scenarios described in the Figure 5.22. The goal of the experiment was to answer our research question, studying the effects of integrating a decision process that selects interaction workflows to automatically adapt to different environment conditions (i.e. context) aiming to: 1) improve the usability of an interactive agent and 2) make the human-machine interaction component of the system more robust (i.e. fewer failures). The goal was to detect people under scenarios with varying light conditions and in different backgrounds.

Visual human detection background

As an initial part for this study, we selected two algorithms that are commonly used for people detection, but they work in complementary and similar conditions with different performances.

We summarize some of the approaches that exist to perform visual human detection. The distinct approaches can be classified in two main categories: visual detection based on appearance and visual detection based on motion. Furthermore, we can sub-divide the methods based on appearance into Global representation or Part-based Representation. Methods based on regions like, for example, Haar-like features, 3D Haar-like features, Color Body Parts, Skin Color Body Parts, usually have better performance but are more computational expensive. We present a summary taxonomy for these methods in Figure 5.26.

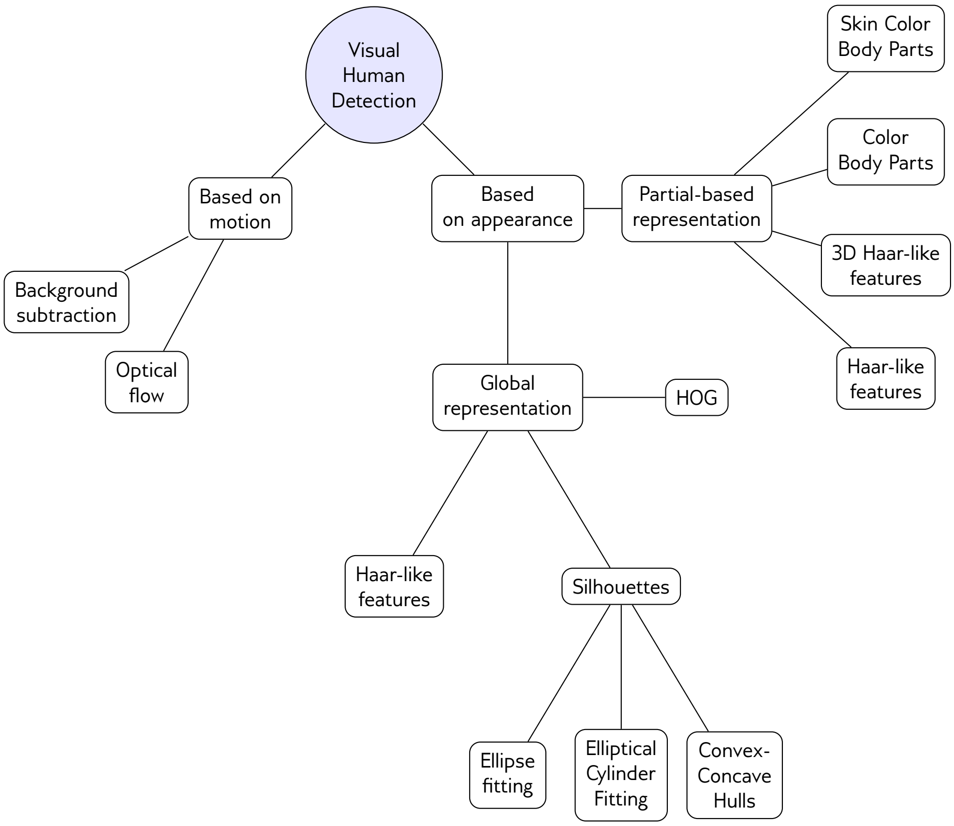


Figure 5.26: Taxonomy of visual methods for human detection

Haar-like features Based of rectangular features similar to Haar basis functions. Simple features that can be computed very rapidly using an integral image. Although these methods are very efficient, the Haar-like features are not capable of handling the complexity in real-world images and videos.

Histograms of Oriented Gradients based detection The advantage of HOG features is that each image cell is statistically represented by a histogram of the gradient orientations and magnitudes, thus it is more invariant to illumination, shadows, etc.

Methods based of features of silhouette Systems based of silhouette use to obtain worse results than methods based on shape, but they have lower complexity. They do not need accurate silhouettes and therefore approximation is enough, whilst shape models need to obtain accurate silhouettes. And it could be so complicated on complex scenarios: light conditions changes, shadows, motion velocity, etc. For this reason these methods use to be a simple and fast alternative. Some examples include algorithms Ellipse fitting, Elliptical Cylinder Fitting, Convex-Concave Hulls.

Background subtraction methods While this approach is quite efficient for detecting isolated moving objects, it is not capable of detecting static objects or separating individual humans when they are close to one another. However, it has a few limitations: it requires the video is taken by a static camera - only moving objects will be detected; it is difficult to separate humans in a group; and since it does not have a specific appearance model for the target, it cannot differentiate humans from other moving objects, for example cars or animals.

Optical flow The computational cost of calculating motion features is relatively high.

In table 5.2, we present a quick comparison between algorithms. In this table, we took into consideration main advantages, limitations and the computational complexity for each algorithm. This analysis, assumes the original implementation of the algorithms. Therefore, we disregarded any type of optimized version that may exist for a specific algorithm, which could outperform the original version for a given problem.

Table 5.2: Comparison table between algorithms

Algorithm	**Advantage**	**Limitation**	**Complexity**
Background subtraction	Invariant to shape and light conditions	Cannot detect still objects and the sensor must be static	Low

Optical flow	Invariant to shape and light conditions	Cannot detect still objects and the sensor must be static	High
Global Haar-like features	Fast	Sensible to pose changes and light conditions	Low
Global HOG features	Invariant to light conditions	Sensible to changes in object orientation	Low
Global Ellipse fitting	Low complexity, robust to shape changes	Lower accuracy	Low
Global Elliptical Cylinder Fitting	Low complexity, robust to shape changes	Lower accuracy	Low
Global Convex-Concave Hulls	Low complexity, robust to shape changes	Lower accuracy	Low
Skin Color Body Parts	More accurate	Sensible to light changes	High
Color Body Parts	More accurate	Sensible to light changes	High
Partial-based representation 3D Haar-like features	More accurate	Sensible to pose changes and light conditions	High

Partial-based representation Haar-like features	More accurate	Sensible to pose changes and light conditions	High

Experimental design

We structured the experiment according a multivariate analysis of variance (MANOVA), a repeated measures design in which the samples are analyzed by all different approaches being studied, comparing the difference between means. In this experiment, we were focused on the analysis of the effects on specificity for each approach used. We considered as our primary variables the light intensity and number of persons in the scene. Considering that this experiment would lead to a very large number of images to analyze (i.e., a large population), we decided to use random samples of data to perform our trial. The sample size was calculated using the "Test 1 Mean: 1-Sample, 1-Sided" [179] method, which is useful for tests concerning whether a given result is equal to a reference value. In our case, we want to measure the result for specificity of the different algorithms; hence, based on previous work [166], we used the values for the Null Hypothesis mean (μ_0) equal to 0.5, the True mean (μ) equal to $\mu_0 \pm 0.05$, the Error Standard deviation equal to 0.2, the Power equal to 0.85 and the Type I error rate α equal to 0.1. The resulting minimum sample size n was equal to 73 images.

Experimental setting

In our experimental setting we prepared the environment in such terms that it could replicate typical living room conditions. This was achieved by performing a series of data collections acquiring visual data and luminance (i.e. RGB camera plus light sensor) in a setup environment that replicated typical room conditions as they can be observed in relevant environments (i.e. not in laboratory controlled conditions). Consequently, the resulting dataset considers the typical changes of the environment of operation as they are observed in relevant application environments (i.e., not in the controlled environment of a laboratory). More specifically, we performed an initial characterization of light conditions considering different variations of light intensity and illumination source, as summarized in Table 5.3. Finally, we conducted our data collection in a room environment with normal and dimmed light conditions with luminance between 0 to 20 lux. Light conditions is known to have a strong influence in visual-based people detection, hence it is a known limitation. We setup an experiment where the agent has the capability of executing the two detection algorithms Haar-like features detection and Histograms of Oriented Gradients (HOG) features detection, but it must be capable of selecting the one that ensures better performance based on light conditions.

Table 5.3: Relevant characteristics of environment conditions (context of operation).
*depending on sensor range

Environment	Description	luminance range (lux)
Room	natural day light	[50, 400]
Room	artificial light	[10, 250]
Room	normal light	[10, 50]
Room	dimmed light	[1, 10]
Room	total darkness	0
Outdoor	day light	> 1000
Other	direct contact with lamp	up to 3000*

Experimental implementation

First, we used Protege[34] for designing and working with ontologies. The next step was to study the behavior of the algorithms using our previous approach described in [166], where we used the INRIA dataset[35]. From this study, our problem was defined as POMDP and solved for an infinite horizon that converged for a tolerable range of marginal improvement for the resulting policy graph. To achieve this solution, we used Anthony Cassandra's POMDP solve[36]. The resulting policy graph was incorporated into the specific ontology for our particular scenarios as the literals of the Data Property *policyGraph* in InteractionWorkflow class (instanciated in interactionPolicy1). Following this initial setup, we collected a dataset of aggregated visual and light information (i.e., video with 78 frames plus time-stamped light data in an additional file). We selected two algorithms that are commonly used for person detection – Haar-like features and Histograms of Oriented Gradients (HOG). These two approaches implemented the same functionality, but their performance differs depending of illumination conditions. We measured precision, recall, f-measurement and computational time for these different runs.

[34] `https://protege.stanford.edu`

[35] `http://pascal.inrialpes.fr/data/human/` for people detection and measured precision, recall, f-measurement and computational time for the Haar-like features and HOG algorithms

[36] `http://www.pomdp.org/code/index.html`

Defining POMDP components

The POMDP relies on defining the set of states, the expected observations from those states, the action transition matrix and the reward structure.

In our experiment, without restricting generalization, we consider a simple example that we defined as:

- $\mathcal{S} = \{s_0 =$ person detected, $s_1 =$ person not detected$\}$
- $\mathcal{A} = \{a_0 =$ haar detection, $a_1 =$ hog detection, $a_2 =$ check light$\}$
- $\mathcal{O} = \{o_0 =$ dark light, $o_1 =$ good light, $o_2 =$ bright light$\}$

The state transitions $T(s, a, s')$ can be defined as:

$$T_{a_0} = \begin{bmatrix} s_0, s_0 & s_0, s_1 \\ s_1, s_0 & s_1, s_1 \end{bmatrix} = \begin{bmatrix} 0.9 & 0.1 \\ 0.9 & 0.1 \end{bmatrix}$$

$$T_{a_1} = \begin{bmatrix} s_0, s_0 & s_0, s_1 \\ s_1, s_0 & s_1, s_1 \end{bmatrix} = \begin{bmatrix} 0.9 & 0.1 \\ 0.9 & 0.1 \end{bmatrix}$$

$$T_{a_2} = \begin{bmatrix} s_0, s_0 & s_0, s_1 \\ s_1, s_0 & s_1, s_1 \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

The observation probabilities $O(a, s, o)$ can be defined as:

$$O_{a_0} = \begin{bmatrix} s_0, o_0 & s_0, o_1 & s_0, o_2 \\ s_1, o_0 & s_1, o_1 & s_1, o_2 \end{bmatrix} = \begin{bmatrix} 0.494 & 0.402 & 0.104 \\ 0.388 & 0.418 & 0.194 \end{bmatrix}$$

$$O_{a_1} = \begin{bmatrix} s_0, o_0 & s_0, o_1 & s_0, o_2 \\ s_1, o_0 & s_1, o_1 & s_1, o_2 \end{bmatrix} = \begin{bmatrix} 0.450 & 0.435 & 0.115 \\ 0.395 & 0.395 & 0.210 \end{bmatrix}$$

$$O_{a_2} = \begin{bmatrix} s_0, o_0 & s_0, o_1 & s_0, o_2 \\ s_1, o_0 & s_1, o_1 & s_1, o_2 \end{bmatrix} = \begin{bmatrix} 0.000 & 0.500 & 0.500 \\ 0.000 & 0.500 & 0.500 \end{bmatrix}$$

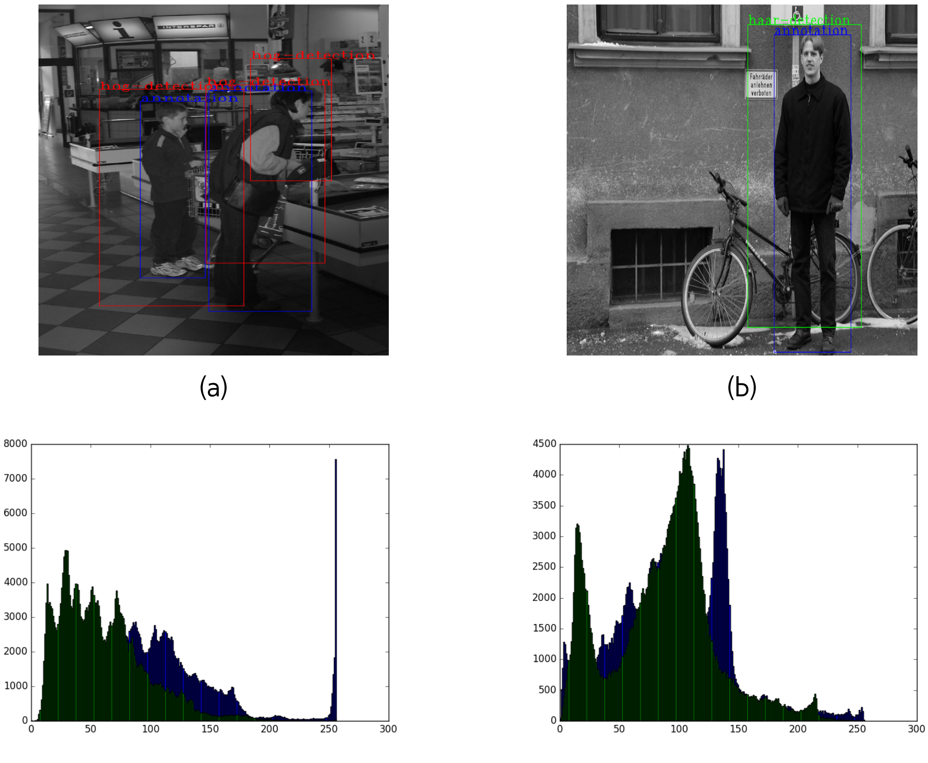
The rewards $R(s, a, s', o)$ can be defined in such a way that a positive reward is given to an action that leads to the state of a person detected, but penalizes otherwise. Penalties were set as to encourage the action of checking context (i.e. light conditions) before we achieve the state of person not detected. This scheme of rewards is described as:

$$R(s_0, a_0, s', o) = 20 \quad R(s_1, a_0, s', o) = -100$$
$$R(s_0, a_1, s', o) = 20 \quad R(s_1, a_1, s', o) = -100$$
$$R(s, a_2, s', o) = -5$$

5.6.5 Results

The results obtained revealed that, initially for the selected dataset, both algorithms work correctly in 80% of the situations, and in the remaining 20% at least one fails in detecting people (19% one fails, 1% both fail).

The results for HOG features detection showed that it is more precise than Haar-like features detection overall (49.6% more precise, 27% increased recall and improvement of f-measurement of 43.5%). However, for the cases of low light conditions, HOG detection performed poorly, not being capable of detecting in most of the cases (failing in 12% of the dataset). On the other hand, the same applies for Haar-like features detection for cases with high light conditions (failing in 9% of the dataset). Figure 5.27 illustrates an example of the results obtained in the dataset.



(a) (b) (c) (d)

Figure 5.27: Example of images with detected persons and image histograms. HOG detection performing better in brighter images and Haar detection performing better in darker images.

For the cases where algorithms made at least one detection, we analysed computational time, precision, recall and f-measurement. The results are depicted in

Figure 5.28.

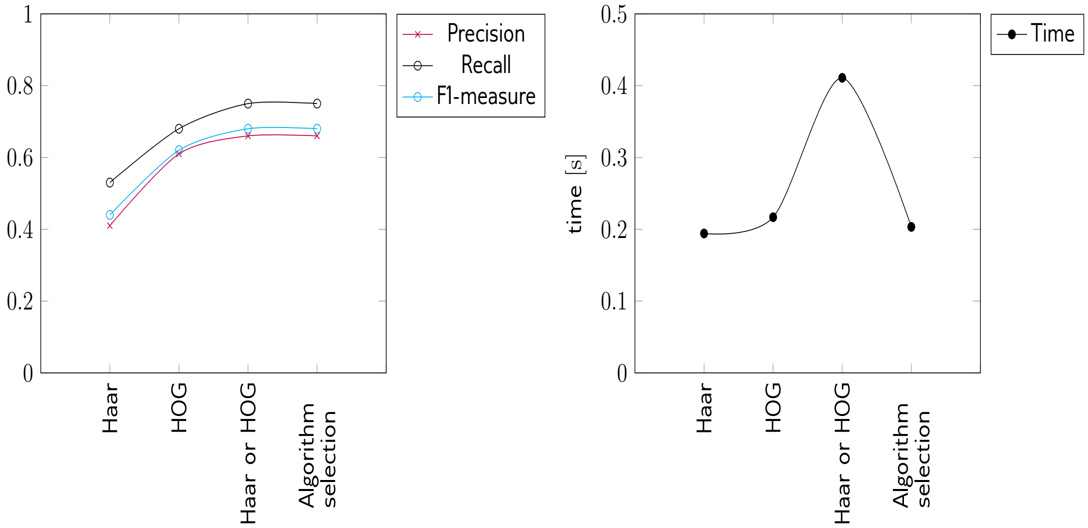


Figure 5.28: Experimental results for the different approaches. On the left, lines represent the results for precision, recall and f-measurement. On the right, line represents computational time taken to complete detection.

Attending to the obtained results, we conclude that when performing the experiment using both algorithms running concurrently (i.e., logical OR), we were able to improve the precision in 8.1%. However, this approach incurs added computational time (increase of 89.5%), as one algorithm executed after the other lengthened both computational times. Our approach mitigates this problem (increase of 4.7%) while achieving equivalent improvements in terms of precision and recall (precision = 66%, recall = 75%). The lower computational time results from the decision process selecting the optimal algorithm to be executed for a given image. The computational time until detecting a person includes the cases when the selected action is *check-light*, but this can be neglected compared to the computational time required for executing each algorithm.

In spite of the results described above, we must identify limitations for the approach that we followed in this experiment. Given that our results support only the applicability of our approach to an experimental setup for the most simple case, we assume that for some simple examples like in our example scenario, we could have achieved similar results by using threshold for light conditions. However, for more complex systems, this approach would be difficult to implement because it would require expert analysis and it would be impractical for systems with a large set of conditions. On the other hand, the proposed approach inherits the limitations often identified for POMDP. These limitations are associated with a greater computational

Table 5.4: Statistical measures of the overall performance for decision process compared to single selection of body detection algorithms.

Metric	DP	Haar	Hog
Recall	0,19	0,15	0,48
Specificity	0,60	0,31	0,20
Precision	0,44	0,18	0,49
Negative predictive value	0,31	0,28	0,19
Fall-out	0,40	0,69	0,80
False negative rate	0,81	0,85	0,52
False discovery rate	0,56	0,82	0,51
Accuracy	0,34	0,23	0,37
F1 score	0,27	0,16	0,48

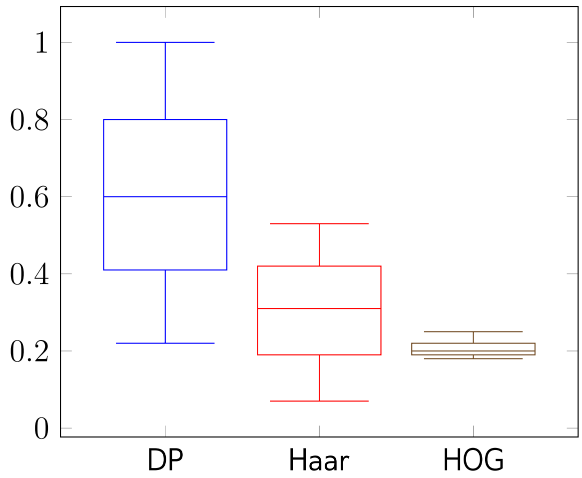


Figure 5.29: Box plot for Specificity.

complexity for models that consider more than two states or that require continuous spaces, which result in unfeasible implementations. However, provided these limitations are solved, for example, by using distributed programming, this approach benefits from being scalable and flexible to change. The model can be adapted to incorporate new variables and trained to generate updated policies (i.e., course of actions). Extrapolating our case study for more complex setups, the improvement in performance and autonomy will prove to be significant.

We analyzed the data that corresponded to three runs for each video. In the first run, we used the selected action for decision process; in the second run, we used only the Haar detection algorithm; and in the third run, we used only HOG detection.

The results for the statistical analysis of the detections outcomes were compiled into Table 5.4 and Figure 5.29.

Figure 5.30: Examples of frames from the scenario dataset. From top-left to bottom-right: we see the first two frames and the last, where we have false positives for Haar and HOG algorithms, but the action from the decision process is "Check-light", which contributes with less errors; the next three frames represent examples of correct detections by the Haar or Hog algorithms and the action selected by the decision process corresponding to most probable algorithm to work under the detected light conditions; at the bottom we summarize the overall results for true positives/negatives in light green, False positives in light red and False negatives in yellow. The first row represents the periods where light conditions were considered normal or dark. The following rows group the results for the decision process, Haar and HOG algorithms.

In Figure 5.30, we present some examples of frames acquired and a visualization of the recorded hits, misses and errors for each run.

The experimental setup described before allowed us to obtain results that confirm the second objective. From the obtained results, two main advantages can be observed from the statistical measures: first, the specificity value for DP is on average 2.5 times the specificity for the Haar and HOG algorithms when used in single operation (i.e., getting less errors resulted in a higher value for the true negative rate); second, precision for the DP is 11% less than that of the HOG algorithm, which showed the best overall performance. Attending to these results, we confirmed the second statement in our hypothesis.

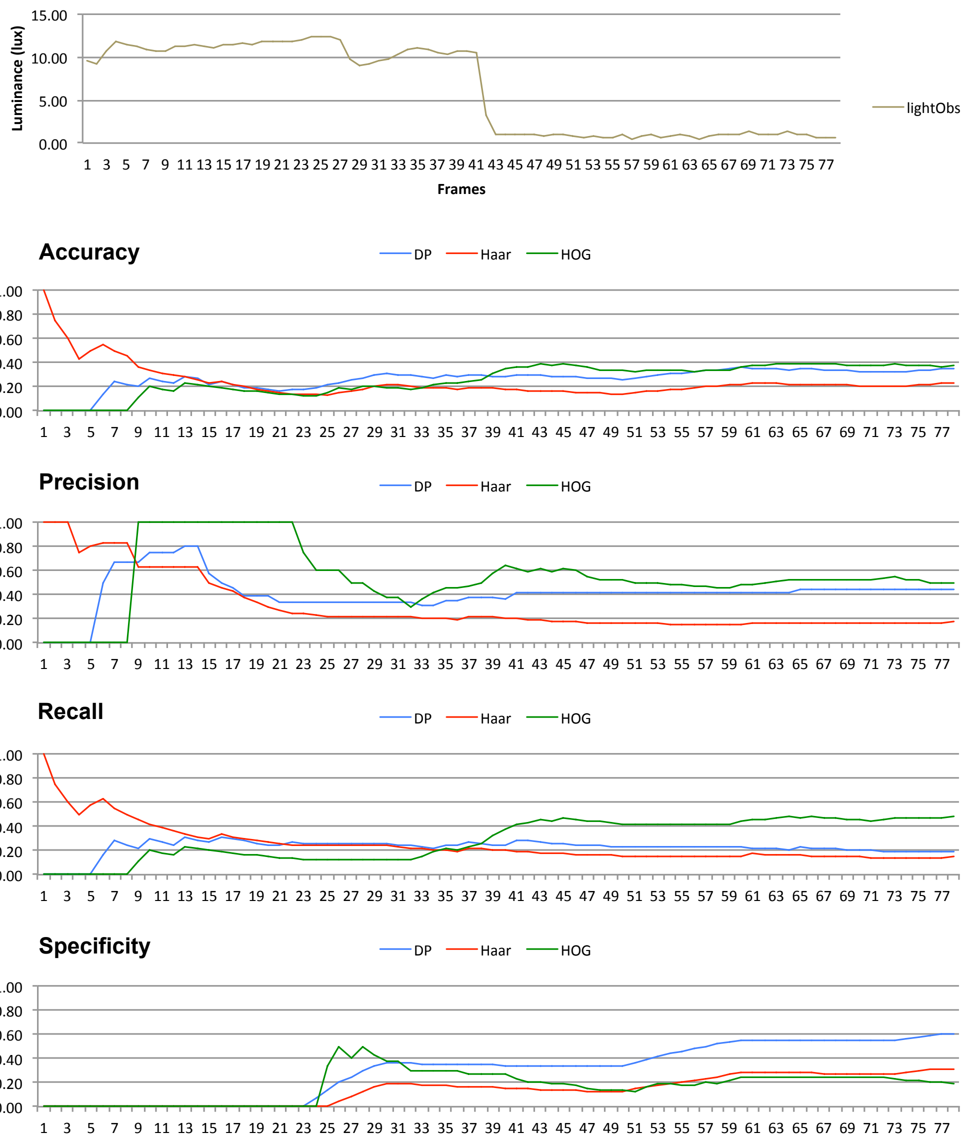


Figure 5.31: Graphics of performance metrics.

Nevertheless, the main limitation observed from the statistical measures is that our approach resulted in lower recall. This limitation may be due to limited variations in the environment conditions, which may have not covered in sufficient detail the behavior of the overall system (i.e., our test focused mainly the operation in a room with normal and dimmed light conditions). Analyzing Figure 5.30, we observe that for constant "dark-light" conditions, these observations resulted in the decision process constantly selecting action "Check-light" (i.e., using the policy graph from interactionPolicy1). This action corresponds to a sensing action instead of trying to perform detection. Comparing the outcome of this action with the two other options, we observe that performing Haar detection would result in an equivalent recall rate but with much less precision and specificity. Alternatively, using HOG detection would result in higher recall but at the same time lose specificity. Overall, assuming the issues involved in interactive features and based on the lessons learned from previous works [5], preventing erroneous detections is as relevant as the hit rate. By proving our second objective, we can claim the implicit demonstration of the first part of our hypothesis. Given that the usability of an agent is intrinsically related with not only performing the correct action but also not performing the wrong one, it becomes trivial that our approach can achieve this first objective. Nevertheless, we will plan for future work gathering more information to better corroborate this claim.

This experiment was designed with a clear intention of proofing the concept that incorporating redundancy and fall-back strategies in interaction functionalities should result in the agent's self-adaptation to its context. Therefore, these results are compared mainly in terms of the specificity of using our framework in relation to the previous results of mainstream research projects in this field – CaMeLi and GrowMeUp. In these particular examples, the feature for person detection was implemented using only the Haar-like features algorithm. Hence, using our framework can improve their specificity for this feature in near 2.5 times. Because we did not focus on implementing new classification methods that could be compared to other mainstream approaches (e.g., classifiers for people detection), a thorough comparison between the performances of different classification approaches was not covered in our study. Nevertheless, we foresee that existing systems and mainstream research results may improve using the proposed framework. For example, in a related work in progress experiment, we are using YOLO [180] for practical assisted living applications in a home environment. In this setting, we are observing YOLO has high recall for person recognition. However, regarding object recognition it falls lower than required for practical application. We believe this situation could be improved if each neural network is previously trained to perform in a specific context and then we use our framework to select the best neural network for the context of operation.

5.7 Summary discussion

In this chapter, we proposed a framework for improving the design of ASCs, which we previously identified as being extremely challenging due to the limitations imposed by current system architectures and availability of perception features that operate correctly in multiple environments (i.e. contexts). Therefore, the agents' perception features and knowledge representation need to take into account how to represent a context model. To attain this, we proposed a sequence of steps to describe the agent's behaviour, create the upper ontology (classes, properties), which captures the information that is common across the domains of all scenarios were the agent operates, create the specific scenario ontologies and finally use this knowledge to model interaction workflows that will be used in the decision process.

The goal of the proposed approach was to test our hypothesis stating that if the scenario (i.e.context) where the agent is operating changes, then the interaction workflow should automatically adapt to the new scenario, 1) to improve the usability of an interactive agent; 2) to make the human-machine interaction component of the system more robust (i.e. less fails). The experimental setup described before allowed us to obtain results that confirm the second objective. From the obtained results, two main advantages can be observed from the statistical measures. First, we obtained a similar precision to the best performing algorithm overall (i.e. HOG). Second, we achieved a higher specificity (i.e. less errors), which proves the second statement in our hypothesis. Nevertheless, the main limitation observed from the statistical measures is that our approach resulted in a lower recall. However, this may be due to limited variations in the environment conditions, which may have not covered in sufficient detail the behaviour of the overall system (i.e. as our test focused mainly the operation in a room with normal and dimmed light conditions). From analyzing figure 5.30 we observed that for the second half of data, which corresponds essentially to a continuously observed "dark-light" conditions. This observations resulted in the decision process constantly selecting action "Check-light" (i.e. using the policy graph from interactionPolicy1). This action correspond to a sensing action instead of trying to perform a detection. Comparing the outcome of this action with the two other options we observe that performing Haar detection would result in an equivalent recall rate, but with much less precision and specificity. On the other hand, using HOG detection would result in higher recall but at the same time losing specificity. Overall, assuming the issues involved in interactive features, and based on the lessons learned from previous works [5], preventing erroneous detections is as relevant as the hit rate. By proving our second objective, we can claim the implicit demonstration of the first part of our hypothesis. Given that the usability of an agent is intrinsically related with not only performing the correct action but also not performing the wrong one, it becomes trivial that our approach can achieve this first objective. Nevertheless, we will plan for future work gathering more information to corroborate this claim.

Chapter 6

Sharing knowledge between multiple Artificial Social Companions

Reaching to this point, if we consider an agent working standalone, we clearly identify some limitations on our framework. Particularly, what regards sharing knowledge between artificial social companions and with supporting infrastructures (i.e. related with integration and interoperability with other systems) and how we could offload computationally intensive tasks that are not required to be executed during runtime (e.g. calculating POMDP policy graphs - interaction worflows).

In this sense, we conceptualized our framework in a cloud based approach to tackle these constrains [28,163].

6.1 Cloud Computing

Buyya et.al, in [181] considered that "Cloud computing refers to the provision of computational resources on demand via a computer network". An example of how this concept works is found in everyday services provided by companies like Google when they provide a word processing service to clients without actually possessing the software to do it locally in their machines. Since the cloud is the underlying delivery mechanism, cloud based applications and services may support any type of software application or service in use today.

Extending that concept, cloud computing allows a functional separation between the resources used and the user's computer, usually residing outside the local network. Consumers now routinely use data intensive applications driven by cloud technology which were previously unavailable due to cost and deployment complexity.

In many companies' employees and company departments are bringing a flood of consumer technology into the workplace which raises legal compliance and security concerns for the corporation. These issues could be tackled using cloud computing, where the company would contract one service that could be used by each one of its employees.

The technical foundations of Cloud Computing are tightly coupled with the Service-Oriented Architecture (SOA) vision and web services. There refer to well-known and widely recognized and accepted as a suitable architectural style for developing modern web-based applications [182].

SOA proposes an architectural model which objectives are: improve efficiency, improve agility, and improve productivity of an enterprise considering services as the principal means over which solution logic is represented for supporting the realization of strategic aims related with service-oriented computing. A SOA implementation can involve a mixture of technologies, Application Programming Interface (API), auxiliary infrastructure extensions, and various other products.

Service-oriented computing concept becomes a distinct architectural model that has been considered by the community as one that can fully leverage the open interoperability potential of web services. For instance, if applications reusable logic is exposed as web services, the reuse potential is significantly increased.

Since service logic is now possible to be accessed through open communications framework, it becomes available to a wider range of service consumers. Given that web services provide a communications framework based on physically decoupled contracts it allows services communication to be fully standardized independently from its implementation. This allows a potentially high level of service abstraction while rising the possibility to decouple the service from proprietary implementation details.

6.2 Cloud Robotics

Cloud computing is now so popular that distinct research groups are exploring this idea applied to robotics. Robots are being prepared to connect to a cloud computing infrastructure and access vast amounts of processing power and data. This approach, which some are calling "cloud robotics," would allow robots to off-load more computational intensive tasks and even originate a more flexible and cooperative machine learning mechanism. The Cloud Robotics concept is depicted in Figure 6.1.

Currently conventional robots are limited to the built-in hardware and software, and usually dedicated to a specialized task operating a well-known and structured

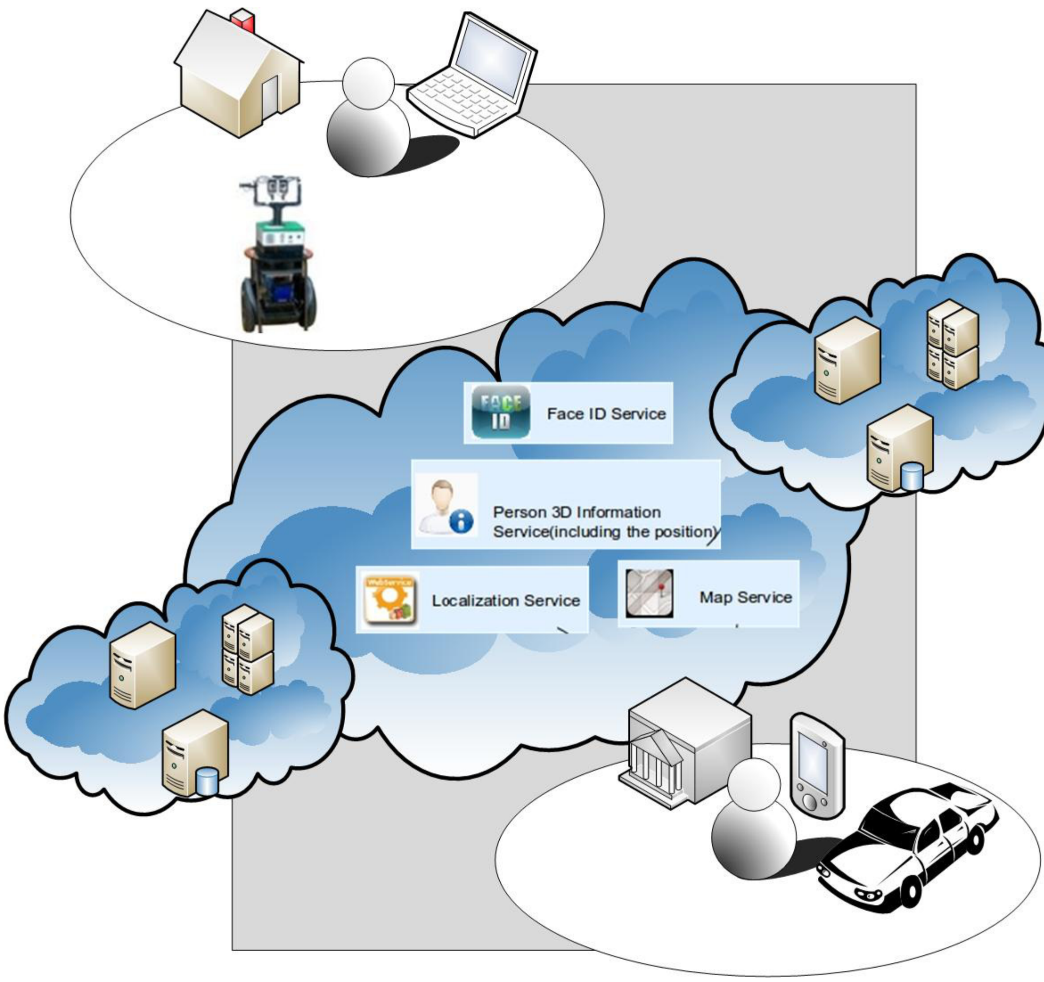


Figure 6.1: Cloud Robotics concept

environment. To these robots, every task from moving a foot to recognizing a face requires a significant amount of processing and pre-programmed information.

Robotic systems are advantageous when physical operations are required, but due to some technological and major economic reasons, there is still a gap until we can find service and assistive robots working together with the common consumer people. Nevertheless, more specific systems can be implemented and tested in some specific domains. In this section we presented some initial remarks regarding the principles of SOA and Cloud Computing as well as Cloud Robotics and in which terms the usefulness of the robotic systems can be found. Thus, our motivation is to present the benefits of applying SOA principles in the design of an infrastructure to support a robot undertaking more complex tasks. Hence, appling these principles to the implementation of artificial cognitive systems aiming to address some of the challenges in the domains of developmental robotics and behavior analysis for the next 5 to 10 years [183]. Thereby, the paper presents the conceptual design for a service robot system supported by cloud computed services focused on the exchange and learning of relevant information that might be applied in human-robot cooperative tasks.

Cloud Robotics and similar subjects (e.g. Internet Robots, Robots as Web Services, etc.) are assisting to an increase of interest by the scientific community. In one hand the basic concepts related with these topics are very attractive for the future developments in robotics, which will demand more computational resources as the tasks complexity increases. This is particularly expected to occur in the field of smart social robots. In the other hand, although these concepts are not new, with some works dated back in the 1990s, we are now in better conditions to give these approaches a renewed try.

6.2.1 The cloud robotics hype

According to James Kuffner [184], who introduced the term "cloud robotics" in 2010, "cloud-oriented robots could switch computationally heavy tasks to remote servers, relying on smaller and less power hungry on-board computers".

Steve Cousins of Willow Garage aptly summarized the idea: "No robot is an island." Cloud Robotics recognizes the wide availability of networking, incorporates elements of open-source, open-access, and crowd sourcing to greatly extend earlier concepts of "Online Robots" and "Networked Robots".

C. Costea in [185], included cloud robotics in a recently published survey on the state-of-the-art for applications and trends in mobile cloud computing.

Cloud robotics has the potential to improve performance in at least five areas:

1. Dealing with large amounts of data: indexing a global library of images, maps, and object data,

2. Cloud Computing: parallel grid computing on demand for statistical analysis, learning, and motion planning,

3. Open-Source / Open-Access: humans sharing code, data, algorithms, and hardware designs,

4. Collective Robot Learning: robots sharing trajectories, control policies, and outcomes, and

5. Crowd sourcing and call-centres: offline and on-demand human guidance for evaluation, learning, and error recovery.

6.2.2 Cloud robotics related work

B. Kim et al. [186] , discussed a ubiquitous control platform for an autonomous robot that can access distributed application logic based on recent network technologies like XML, SOAP, WSDL and UDDI. To solve the ad hoc problem of how the distributed application logic can be invoked by the robot "Web Services" are presented as the best solution. Web services can speed development with a more flexible infrastructure where multiple services can work together to provide data and services for the application.

L. Vasiliu et al. [187], proposed a solution demonstrating how semantic web and web services could be applied on robotics in order to facilitate cooperation between robots for joint tasks execution. By implementing Semantic Web Services for isolated robots within a network viewpoint they can be regarded as distributed web services that communicate between each other semantically, allowing a real time operation.

Y. Chen et al [188], presented research on service-oriented robotics computing and their design, and the implementation and evaluation of Robot as a Service (RaaS) unit. Similarly, we presented in [28, 29], a conceptual design for a service robot system supported by cloud computed services. The mobile robot acted as a service provider and consumer. Services were published into a common service repository, thus making them discoverable by other remote services. A service could correspond to a skill learned by the robot, which would be published in the cloud and be usable by other robotic agents. The robot relied on the cloud to obtain new services, downloading the requested skill from the cloud repository.

Follow a presentation of some well-known Cloud Robotics projects. In a larger scale, RoboEarth [189] is a European project led by the Eindhoven University of

Technology, in the Netherlands, to develop a "World Wide Web for robots", a giant database where robots can share information about objects, environments, and tasks. Considering the large amount of information shared on such a vast scale, and with businesses and academics contributing independently on a common language platform, RoboEarth has the potential to provide a powerful source to leverage any robot's 3D sensing, acting and learning capabilities.

As for using cloud robotics to decrease the demand on local computational resources, DAvinCi [190] is a software framework attempting to offload data and computationally intensive workloads from the onboard resources of robots to a backend cluster system shared by multiple clients. This framework performs computationally intensive tasks and enables the exchanging of useful data obtained from local preprocessing. Sharing the same motivation, ASORO (A-Star Social Robotics Laboratory) (http://www.asoro.astar.edu.sg/research_main.html) from a Singapore researchers group tries built a cloud-computing infrastructure that allows robots to generate 3-D maps of their environments much faster than they could with their onboard computers.

Google engineers developed Android-powered robot software that allows a smartphone to control robots based on platforms like Lego Mindstorms, iRobot create, and Vex Pro.

Researchers at the Laboratory of Analysis and Architecture of Systems, in Toulouse, France, are creating "user manual" repositories of everyday objects to help robots with manipulation tasks.

Finally, at a children's hospital in Italy, NAO (http://www.aldebaran-robotics.com/) humanoid robots, created by the French firm Aldebaran Robotics, will rely on a cloud infrastructure to perform speech recognition, face detection, and other tasks that might help improve their interaction with patients.

Cloud robotics allows for the bidirectional interchange of information about the local context of the robot and the global smart space context, involving the robotic nodes, the smart environment and also the end-user. However, robotic integration and interoperability within the smart space involves the actual understanding of the events, sequences of events and the collection of episodes composing each context, and, concurrently, the deliberate blurring of the differentiation between the smart environment and the robotic nodes - this represents the "inside-out robot" paradigm introduced by Crowley et al. [60,191,192].

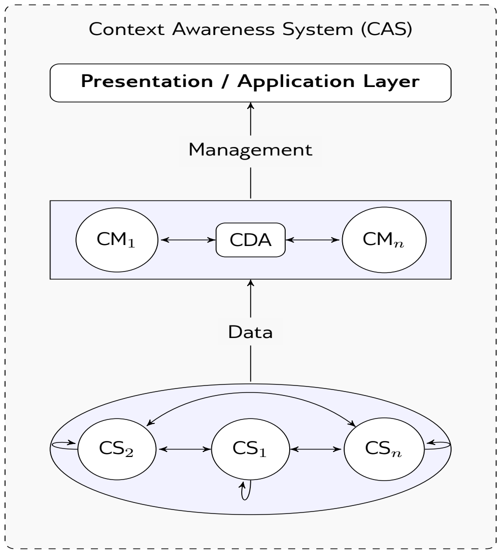


Figure 6.2: Context Awareness System general overview

6.3 Technical Adaptations for Context Awareness

In spite of important breakthroughs and progresses assisted in service robotics, there are challenges still need to be tackled in order to obtain sophisticated solutions like the one described above. The proposed scenario would require that all the involved robotic systems could be capable of networking and sharing contextual information about the user.

The importance of context awareness in service robots is motivated by the dynamical characteristics of the information related with a person's activities. In order to obtain a ubiquitous robotized environment the capability of being sensitive to changes in context is determinant to correctly adapt system's reaction to the moment intentions of the user.

Context awareness has being studied in the scope of mobility systems, usually related with personal devices like Personal Digital Assistants and Smart Phones, and how they adapt to changes in network connections. The application of context awareness in robotics will require the integration of much more information sources than the ones available in the first examples.

However, the particular case of sharing context information between distinct and heterogeneous networks without having to develop an entire system from scratch is still a challenge.

In Figure 6.2, a schematic is presented illustrating the concept behind the context information sharing and management between different Context Management Systems (CMS). In the figure the CMSs represent the different systems corresponding to a robot and the smartroom, which manage their own contextual information and share that information between them using a bridge like adapter called Context Discovery Adapter (CDA). This approach was firstly proposed by P. Pawar, et.al in [193], where a Context Discovery Adapter was presented as a solution to bridge two different CMSs in two distinct and heterogeneous networks (one dedicated to ad-hoc connections and other focused on mobile networks). In the scope of the cooperation between the network of robots and the smart-room, an approach like CDA could prove advantageous, although it would require extending the current implementation to a more diversified group of devices.

In order to satisfy the needs of similar scenarios, the system could be though as the composition of a set of services provided by each robot, thus making it following service oriented architecture. We propose a service oriented architecture that will address the collaboration between a group of robots and a smart-room taking into consideration context sharing and management.

The contextual information available in each robot will be combined to produce a representation of the user's situation. In Figure 6.3 we provide a representation of the system's architecture design. The objective of distribute the robots workload according to different infrastructural nodes, the routing mechanism should consist in a set of brokers, according to the type of data to be processed. The brokers encapsulate a group of messaging queues for distributing the load among different nodes. The architecture proposed in this paper aims to integrate the mobile robot and associated functional capabilities as services. The mobile robot will act as service provider and consumer. Services will be published into a common service repository, thus making them discoverable by other remote services. A service could correspond to a skill learned by the robot, which would be published in the cloud and be usable/learned by other robotic agents (even in a remote location).

In order to keep up with user's service requests and for the sake of the system's "evolution", the robot will rely on the cloud to obtain new services. In case it cannot fulfill its user request the robot will connect to the cloud skill repository and download the requested skill. This mechanism will allow for a dynamic behavior, and will allow for a lower degree of computational requirement in the robots hardware, since the services can be loaded and unloaded according to needs of the user. Figure 6.4 illustrates the mechanism of reacting to a service request that is unknown to the robot. Besides the possibility of fetching new skills from the cloud, the robot will have mechanisms of learning. The learning mechanism will allow the user to teach him new skills that may be very specific or may not be present in the cloud skills repository. In Figure 6.5 we illustrate the learning and publish procedure. The cloud

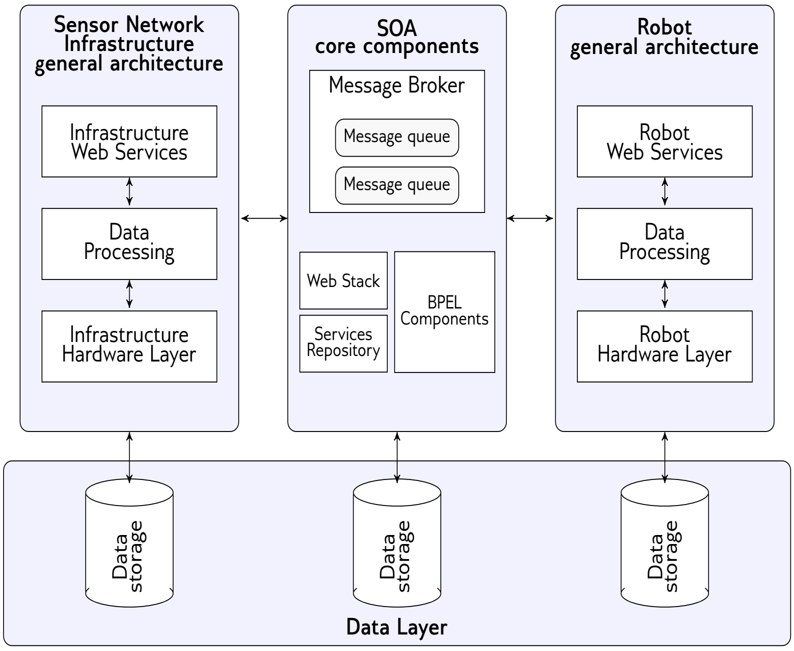


Figure 6.3: Generic framework for cloud robotics

skills repository consists in a public area where different robotic systems would load their different learned skills in order to share them with sibling robotic platforms.

6.4 Knowledge matching

According to Kalfoglou and Schorlemmer in [194], knowledge matching is the process of determining the intersections between two distinct representations (e.g. ontologies), with the main goal of finding a mapping function between various representations in order to enable a common understanding that support communication among existing and new domains. The Knowledge Matching problem arises from the conjugation of two prime conditions. On one hand, describing all the *things* and *stuff* in one unique knowledge representation is unattainable. On the other hand, the distributed essence of the knowledge, allow an increasing number of representations to be created using different terms and taxonomies that may represent similar domains. Therefore, the knowledge matching problem address the interoperability between knowledge representations (e.g. ontologies), which can only be possible if correspondences between their concepts have been identified and established. This procedure was partially presented by Ehrig et al. in [195] on their ontology matching approach and corresponds to a canonical technique that is typically followed by

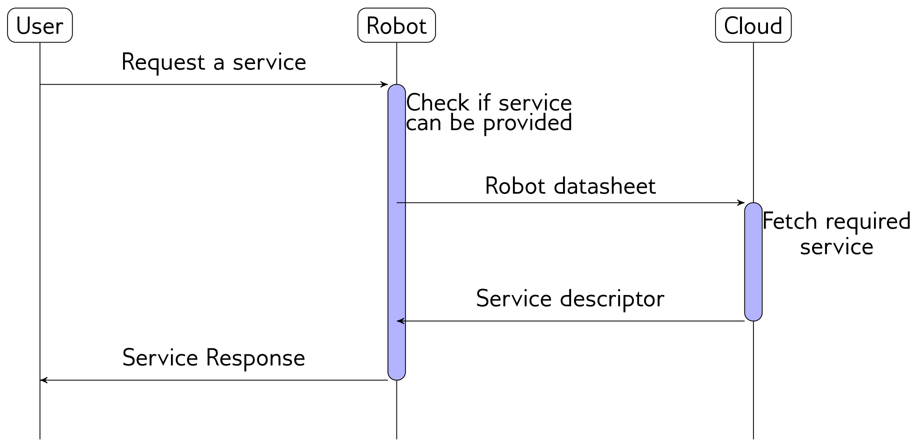


Figure 6.4: Obtaining new skills from the cloud.

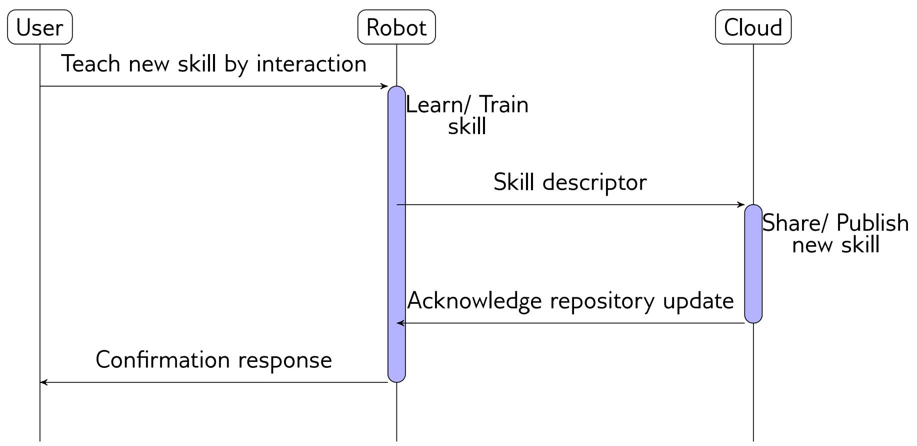


Figure 6.5: Teaching and sharing new skills.

ontology matching frameworks.

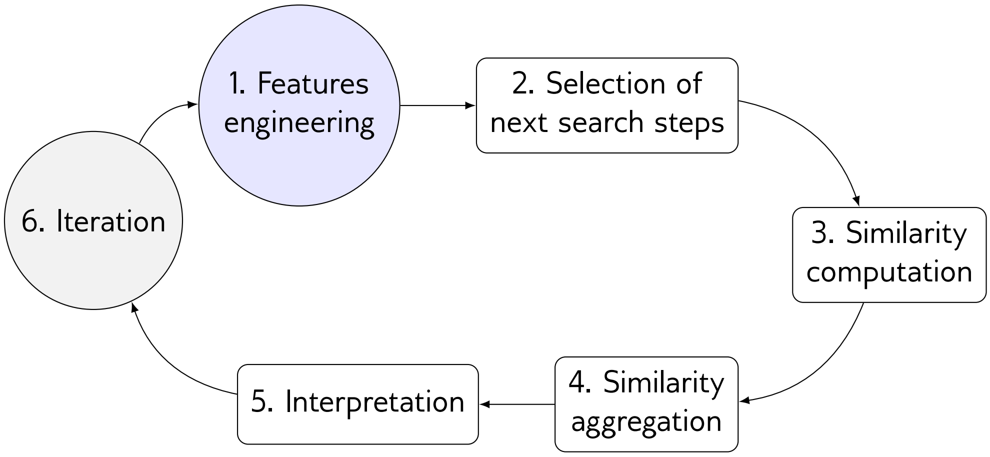


Figure 6.6: Ontology matching process

Figure 6.6 illustrates the six main steps for the process. The ontology matching process is started with two ontologies, which are going to be mapped onto one another, at the end of the approach. *Feature Engineering* transforms the initial representation of ontologies into a format digestible for the similarity calculations. The derivation of ontology mappings takes place in a search space of candidate mappings realized through the *Selection of Next Search Steps*. This step may choose, to compute the similarity of a restricted subset of candidate concepts pairs {(e, f)|e $\in O_1$, f $\in O_2$} and to ignore others. The *Similarity Computation* determines similarity values between candidate mappings based on their definitions in two matching ontologies. These different similarity values for one candidate pair must be aggregated into a single aggregated similarity value during the *Similarity Aggregation* step. *Interpretation* uses the individual or aggregated similarity values to derive mappings between entities from the two comparing ontologies and then several algorithms perform *Iteration* over the whole process in order to bootstrap the amount of structural knowledge. Iteration may stop when no new mappings are proposed.

Because ontologies can be compared from many different points of view, many tools and methods have been proposed over the recent decade. Following, we will summarize the relevant works related to this topic.

6.4.1 Ontology matching tools and heuristic methods

Most of the survey presented in [194] can be used as a starting point to our study and a summary of their conclusions will be presented here.

Tools

Noy and Musen developed a series of tools for performing ontology mapping, alignment and versioning. These tools are SMART [196], PROMPT [197] and PROMPTDIFF [198]. The tools used linguistic similarity matches between concepts for initiating the merging or alignment process, and then used the underlying ontological structures of the Protégé–2000 environment (classes, slots, facets) to inform a set of heuristics for identifying further matches between the ontologies.

McGuinness et al. [199] developed a similar tool for the Ontolingua editor. Chimaera would analyze the ontologies to be merged, and if linguistic matches were found, the merge would be done automatically, otherwise the user would be prompted for further action.

Doan et al. [200] developed a system, GLUE, which employed machine learning techniques to find mappings. Given two ontologies, for each concept in one ontology, GLUE would find the most similar concept in the other ontology using probabilistic definitions of several practical similarity measures.

Lacher and Groh [201] presented CAIMAN, another system which used machine-learning for ontology mapping. The authors elaborated on a scenario where members of a community would like to keep their own perspective on a community repository. Their mapping mechanism used machine learning techniques for text classification.

Prasad et al. [202] presented a mapping mechanism which used text classification techniques as part of their web-based system for automatic notification of information technology talks (ITTalks).

Mitra and Wiederhold [203] developed the ONtology compositION system (ONION) which provides an articulation generator for resolving heterogeneity in different ontologies.

Compatangelo and Meisel [204] developed a system, ConcepTool, which adopts a description logic approach to formalize a class-centred, enhanced entity-relationship model. Their work aims to facilitate knowledge sharing, and ConcepTool is an interactive analysis tool that guides the analyst in aligning two ontologies.

Heuristics and Ontology Matching

According to [194] they found that using heuristics was among the most popular techniques for performing ontology matching. In their study they claimed *"heuristics are cheap to develop and easy to deploy, and support automation. However, the main problem with heuristics is that they are easily defeasible. Even well-crafted heuristics for a particular case can fail in similar situations"*. Heuristic methods refer to experience-based techniques to problem solving. These methods are advantageous in finding acceptable approximations, when an exhaustive search for a solution is impractical [205]. Heuristic methods can be either deterministic or probabilistic. For deterministic methods, given a search space and an evaluation function, some would always return the same solution (e.g. dynamic programming), while others could generate different solutions based on the initial configuration or starting point (e.g. a greedy algorithm or the hill-climbing technique). Probabilistic methods incorporate random variation into the search for optimal solutions. These methods (e.g. simulated annealing) could return different final solutions even when given the same initial configuration. Modern heuristic methods were given a special attention, including simulated annealing, tabu search and genetic algorithms. These techniques can be divided in non-population and population approaches. In non-population approaches (i.e. simulating annealing, tabu search) each method relies on a single solution as the basis for future exploration. They either process complete solutions in their entirety, or they construct the final solution from smaller building blocks. Simulated annealing and tabu search process complete solutions, and you could obtain a potential answer (although quite likely a sub-optimal one) by stopping any of these methods at any particular iteration. They always have a single "current best" solution stored that they try to improve in the next step. In population methods (i.e. genetic algorithms) a different idea has been followed. These approaches abandon the idea of processing only a single solution, and consider a population of solutions. In spite of giving the idea that these only perform a parallel computation of solutions, an additional component that can make population-based algorithms essentially different from other problem-solving methods. These methods use the concept of competition between solutions in a population, simulating the evolutionary process similar to natural evolution.

The application of heuristic methods, on ontology matching, could be observed for example in [206], where the authors presented a genetic algorithm-based optimization procedure for the ontology matching problem. They modeled the problem of ontology matching as an optimization problem of the mapping between two distinct ontologies. In their approach, they considered each solution as a one-dimensional integer array, representing the mapping between the two ontologies (i.e. concept i in $Ontology_1$ map concept j in $Ontology_2$). Each generated solution was then evaluated for the Tversk's similarity [207] function as fitness function, and the best solution was chosen to create the next generation of solutions. They conducted their experiments on OAEI2005 benchmark test suit, and used standard evaluation measures to assess the results (i.e. Precision, Recall and FMeasure) [208]. Another example, in the work conducted by Ehrig and Staab in [195], the Quick Ontology Mapping (QOM) was implemented as a way to trade off between effectiveness and efficiency of the mapping generation algorithms. They showed QOM to have a lower run-time complexity than existing prominent approaches like PROMPT, Anchor-PROMPT and GLUE.

6.5 Experimenting with Knowledge Matching

In the scope of this thesis, we conducted a comparison experiment between two different meta-heuristic techniques applied to the problem of Knowledge Matching. The objectives for this experiment were the implementation, simulation, evaluation and comparison between the Simulated Annealing and Genetic Algorithm approaches for the Knowledge Matching problem.

6.5.1 Environment

Much of the experimental results depend on the systems where simulations are executed. For the sake of future benchmarking we will provide a general description of our system configuration.

Hardware specifications The simulations were executed in a personal computer with a Intel(R) Core(TM) i7 CPU Q720 @1.60Ghz, and 6GB of installed memory (RAM). Other details about the hardware would be redundant, since these two characteristics are most influential. This is easily explained due to the fact that we loaded the ontology statement into a memory repository, avoiding hard drive reads and writes. The computational processing was provided by the group of the CPU 8 cores, avoiding any kind of additional parallel computing technology.

Programming Environment The algorithms were implemented using Java. The operative system was Microsoft Windows 7 64-bits Professional. For the sake of development agility we used the integrated development environment (IDE) Eclipse Indigo with the Maven extension, for project automation. The choice of programming language was much motivated because of the existence of tools and libraries that could help in the implementation. For the ontology parsing, storage and query we used Sesame from openrdf. The genetic algorithm implementation used WatchMaker Framework, provided with Apache Mahout. This pair of tools were very helpful in the implementation stage of this work. Regarding the simulated annealing algorithm no additional tool was used, its implementation followed the pseudo-code provided in the literature.

6.5.2 Ontology abstract model

Before understanding how to solve the knowledge matching problem, we need to first understand in more detail what are the building blocks of an ontology representation, which we have seen in the previous section that these correspond to *concepts*.

The matching problem resides on finding out what concepts are equivalent in two different representations. Hence, a simplified approach to compare two different ontologies will be to analyse the features of each concept. They are three main features for each and every concept: Name, Property and Intance, which according to some authors, as proposed in [206], are known as Intention for local information.

Overall, by adopting the same notation as these authors proposed, we will have that an ontology with n concepts is modelled as a tuple : $OntologyModel = (V_C, OntoInT)$, where $V_C = \{v_{c_i} | v_{c_i} = i, 1 <= i <= n\}$ is a set of sequence number of concept, where v_{c_i} denotes that the i*th* concept in the ontology is ranked with a sequence number of i; and $OntoInT_{V_C} =< v_{c_i}, \mathit{Type}, x_{c_i} >$ is a set of intentional features of the ontology, where $1 <= i <= n$, $\mathit{Type} = 'N', 'P', 'I'$ and

$$x_{c_i} = \begin{cases} n_{c_i} \in N_{c_i}, & \text{if } \mathit{Type} \text{ is 'N'} \\ p_{c_i} \in P_{c_i}, & \text{if } \mathit{Type} \text{ is 'P'} \\ i_{c_i} \in I_{c_i}, & \text{if } \mathit{Type} \text{ is 'I'} \end{cases}$$

The ontology matching problem then becomes a search for a suitable mapping function that maximizes the similarity between two distinct ontology representations.

6.5.3 Ontology matching algorithms

In order to address this problem, we were inspired by previous approaches used by Rodríguez and Egenhofer in [209] to address the problem of determining the semantic similarity among entity classes from different ontologies; and by Junli Wang et.al in [206], where they took into consideration the intentional and extensional features for a given concept. Consequently, we followed an evolutionary algorithm were each individual in the population (i.e. candidate solutions) was represented as an array of integer numbers obtained by a stochastic process.

Simulated Annealing

Simulated Annealing – also known as Monte Carlo annealing, statistical cooling, probabilistic hill-climbing, stochastic relaxation, and probabilistic exchange algorithm – is a local search algorithm, based on an analogy taken from thermodynamics. Often it is introduced as an improved version of hill-climbing algorithm, i.e., a hill-climbing algorithm that is capable of randomly move *dowhill* in such way that yields both completeness and efficiency. In [149, 205] this algorithm is introduced with the examples of crystal formation and metallurgy, i.e., annealing is the process used to temper or harden metals by heating them to a high temperature and then gradually cooling them, thus allowing a the material to reach a low-energy crystalline state. Each step of the algorithm attempts to replace the current solution by a random solution. The new solution may then be accepted with a probability that depends both on the difference between the corresponding function values and also on a global parameter T (called the temperature), that is gradually decreased during the process. The dependency is such that the choice between the previous and current solution is almost random when T is large, but increasingly selects the better or "downhill" solution (for a minimization problem) as T goes to zero. The allowance for "uphill" moves potentially saves the method from becoming stuck at local optima — which are the bane of greedier methods. Most implementations of simulated annealing follow a simple sequence of steps that is described in [205]. We start by referring the number of iterations, i.e. number of candidate solutions tested for each value of temperature. We chose this value as 400, in order to generate enough random candidate solutions and converge to a better solution, i.e. evaluating a larger number of candidate solution would increase the chance of finding the best candidate. The Temperature parameter is used to determine the probability of acceptance of a worse candidate solution. In our experiment, we used a very simple cool down function, where we considered the Temperature decrease at a rate 1/time.

Genetic Algorithm

Genetic algorithms, as described for example in [205], provide a conceptual simplicity that is distinctly different for solving problems when compared to classical methods. We start with a population of initial solutions, which could have been generated by sampling randomly from the search space S. The candidate solutions can be described as vectors, or whatever representation and data structure we select. The evaluation function can be used to determine the relative merit of each of our initial solutions. The chosen evaluation function must be capable of differentiating between two individuals and rank one solution ahead of another. Those solutions that are better, as determined by the evaluation function, are favoured to become parents for the next generation of offspring. The approach used, to create this offspring, often takes parts of two parents and put them together to form an offspring. Another way of using two solutions to generate a new possibility occurs when facing continuous optimization problems. In these cases we can blend parameters of both parents, essentially performing a weighted average component by component. Our Genetic Algorithm (GA) implementation used the WatchMaker Framework[37], provided with Apache Mahout. The Elitist strategy was utilized to keep unchanged for the next

[37] `http://watchmaker.uncommons.org`

generation the 5% of the population with best fitness score. Moreover, we used as parameters for the genetic algorithm: population size of 1000; 3 crossover points; 90% of crossover probability; 0.1% of mutation probability; and allowed the algorithm to iterate through a maximum of 400 generations. For the fitness function 6.1, we used $\alpha = 0.8$ and $\beta = 0.2$, as the weights for each component of the unmatched elements. The parameters for both algorithms were determined heuristically.

Solution preparation

The search space for generating a given individual was bounded by number of concepts of both ontologies. Each candidate solution was an array of length given by the number of concepts in the first ontology O_1 whilst the values for element could vary between 0 and the number of concepts in the second ontology O_2. Value repetition throughout the candidate solution is allowed. Each concept in one ontology could be mapped in more than one concept in the other. For the simulated annealing a single candidate solution was computed at the time, being randomly generated at each iteration. For the genetic algorithm, given its population characteristics, a number of candidate solutions were computed at the beginning, according the population size, and then changed according the mutation and cross-over parameters.

Concepts similarity based evaluation/fitness functions

The evaluation/fitness function was based in the implementation of the Tversky's Similarity. Amos Tversky proposed a new set-theoretical approach to similarity in which objects are represented as collection of features, and similarity is described as a feature-matching process. The motivation and detailed description of Tversky's method is presented in [207]. This approach have been adopted in several works to measure the matching level between two different ontologies [195, 206, 209]. In his method he proposed that any function *f* that satisfies the Matching and Monotonicity assumptions [207] can be called a *matching function*. This function measures the matching degree to which two objects/concepts – viewed as a set of features – match each other. Thus, the assessment of similarity is presented as a feature-matching process, formulated in terms of the mathematical set-theory rather than in terms of a geometrical concept of distance. In [207] he presented the *contrast model* and the *ratio model* for a matching function. In spite of a more detailed explanation was presented for the contrast model, the ratio model holds the advantage to our application, since the similarity is normalized so that S lies between 0 and 1. The ratio model we applied is modeled according to the equation 6.1,

$$
\begin{aligned}
&Similarity_{O_1,O_2}(M) = \\
&= \frac{f((F_{O_1} \in F_{O_2})|M)}{f((F_{O_1} \in F_{O_2})|M) + \alpha \times f((F_{O_1} - F_{O_2})|M) + \beta \times f((F_{O_2} - F_{O_1})|M)}
\end{aligned}
\tag{6.1}
$$

where $f((F_{O_1}F_{O_2})|M)$ are the matched elements of both ontologies for the mapping M, $f((F_{O_1} - F_{O_2})|M)$ and $f((F_{O_2} - F_{O_1})|M)$ are respectively two sets of the unmatched elements with respect to the mapping M. The relative importance of the two unmatched feature sets is given by the parameters α and β, which range between 0 and 1. Function *f* defines the cardinality of set. This model encompasses a wide variety of similarity models, which differ in the form of the matching function *f* and in the weights assigned to its arguments, hence it generalizes for several set-theoretical models of similarity proposed in literature. An additional consideration, that consider also Additive Clustering Models [210] can be advantageous to the problem of Ontology Matching. We assume that the weighted addition of the properties of any two objects is sufficient to determine the effective similarity between them. This particularity allows us to divide the problem of matching a complex set of features into several of minor complexity. Moreover, we defined a matching rule to verify that F_{O_1} and F_{O_2} will be matched according $f((F_{O_1}F_{O_2})|M)$. This rule regards lexical matching as a direct correspondence matching between strings in the intentional features sets, avoiding for now using thesaurus or other more complex methods for determining synonyms between words.

6.5.4 Benchmarks

The data used for benchmarking was the Ontology Alignment Evaluation Initiative (OAEI) 2005 benchmark test suit [211]. The OAEI is a coordinated international initiative that has been set up for organizing evaluation of ontology matching algorithms. This initiative found motivation in the need to assess the increasing number of methods available for schema matching and ontology integration. The main goal of OAEI is to be able to compare systems and algorithms on the same basis and trace conclusions about the best approaches. The detailed description of the benchmark test suit is available in [211][38]. We selected the domain of its first test, which is Bibliographic references, represented by an ontology containing 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals.

[38] `http://oaei.ontologymatching.org/2015/benchmarks/index.html`

6.5.5 Performance assessment

Performance assessment is an important step to any study involving comparison with other results. In our study, we based our metrics in three performance assessment measures often used in pattern recognition and information retrieval, namely *Precision* (Pre), *Recall* (Rec) and the traditional *f-measure* (F1) [208]. *Precision* can be seen as a measure of exactness or quality, whereas *Recall* can be considered as a measure of completeness or quantity. In other words, a high precision means that an algorithm returned more relevant results than irrelevant, whereas high recall means that an algorithm returned most of the relevant results. The two measures are sometimes used together in the *F1 Score* (or f-measure) to provide a single measurement for a system. Precision and Recall are the quotient of the matched region by respectively the right red circled group and the left green circled group. These assessment measures are given by:

$$Pre = \frac{f(F_{O_1} \cap F_{O_2}|M)}{f(F_{O_1} \cap F_{O_2}|M) + (F_{O_2} - F_{O_1}|M)} \tag{6.2}$$

$$Rec = \frac{f(F_{O_1} \cap F_{O_2}|M)}{f(F_{O_1} \cap F_{O_2}|M) + (F_{O_1} - F_{O_2}|M)} \tag{6.3}$$

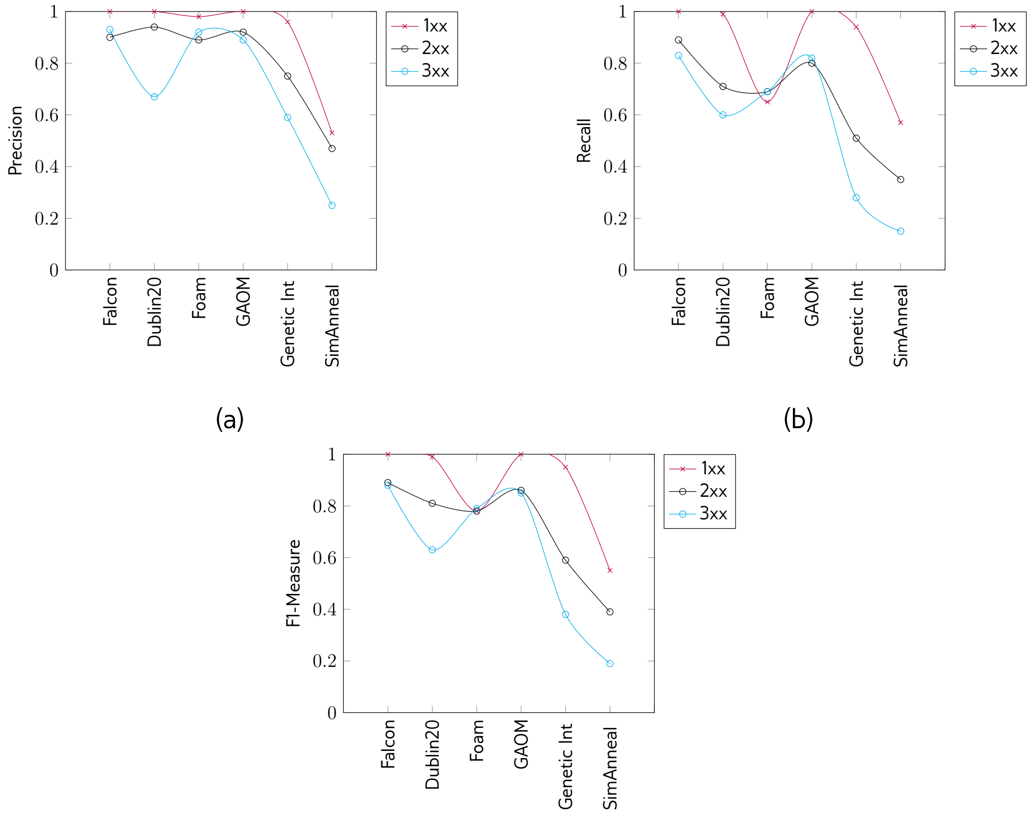
$$F1 = 2 \times \frac{(Pre \times Rec)}{(Pre + Rec)} \tag{6.4}$$

The obtained results are summarized in Table 6.1 and depicted in Figure 6.7.

Table 6.1: Simulations results table

Measure	Benchmark	Falcon	Dublin20	Foam	GAOM	Genetic Int	SimAnneal
	1xx	1.00	1.00	0.98	1.00	0.96	0.53
Precision	2xx	0.90	0.94	0.89	0.92	0.75	0.47
	3xx	0.93	0.67	0.92	0.89	0.59	0.25
	1xx	1.00	0.99	0.65	1.00	0.94	0.57
Recall	2xx	0.89	0.71	0.69	0.80	0.51	0.35
	3xx	0.83	0.60	0.69	0.82	0.28	0.15
	1xx	1.00	0.99	0.78	1.00	0.95	0.55
F1	2xx	0.89	0.81	0.78	0.86	0.59	0.39
	3xx	0.88	0.63	0.79	0.85	0.38	0.19

The last figures present the obtained results for our simulated annealing and genetic algorithms and for several algorithms, in which falcon, dublin20 and foam are the three best matching algorithms mentioned in [211].



(a) (b) (c)

Figure 6.7: Simulation results for: 6.7a Precision, 6.7b Recall and 6.7c F1

6.6 Summary discussion

In this chapter, we introduced a potential approach to share knowledge between multiple ASCs, which could benefit from sharing information or in cases that heterogeneous teams would be required to cooperate regarding a common objective. This led to us in the direction of cloud computing technologies, and the recently introduced concept of cloud robotics. However, a fundamental concern in these approaches is regarding interoperability. It would be improbable two agents could cooperate without a common understanding of the “world”. Therefore, it was relevant to understand and explore solutions for the knowledge matching problem. To this avail, we conducted an experiment where we compared some algorithm implementations and how they perform in a common dataset. From Table 6.1 and Figure 6.7, we can observe an overall better performance from the genetic algorithm regarding the simulated annealing algorithm. However, it shall be expected an improvement in the outputs for both approaches if their parameters were fine tuned by an expert. Disregarding this particularity, in practical terms, the genetic algorithm revealed to be much more prone to converge for better candidate solutions in shorter time and requiring less human intervention. Nevertheless, both algorithms can provide a solution for the problem. The simulated annealing could be advantageous in a scenario where time is not a constrain but computational power is limited. A slight evaluation to computational resources usage, while both algorithms were running, revealed a usage of approximately 90% of the 8 available cores, for the genetic algorithm execution, whilst a 50% usage of 1 core for the simulated annealing algorithm execution. Our algorithm implementations obtained very good results, outperforming Foam in Recall and F1 scores. The slight disparity in the results observed for datasets 2xx and 3xx may be related with our imposed restriction of modelling the ontology using the intentional features only. The extension of the model to take into consideration the relationship between concepts could be considered for future work.

Chapter 7

Conclusion and Future Work

The problem addressed in this thesis was that of achieving natural interaction between humans and machines. Our main purpose was to study how artificial social companions' performance is affected when interaction workflows are incorporated in its information model and decision-making process.

To achieve such objective, we must overcome current limitations of information sharing in decision processes and find computational effective ways to build complex decision processes involved in the interaction process.

We thought that part of our solution could incorporate redundancy and fall-back strategies in terms of interaction functionalities that could result in the agent's self-adaptation to its context (e.g. user model and environment conditions). This would also result in less errors during operation.

We designed and implemented two different approaches of artificial social companions [4–8]. Both approaches aimed to operate as assistive technology in real-world indoor environments. Their primary mission was to help older adults in managing activities of their daily life and connected with their social circle. In CaMeLi[39], we designed and implemented a Virtual Partner (ViP) able to show a wide variety of human-like understanding and responding and solicit the appropriate services to answer the user's needs/requests offering real time complimentary feedback through voice and a wide spectrum of animated facial expressions. On the other hand, in GrowMeUp[40] we designed and implemented an affordable service robotic system able to learn the older persons needs and habits over time and enhance ('grow up'/scale up) its functionality to compensate for the elder's degradation of abilities, to support, encourage and engage the older persons to stay longer active, independent and socially

[39] http://www.cameli.eu/
[40] http://www.growmeup.eu/

involved, in carrying out their daily life at home.

In both systems, we implemented various degrees of human interaction and autonomy that aimed to perform cognitive-like functions and accomplish real-time time goals in terms of interaction and self-sufficiency. This means, user and agent could interact through multiple modalities, which included speech commands, gestures, touch screen, and human interface devices (e.g. keyboard and computer mouse). In both cases, we dedicated most of our efforts developing perception capabilities, user interfaces and integration of these core components that resulted in two fully functional systems.

The results, from our different experiments, confirmed that our approach can improve agent's performance, maintaining precision while improving specificity. Although, we have still some challenges in designing and implementing interaction workflows. Involving the users during the design process allowed us capturing relevant needs and requirements that would tailor the functionalities of the agents. However, implementing interactivity based on pre-defined user scenarios and action scripts is not sufficient to take into account uncertainty associated with noisy inputs, variation in the conditions of the operating environment, or unclear expectations from the user. It is not realistic to expect that users always use the same interaction patterns, never commit a mistake or the environment conditions are unchangeable.

7.1 Contributions

The most relevant contributions of this work are the proposed knowledge model and the decision process model. The integration of these two models results, conceptually, in automatizing the representation of knowledge and in auto-adapting the decision process. We recall the conclusions from our literature survey, in which we observed that current approaches addressing the problem of HMI still find limitations at different levels. Moreover, the lack of redundancy and fall-back strategies in terms of interaction functionalities result often in unexpected system behaviours (e.g. faults, errors or fails) creating barriers introducing new interaction modalities and making interaction more natural.

7.1.1 Publications

The list of publications regarding the relevant topics for this work can be enumerated as follows:

Peer reviewed International Journals

1. J. Quintas, G. S. Martins, L. Santos, P. Menezes, J. Dias, "Toward a Context-Aware Human-Robot Interaction Framework Based on Cognitive Development", in IEEE Transactions on Systems, Man and Cybernetics, PP(99), May 2018, DOI: 10.1109/TSMC.2018.2833384.
2. J. Quintas, P. Menezes, J. Dias, "Information Model and Architecture Specification for Context Awareness Interaction Decision Support in Cyber-Physical Human–Machine Systems", in IEEE Transactions on Human-Machine Systems, 47 (3), 323-331, December 2016, DOI: 10.1109/THMS.2016.2634923.

Peer reviewed Conferences

1. J. Quintas, P. Menezes, J. Dias, "From User Expectations to Agents Behaviour", Autonomous Robot Ontology Workshop (AROW 2017) held at 26^{th} IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2017), Lisboa, Portugal, August, 2017.
2. J. Quintas, P. Menezes, J. Dias, "Interoperability in Cloud Robotics - Developing and Matching Knowledge Information Models for Heterogenous Multi-Robot Systems". 26^{th} IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2017) Special Session on "Cloud Technologies: Empowering Robots to Connect Society", Lisboa, Portugal, August, 2017.
3. Quintas, João and Menezes, Paulo and Dias, Jorge, "Context-Based Decision System for Human-Machine Interaction Applications", in Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on, Oct. 2016, pp. 3906–3911.
4. Quintas, João and Menezes, Paulo and Dias, Jorge, "Auto-Adaptive Interactive Systems for Active and Assisted Living Applications", in Technological Innovation for Cyber-Physical Systems. DoCEIS 2016. IFIP Advances in Information and Communication Technology, Apr. 2016, pp. 161–168.
5. Quintas, João and Almeida, Luıs and Sousa, Elısio and Menezes, Paulo, "A context-aware immersive interface for teleoperation of mobile robots", in Sciences and Technologies of Interaction - SciTecIN, Nov. 2015.
6. P. Menezes, J. Quintas, J. Dias, "The Role of Context Information in Human-Robot Interaction", RoMan 2014 Workshop on Interactive Robots for aging and/or impaired people (WIRAIP2014) held at 23^{rd} IEEE International Symposium on Robot And Human Interactive Communication (IEEE RO-MAN 2014), Edinburgh, August, 2014.

7. J. Quintas, P. Menezes, J. Dias, "Context-based perception and understanding of human intentions". 22^{nd} IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2013), Gyeongju, Korea, August, 2013, pp. 346-347.

8. J. Quintas, P. Menezes, J. Dias, "Context-aware cooperation between human and robotic teams in catastrophic incidents". 22^{nd} IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2013), Gyeongju, Korea, August, 2013, pp. 308-309.

9. J. Quintas, L. Almeida, M. Brito, G. Quintela, P. Menezes, J. Dias, "Context-based understanding of interaction intentions". in Proceedings of the 21^{st} IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2012), Paris, France, August, 2012, pp. 515-520.

10. J. Quintas, P. Menezes, J. Dias, "Cloud Robotics: Towards Context Aware Robotic Networks", in Proceedings of IASTED, The 16^{th} IASTED International Conference on Robotics (Robo 2011), Pittsburgh, USA, November, 2011, pp. 420-427.

11. H. Aliakbarpour, J. Quintas, P. Freitas, J. Dias, "Mobile Robot Cooperation with Infrastructure for Surveillance: Towards Cloud Robotics". Workshop on Recognition and Action for Scene Understanding (REACTS) in the 14^{th} International Conference of Computer Analysis of Images and Patterns (CAIP), Spain, September 2011.

12. J. Quintas, K. Khoshhal, H. Aliakbarpour, M. Hofmann, J. Dias, "Using Concurrent Hidden Markov Models to Analyze Human Behaviours in a Smart Home Environment", in Wiamis 2011, 12^{th} international Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), April, 2011.

13. K. Khoshhal, H. Aliakbarpour, J. Quintas, M. Hofmann and J. Dias, "Probabilistic LMA-based Human Motion Analysis by Conjugating Frequency and Spatial based Features", in Wiamis 2011, 12^{th} international Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), April, 2011.

14. K. Khoshhal, H. Aliakbarpour, J. Quintas, K. Mekhnacha, J. Ros and J. Dias, "LMA-based Human Behaviour Analysis Using HMM", in Technological Innovation for Sustainability. DoCEIS 2011. IFIP Advances in Information and Communication Technology, vol 349., February, 2011, pp. 189-196.

15. H. Aliakbarpour, K. Khoshhal, J. Quintas, K. Mekhnacha, J. Ros, M. Andersson and J. Dias, "HMM-based Abnormal Behaviour Detection Using Heterogeneous Sensor Network", in Technological Innovation for Sustainability. DoCEIS 2011. IFIP

Advances in Information and Communication Technology, vol 349., February, 2011, pp. 277-285.

16. K. Khoshhal, H. Aliakbarpour, J. Quintas, P. Drews, J. Dias, "Probabilistic LMA-based Classification of Human Behaviour Understanding Using Power Spectrum Technique", in Information Fusion (FUSION), 2010 13th Conference on., July, 2010, pp. 1–7.

17. P. Drews, J. Quintas, J. Dias, M. Andersson, J. Nygards, J. Rydell, "Crowd behavior analysis under cameras network fusion using probabilistic methods", in Information Fusion (FUSION), 2010 13th Conference on., July, 2010, pp. 1-8.

7.1.2 Collaborations

During the course of the PhD, other works have been carried out that were aligned with the topics of this thesis. In particular, regarding the development of ICT-based solution for improving Health and Quality of Life. The list below summarizes the results from collaboration on research and development projects that aimed at developing digital solutions for the domain of Active and Assisted Living, which included:

- within the scope of CaMeLi[41] AAL project, as previously presented in this thesis, we developed a virtual assistant for supporting elderly living alone at home. The main relation of this collaboration to the thesis was the direct contribution in terms of understanding the users and building a virtual artificial social companion.

- within the scope of CogniWin[42] AAL project, we developed an instrumented computer mouse capable of measuring electro-physiological data that was integrated with an context-aware intelligent virtual assistant for workplace interventions. The main relation of this collaboration to the thesis was the direct contribution in terms of understanding the users and extending our knowledge about the information domain related with artificial social companions.

- within the scope of two large-size national projects TICE.Healthy[43] and AAL4ALL[44], we participated in the development of two digital platforms for building application ecosystems related with Health and Quality of Life. The main relation

[41] www.cameli.eu
[42] www.cogniwin.eu
[43] http://tice.healthy.ipn.pt
[44] http://tice.healthy.ipn.pt/index.php/en/

of this collaboration to the thesis was the direct contribution in terms of understanding the users, extending our knowledge about the information domain related with other possible context of operation for artificial social companions.

Furthermore, all these collaborations contributed directly to understand the diversity of methodologies adopted for project development and building ICT-based solutions for elderly, which impose challenging barriers to achieve effective system integration that will have a negative repercussion in terms of human-machine interaction.

This thesis, gathered all these challenges, difficulties, experiences and solutions into a framework that addresses, in a integrated way, the understanding of users, the understanding of contexts of operation and methods for technical implementation.

Book chapters

1. Christiana Tsiourti, João Quintas, Maher Ben Moussa, Sten Hanke, Niels Alexander Nijdam, Dimitri Konstantas, "The CaMeLi Framework - A Multimodal Virtual Companion for Older Adults", in Intelligent Systems and Applications, Springer International Publishing, January, 2018, DOI:10.1007/978-3-319-69266-1_10.

Peer reviewed International Journals

1. NP. Rocha, A. Queirós, F. Augusto, YL. Rodríguez, C. Cardoso, JM. Grade, J. Quintas, "Information Persistence Services Designed to Support Home Care", JMIR medical informatics, 3(1):e15, March, 2015, DOI:10.2196/medinform.3699.

2. J. Quintas, Y. Charalambous, C. Tsiourti, J. Dias, "Integration of an Automatic Indoor and Outdoor Activity Monitoring with a Social Network". in Journal of Aging and Physical Activity (JAPA), Volume 20, Supplement, August: Abstracts for the 8th World Congress on Aging and Physical Activity: A Celebration of Diversity and Inclusion in Active Ageing 2012, 20, S278, 2012.

Peer reviewed Conferences

1. Carina Dantas, Ana Luísa Jegundo, João Quintas, Ana Isabel Martins, Alexandra Queirós, Nelson Rocha, "European Portuguese Validation of Usefulness, Satisfaction and Ease of Use Questionnaire (USE)", in Recent Advances in Information Systems and Technologies, April, 2017, pp. 561-570.

2. Carina Dantas, Pedro Balhau, Ana Jegundo, Luís Santos, Christophoros Christophorou, Cindy Wings, João Quintas, Eleni Christodoulou, "Verification Methodology of Ethical Compliance for Users, Researchers and Developers of Personal Care Robots", in Recent Advances in Information Systems and Technologies, April, 2017, pp. 754-762.

3. Tsiourti, Christiana and Ben Moussa, Maher and Quintas, João and Loke, Ben and Jochem, Inge and Albuquerque Lopes, Joana and Konstantas, Dimitri, "A Virtual Assistive Companion for Older Adults: Design Implications for a Real-World Application",in SAI Intelligent Systems Conference 2016, September, 2016, pp. 556-566.

4. Portugal, David and Belk, Marios and Quintas, João and Christodoulou, Eleni and Samaras, George, "Identification of an Individual's Frustration in the Work Environment Through a Multi-sensor Computer Mouse",in International Conference on Human Aspects of IT for the Aged Population, July, 2016, pp.79-88.

5. Portugal, David and Dias, Miguel Sales and Christodoulou, Eleni and Belk, Marios and Quintas, João and Samaras, George and Hanke, Sten and Müllner-Rieder, Markus and Glauser, Christoph and Snene, Mehdi, "CogniWin: An Integrated Framework to Support Older Adults at Work", Demonstration at the 24th ACM Conference on User Modeling, Adaptation and Personalization (UMAP 2016), July, 2016.

6. Belk, Marios and Portugal, David and Germanakos, Panagiotis and Quintas, João and Christodoulou, Eleni and Samaras, George,"A Computer Mouse for Stress Identification of Older Adults at Work", in Proceedings of the 1st International Workshop on Human Aspects in Adaptive and Personalized Interactive Environments (HAAPIE 2016), in conjunction with the 24th ACM Conference on User Modeling, Adaptation and Personalization (UMAP 2016), July, 2016.

7. Hanke, Sten and Meinedo, Hugo and Portugal, David and Belk, Marios and Quintas, João and Christodoulou, Eleni and Sili, Miroslav and Dias, Miguel Sales and Samaras, George, "CogniWin–A Virtual Assistance System for Older Adults at Work",in International Conference on Human Aspects of IT for the Aged Population, August, 2015, pp. 257-268.

8. F. Augusto, J. Quintas, C. Cardoso, M. Grade, Y. Llerena, A. Queiros, N. P. Rocha, "Authorization services of the eVida platform", Information Systems and Technologies (CISTI), 2015 10th Iberian Conference, June, 2015, pp. 1-6.

9. Sten Hanke, Hugo Meinedo, David Portugal, Marios Belk, João Quintas, Eleni Christodoulou, Miroslav Sili, Miguel Sales Dias, George Samaras, "CogniWin - A

Virtual Assistance System for Older Adults at Work", Human Aspects of IT for the Aged Population. Design for Everyday Life, Edited by Zhou, Jia and Salvendy, Gavriel, January, 2015, pages 257-268.

10. YL Rodriguez, C Cardoso, M Grade, F Augusto, A Queirós, J Quintas, "Information persistence architecture for informal and formal care providers", New Perspectives in Information Systems and Technologies, 2 (2), 2014, pp. 365-375.
11. Pedro Catré, João Quintas, Alcides Marques, Jorge Dias, "TICE.Healthy Framework for Developing an Ecosystem of Applications in the Domain of Informal Health", GLOBAL HEALTH 2013, The Second International Conference on Global Health Challenges, Lisboa, Portugal, November, 2013, pp. 49-55.
12. S. Pedro, J. Quintas, P. Menezes, "Sensor-Based Detection of Alzheimer'S Disease-Related Behaviors", The International Conference on Health Informatics, Vilamoura, Portugal, November, 2013, pp. 276-279.
13. Y. Llerena Rodriguez, A. Queirós, N. Pacheco Rocha, C. Cardoso, M. Grade, J. Quintas, "We.Can platform: An open management architecture for the information persistence", e-Health Networking, Applications & Services (Healthcom), 2013 IEEE 15th International Conference on, October, 2013, pp. 49-53.
14. Anders Kofod-Petersen, Cindy Wings-Kölgen, Paul Koster, Eleni Chrisodoulou, George Samaras, João Quintas, Ana Leal, Ingvild Ødegård, Roy Beumers, "Co-Living – Successfully increasing socialisation among elderly", AAL Forum 2013 – Impacting Individuals, Society and Economic Growth Conference; Norköpping, Sweden, September, 2013.
15. P. Catré, A. Marques, J. Quintas, J. Dias, "TICE-Healthy: A Dynamic Extensible Personal Health Record". in HEALTHINF2013, Barcelona, Spain, February, 2013, pp. 348-351.

Scientific and Technical Support of Master Thesis and MSc Students

1. Duarte, Ana Margarida Simões, "Hospital2Home - A technological approach for self-measurement of blood pressure during sleep.", 2018, Mestrado Integrado em Engenharia Biomédica, Universidade de Coimbra, supervisors: Quintas, João Manuel Leitão and Cortesão, Rui Pedro Duarte.
2. Ramos, Pedro Gilherme Santos Carvalho Borges, "TopoNiMiA - TOPOlogical NavIgation for Mobile robotIc Agents", 2018, Mestrado Integrado em Engenharia Electrotécnica e de Computadores, Universidade de Coimbra, supervisors: Menezes, Paulo Jorge Carvalho and Quintas, João Manuel Leitão.

3. Reis, Adriana Patrícia Monteiro, "Sistema de monitorização para serviço de urgência em ambiente hospitalar", 2017, Mestrado em Engenharia Biomédica – Área de Especialização em Instrumentação Biomédica, Instituto Superior de Engenharia de Coimbra, supervisors: Coutinho, Fernanda and Ferreira, João and Ferreira, Luís and Quintas, João.

4. Melo, Joana Isabel F. Santos, "Medidor de Glicose Minimamente Invasivo - Prova de Conceito", 2016, Mestrado Integrado em Engenharia Física, Universidade de Coimbra, supervisors: Martins, João Pedro and Domingues, José Paulo. In collaboration with: Instituto Pedro Nunes - Laboratory for Automation and Systems (Quintas, João Manuel Leitão) and Serviço de Medicina Nuclear, SESARAM E.P.E. (Macedo, Rafael).

5. Pedro, Samuel do Nascimento Lago, "Sensor-based detection of Alzheimer's disease-related behaviours", 2013, Mestrado Integrado em Engenharia Biomédica, Universidade de Coimbra, supervisors: Menezes, Paulo Jorge Carvalho and Quintas, João Manuel Leitão.

Co-supervised Bachelor Internships

1. Ferreira, Júlia, "Classificação de padrões na reabilitação da marcha em doentes pós-AVC, a partir de dados de sensores de pressão flexíveis", 2018, Licenciatura em Engenharia Biomédica - Bioeletrónica, Instituto Superior de Engenharia de Coimbra, supervisors: Macedo, Milton and Quintas, João.

2. Jordão, Luís Miguel, "AVI - Assistente Virtual para Idosos", 2017, Licenciatura em Engenharia Informática - Ramo de Desenvolvimento de Aplicações, supervisors: Simões, Anabela and Quintas, João.

3. Sousa, Patrícia, "Validação da utilização de sensores electrofisiológicos e avaliação da sua aplicabilidade em testes cognitivos como parte de protocolo para um sistema de apoio à decisão nos cuidados em pacientes com insuficiência cardíaca - Implementação do procedimento de validação do dispositivo", 2017, Licenciatura em Engenharia Biomédica - Bioeletrónica, Instituto Superior de Engenharia de Coimbra, supervisors: Macedo, Milton and Quintas, João.

4. Santos, Lúcia, "Validação da utilização de sensores electrofisiológicos e avaliação da sua aplicabilidade em testes cognitivos como parte de protocolo para um sistema de apoio à decisão nos cuidados em pacientes com insuficiência cardíaca - Testes cognitivos baseados no dispositivo", 2017, Licenciatura em Engenharia Biomédica - Bioeletrónica, supervisors: Macedo, Milton and Quintas, João.

5. Santos, Susana, "Validação da utilização de sensores electrofisiológicos e avaliação da sua aplicabilidade em testes cognitivos como parte de protocolo para um sistema de apoio à decisão nos cuidados em pacientes com insuficiência cardíaca - Comparação com outras opções, avaliação do benefício potencial em cuidados de IC descrevendo requisitos para avaliação dessas soluções", 2017, Licenciatura em Engenharia Biomédica - Bioeletrónica, supervisors: Macedo, Milton and Quintas, João.

7.2 Future Work

Our future work will continue to mature this framework and will focus on usability validation and implementation of a distributed processing approach for planning algorithms. Moreover, we are interested in **exploring a parallel based implementation for the decision process (i.e. using for example the MapReduce programming model), thus taking more advantage of the paradigm of cloud robotics.**

7.2.1 Parallel implementation

We identify that the most computational intensive part of the POMDP is the calculation of optimal policies for a given model, with the Value Iteration Algorithm (VIA) being the most common choice. Therefore, we are interested in using parallel computing to compute this part, again pushing towards the paradigm of cloud robotics. To achieve this implementation, we are considering to explore the MapReduce programming model [212,213] to implement the VIA part in the decision process adopted in our approach. The basic idea is to define the *Map* and *Reduce* functions that allow to implement matrix-vector multiplication in a parallel method.

7.2.2 Distributed implementation

In fact, we identified that the mathematical formalism of POMDPs present two main issues that can cause a bottleneck in our approach: first, if we have to deal with a large amount of contexts, we might not have an effective approach (i.e. the robot could need information before it is available) and second if the decision model must take into account with a large belief space (i.e. more that two states with more than three actions) this is proven to take a long period of time before converging to an optimal policy. These assumptions lead us in the direction of considering distributed processing to implement our approach, which will take advantage of the paradigm of cloud robotics [28].

Decentralized POMDP (Dec-POMDP) [214] is a distributed way to implement large scale decision systems based on local information for multi-agent approaches. In these approaches, each agent calculates a decision based on individualized information (i.e. context of each agent) thus contributing to the overall decision that optimized the action of the entire group. Nevertheless, the common assumption is that planning is done in an off-line phase, after which the plans are executed in an on-line phase. In spite of the on-line phase is completely decentralized, the off-line planning phase is centralized; a single computer computes the joint plan and subsequently distributes it to the agents.

However, in our approach we do not have collaborative decision making and Dec-POMDP does not address the problem of high dimensionality that can be involved the decision model. Therefore, this approach is not best suited to address our problems and further investigation will be required to understand how this challenge may the addressed in an effective way.

Appendix A

Ontology Development

A.1 Ontology development life cycle

Capturing knowledge and representing by any means is an iterative process. On the particular case of ontology based approaches, as discussed in several research works [215, 216], there is no definitive methodology for developing ontologies and several versions can model the same domain correctly. Hence, experts have agreed on fundamental principles for ontology design:

- concepts in the ontology should be close to objects (physical or logical) and their relationships in the domain of interest.
- there is no single way to model a domain correctly using ontologies.
- ontology development is essentially an iterative process.

Therefore, ontology development life cycle typically follow a pattern with starting point in a rough first-pass at the ontology concepts and terminologies, then revise and refine the evolving ontology, and finally fill in the details. The steps involved in the ontology development life cycle are summarized in Figure A.1.

Some pragmatic heuristics can help us explaining, in more detail, the steps 3 and 4 of Figure A.1, which refer directly to the more practical part of the ontology development life cycle. In step "3. Enumerate important terms in the ontology", an expert should list the main elements of the ontology; this list can be written down "on paper", sketched in a text editor, mind-map, concept map or whatever other tool that allow to define the ontology's basic structure. The list will contain the following:

- Main categories (i.e. classes);

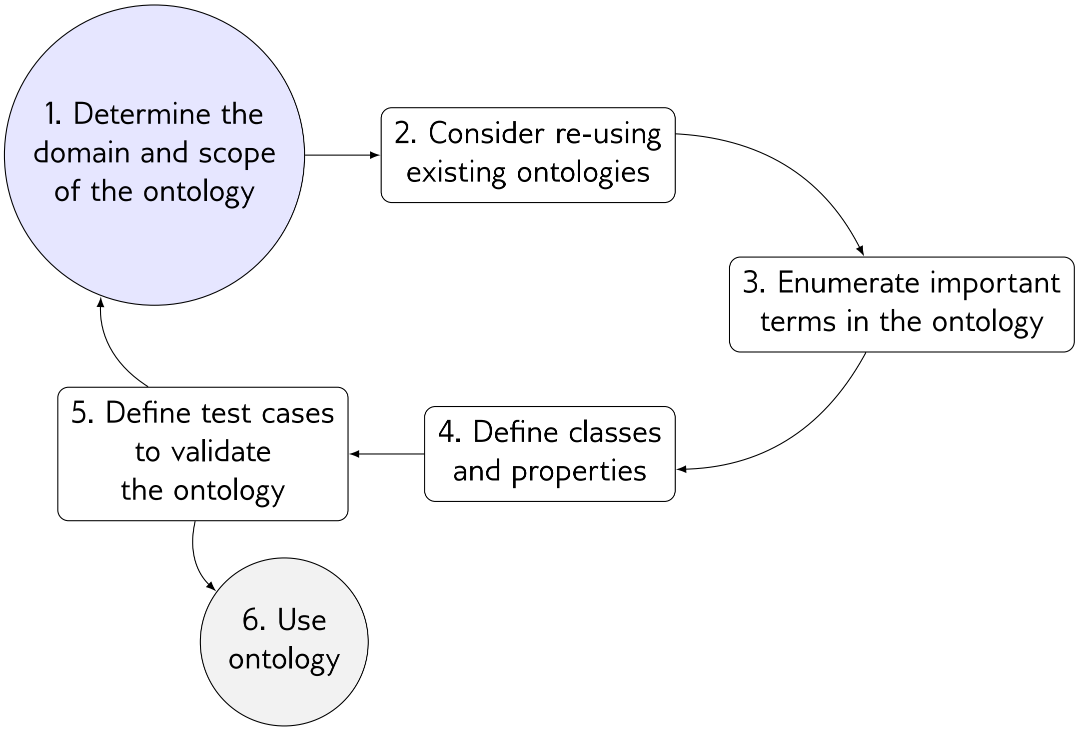


Figure A.1: Ontology development life cycle.

- Main topics (i.e. each type of main categories - sub-classes);
- List of properties and their domains and ranges (i.e. predicates).

In step “4. Define classes and properties”, an expert can start creating/building the ontology based on the list resulting from the step 3. This fourth step can be broken down into five operations, as the following:

1. Create the top hierarchy (i.e. start from the general concept *Thing* and create the hierarchy as: domainEntity -> independentEntity -> Value);
2. Create the skeleton hierarchy (i.e. create the top classes; create each branch of the taxonomy);
3. Create object properties (e.g. relational properties and modifier properties);
4. Add the properties characteristics (e.g. functional, inverse, symmetric, transitive, reflexive, etc.; domains and ranges for each characteristic);
5. Add values for object properties.

A.2 Standard formats for describing an ontology

An ontology can be described using different format types. Popular formats comprise Resource Description Framework (RDF), the Web Ontology Language (OWL) and Semantic Web Rule Language (SWRL).

The RDF [217] is a framework for representing information in the Web. RDF is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax formats.

The OWL language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. OWL has three increasingly-expressive sub-languages: OWL Lite, OWL DL, and OWL Full [218].

The Semantic Web Rule Language (SWRL) is a proposal for a Semantic Web rules-language, combining sublanguages of the OWL (OWL DL and Lite) with those of the Rule Markup Language (Unary/Binary Datalog) [219].

A.3 Building blocks for OWL Ontologies

From the previous formats, we noted OWL has a richer set of operators - e.g. intersection, union and negation. In comparison to other formats, OWL is based on a different logical model that makes it possible for concepts to be defined as well as described. Complex concepts can therefore be built up in definitions out of simpler concepts. The main components of an ontology are described as *Classes*, *Properties* and *Individuals*.

A.3.1 Classes

Classes are interpreted as sets that contain individuals. They are described using formal (mathematical) descriptions that state precisely the requirements for membership of the class. For example, the class Cat would contain all the individuals that are cats in our domain of interest. Classes may be organised into a superclass-subclass hierarchy, which is also known as a taxonomy. Subclasses specialise ('are subsumed by') their superclasses. One of the key features of OWL-DL is that these superclass-subclass relationships (subsumption relationships) can be computed automatically by a reasoner. The word concept is sometimes used in place of class. Classes are a concrete representation of concepts.

Disjoint classes

OWL Classes are assumed to 'overlap'. We therefore cannot assume that an individual is not a member of a particular class simply because it has not been asserted to be a member of that class. In order to 'separate' a group of classes we must make them disjoint from one another. This ensures that an individual which has been asserted to be a member of one of the classes in the group cannot be a member of any other classes in that group.

Necessary and Sufficient Conditions - Primitive and Defined Classes

Necessary conditions can be read as, "If something is a member of this class then it is necessary to fulfil these conditions". With necessary conditions alone, we cannot say that, "If something fulfils these conditions then it must be a member of this class". A class that only has necessary conditions is known as a *Primitive Class*. A class that has at least one set of necessary and sufficient conditions is known as a *Defined Class*. In Defined Classes, any individual that satisfies the definition will belong to the class. In OWL it is possible to have multiple sets of necessary and sufficient conditions (i.e. *Equivalent classes*).

Enumerated Classes

As well as describing classes through named super-classes and anonymous super-classes such as restrictions, OWL allows classes to be defined by precisely listing the individuals that are the members of the class. For example, we might define a class DaysOfTheWeek to contain the individuals (and only the individuals) Sunday, Monday, Tuesday, Wednesday, Thursday, Friday and Saturday. Classes such as this are known as enumerated classes.

Intersection and Union classes

An intersection class is described by combining two or more classes using the AND operator. A union class is created by combining two or more classes using the OR operator.

A.3.2 Properties

Properties are binary relationships on individuals - i.e. properties link two individuals together. They are also known as roles in description logics and relations in Unified Modeling Language (UML) and other object oriented notions. There are three main types of properties, *Object properties*, *Datatype properties* and *Annotation properties*.

Object Properties characteristics

Object properties link an individual to an individual. Although there is no strict naming convention for properties, we recommend that property names start with a lower case letter, have no spaces and have the remaining words capitalised. We also recommend that properties are prefixed with the word 'has', or the word 'is', for example *hasPart*, *isPartOf*, *hasManufacturer*, *isProducerOf*. Not only does this convention help make the intent of the property clearer to humans, it is also taken advantage of by the 'English Prose Tooltip Generator'a, which uses this naming convention where possible to generate more human readable expressions for class descriptions. Each object property may have a corresponding *Inverse property*. If some property links individual *a* to individual *b* then its inverse property will link individual *b* to individual *a*. Futhermore, the complete set of Object Properties characteristics is summarized in table A.1.

Datatype Properties characteristics

Datatype properties describe relationships between individuals and data values. Datatype properties link an individual to an eXtensible Markup Language (XML) Schema Datatype value or an rdf literal. In other words, they describe relationships between an individual and data values. A datatype property can be used to relate an individual to a concrete data value that may be typed or untyped. A datatype property can also be used in a restriction to relate individuals to members of a given datatype. Built in datatypes are specified in the XML schema vocabulary and include integers, oats, strings, booleans etc. In addition to using the predefined set of datatypes we can further specialise the use of a datatype by specifying restrictions on the possible values. For example, it is easy to specify a range of values for a number.

Annotation Properties characteristics

Annotation properties can be used to add information (meta-data - data about data) to classes, individuals and object/ datatype properties. OWL allows classes, prop-

Table A.1: Object Properties characteristics

Characteristic	Description
Functional	Functional properties are also known as single valued properties and also features. If a property is functional, for a given individual, there can be at most one individual that is related to the individual via the property.
Inverse Functional	If a property is inverse functional then it means that the inverse property is functional. For a given individual, there can be at most one individual related to that individual via the property.
Transitive	If a property is transitive, and the property relates individual a to individual b, and also individual b to individual c, then we can infer that individual a is related to individual c via property P. If a property is transitive then its inverse property should also be transitive. Note that if a property is transitive then it cannot be functional. The reason for this is that transitive properties, by their nature, may form 'chains' of individuals. Making a transitive property functional would therefore not make sense.
Symmetric	If a property P is symmetric, and the property relates individual a to individual b then individual b is also related to individual a via property P. Put another way, the property is its own inverse property.
Asymmetric	If a property P is asymmetric, and the property relates individual a to individual b then individual b cannot be related to individual a via property P.
Reflexive	A property P is said to be reflexive when the property must relate individual a to itself.
Irreflexive	If a property P is irreflexive, it can be described as a property that relates an individual a to individual b, where individual a and individual b are not the same.

erties, individuals and the ontology itself (technically speaking the ontology header) to be annotated with various pieces of information/ meta-data. OWL has five pre-defined annotation properties that can be used to annotate classes (including anonymous classes such as restrictions), properties and individuals:

1. *owl:versionInfo* - in general the range of this property is a string.

2. *rdfs:label* - has a range of a string. This property may be used to add meaning-

ful, human readable names to ontology elements such as classes, properties and individuals. rdfs:label can also be used to provide multi-lingual names for ontology elements.

3. *rdfs:comment* - has a range of a string.

4. *rdfs:seeAlso* - has a range of a URI[45] which can be used to identify related resources.

5. *rdfs:isDefinedBy* - has a range of a URI reference which can be used to reference an ontology that defines ontology elements such as classes, properties and individuals.

Properties - Domain/Range

Properties may have a domain and a range specified. Properties link individuals from the domain to individuals from the range. It is important to realise that in OWL domains and ranges should not be viewed as constraints to be checked. They are used as 'axioms' in reasoning. It is possible to specify multiple classes as the range for a property.

A.3.3 Restrictions (Describing vs Defining Classes)

A *Restriction* describes an anonymous class (i.e. an unnamed class) of individuals based on the relationships that members of the class participate in. The anonymous class contains all of the individuals that satisfy the restriction (i.e. all of the individuals that have the relationships required to be a member of the class). In OWL, restrictions can be divided into three main categories: Quantifier Restrictions, Cardinality Restrictions, hasValue Restrictions.

Quantifier restrictions (Existential and Universal Restrictions)

Existential restrictions describe classes of individuals that participate in at least one relationship along a specified property to individuals that are members of a specified class. Existential restrictions are the most common type of restrictions in OWL ontologies. An existential restriction describes a class of individuals that have at least one (some) relationship along a specified property to an individual that is a member of a specified class. Existential restrictions are also known as *Some Restrictions*, or as

[45]Uniform Resource Identifier (URI)

some values from restrictions. *Universal restrictions* describe classes of individuals that for a given property only have relationships along this property to individuals that are members of a specified class. Universal restrictions are given the symbol $\forall$. They constrain the relationships along a given property to individuals that are members of a specific class. Universal restrictions are also know as *AllValuesFrom Restrictions*.

Cardinality restrictions

In OWL we can describe the class of individuals that have at least, at most or exactly a specified number of relationships with other individuals or datatype values. The restrictions that describe these classes are known as Cardinality Restrictions. For a given property P, a Minimum Cardinality Restriction specifies the minimum number of P relationships that an individual must participate in. A Maximum Cardinality Restriction specifies the maximum number of P relationships that an individual can participate in. A Cardinality Restriction specifies the exact number of P relationships that an individual must participate in.

hasValue restrictions

A hasValue restriction, denoted by the symbol $\ni$, describes the set of individuals that have at least one relationship along a specified property to a specific individual.

A.3.4 Individuals

Individuals, also known as instances, can be referred to as being the particular “objects” belonging to a class.

A.3.5 Testing the ontology

A strategy that is often used as a check, so that we can see that we have built our ontology correctly, is to add classes known as *Probe Classes* in order to test the integrity of the ontology. This process typically happens when classifying the ontology using a *reasoner*.

Bibliography

[1] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," in *Robotics and Autonomous Systems,Special issue on Socially Interactive Robots*, vol. 42, no. 3-4, Mar. 2003, pp. 143–166.

[2] K. Severinson-Eklundh, A. Green, and H. Hüttenrauch, "Social and collaborative aspects of interaction with a service robot," in *Robotics and Autonomous Systems,Special issue on Socially Interactive Robots*, vol. 42, no. 3-4, Mar. 2003, pp. 223–234.

[3] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: A survey," in *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, Feb. 2007, pp. 203–275.

[4] J. Quintas, L. Almeida, M. Brito, G. Quintela, P. Menezes, and J. Dias, "Context-based understanding of interaction intentions," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, Sep. 2012, pp. 515–520.

[5] C. Tsiourti, M. B. Moussa, J. a. Quintas, B. Loke, I. Jochem, J. A. Lopes, and D. Konstantas, "A virtual assistive companion for older adults: Design implications for a real-world application," in *SAI Intelligent Systems Conference 2016*, Sep. 2016, pp. 556–566.

[6] D. Portugal, L. Santos, P. Alvito, J. Dias, G. Samaras, and E. Christodoulou, "SocialRobot: An interactive mobile robot for elderly home care," in *In Proc. of the IEEE/SICE International Symposium on System Integration (SII 2015)*, Dec. 2015, pp. 811–816.

[7] G. S. Martins, L. Santos, and J. Dias, "Towards a context-aware adaptable services framework with applications in elderly care," *In Workshop on Improving the Quality of Life in the Elderly using Robotic Assistive Technology, International Conference of Social Robotics*, Oct. 2015.

[8] ——, “The GrowMeUp project and the applicability of action recognition techniques,” *In Proceedings of 3rd Workshop on Recognition and Action for Scene Understanding (REACTS2015)*, Sep. 2015.

[9] S. Kleanthous, C. Christophorou, C. Tsiourti, C. Dantas, R. Wintjens, G. Samaras, and E. Christodoulou, “Analysis of elderly users’ preferences and expectations on service robot’s personality, appearance and interaction,” in *Human Aspects of IT for the Aged Population. Healthy and Active Aging: Second International Conference, ITAP 2016, Held as Part of HCI International 2016, Proceedings*, Jul. 2016, pp. 35–44.

[10] G. H. Bower and N. E. Miller, “Rewarding and punishing effects from stimulating the same place in the rat’s brain,” *Journal of Comparative and Physiological Psychology*, vol. 51, no. 6, pp. 669–674, 1958.

[11] G. H. Bower, “Partial and correlated reward in escape learning.” *Journal of Experimental Psychology*, vol. 59, no. 2, pp. 126–130, 1960.

[12] G. H. Bower and N. E. Miller, “Effects of amount of reward on strength of approach in an approach-avoidance conflict.” *Journal of Comparative and Physiological Psychology*, vol. 53, no. 1, pp. 59–62, 1960.

[13] G. H. Bower, “Correlated delay of reinforcement.” *Journal of Comparative and Physiological Psychology*, vol. 54, no. 2, pp. 196–203, 1961.

[14] ——, “A contrast effect in differential conditioning.” *Journal of Experimental Psychology*, vol. 62, no. 2, pp. 196–199, 1961.

[15] G. H. Bower and K. Holyoak, “Encoding and recognition memory for naturalistic sounds.” *Journal of Experimental Psychology*, vol. 101, no. 2, pp. 360–366, 1973.

[16] G. H. Bower, “Selective facilitation and interference in retention of prose.” *Journal of Educational Psychology*, vol. 66, no. 1, pp. 1–8, 1974.

[17] G. H. Bower, S. G. Gilligan, and K. P. Monteiro, “Selectivity of learning caused by affective states.” *Journal of Experimental Psychology: General*, vol. 110, no. 4, pp. 451–473, 1981.

[18] G. H. Bower, “Mood and memory.” *American Psychologist*, vol. 36, no. 2, pp. 129–148, 1981.

[19] G. H. Bower, S. Thompson-Schill, and E. Tulving, “Reducing retroactive interference: An interference analysis.” *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 20, no. 1, pp. 51–66, 1994.

[20] G. H. Bower, A. D. Wagner, S. E. Newman, J. D. Randle, and M. J. Hodges, "Does recoding interfering material improve recall?" *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 22, no. 1, pp. 240–245, 1996.

[21] G. H. Bower and M. Rinck, "Selecting one among many referents in spatial situation models." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 27, no. 1, pp. 81–98, 2001.

[22] L. W. Barsalou, "Context-independent and context-dependent information in concepts," *Memory & Cognition*, vol. 10, no. 1, pp. 82–93, Jan. 1982.

[23] B. A. Mellers and A. D. J. Cooke, "The role of task and context in preference measurements," *Psychological Science*, vol. 7, no. 2, pp. 76–82, 1996.

[24] L. Boroditsky, "Does Language Shape Thought?: Mandarin and English Speakers' Conceptions of Time," *Cognitive Psychology*, vol. 43, no. 1, pp. 1–22, Aug. 2001.

[25] P. H. Thibodeau and L. Boroditsky, "Metaphors We Think With: The Role of Metaphor in Reasoning," *PLoS ONE*, vol. 6, no. 2, p. e16782, Feb. 2011.

[26] ——, "Natural Language Metaphors Covertly Influence Reasoning," *PLoS ONE*, vol. 8, no. 1, p. e52961, Jan. 2013.

[27] ——, "Measuring effects of metaphor in a dynamic opinion landscape," *PLOS ONE*, vol. 10, no. 7, pp. 1–22, Jul. 2015.

[28] J. Quintas, K. Khoshhal, H. Aliakbarpour, M. Hofmann, and J. Dias, "Using concurrent hidden markov models to analyze human behaviours in a smart home environment," in *Proceeding of International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2011)*, Apr. 2011, pp. uuid:1ba89b49-6eb5-4b6f-b81b-f7a787e45 114.

[29] H. Aliakbarpour, K. Khoshhal, J. Quintas, K. Mekhnacha, J. Ros, M. Andersson, and J. Dias, "Hmm-based abnormal behaviour detection using heterogeneous sensor network," in *Technological Innovation for Sustainability. DoCEIS 2011. IFIP Advances in Information and Communication Technology*, vol. 349, 2011, pp. 277–285.

[30] J. F. Ferreira, J. Lobo, P. Bessiére, M. Castelo-Branco, and J. Dias, "A bayesian framework for active artificial perception," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 43, no. 2, pp. 699–711, Apr. 2013.

[31] J. A. Prado, C. Simplício, N. F. Lori, and J. Dias, "Visuo-auditory multimodal emotional structure to improve human-robot-interaction," *International Journal of Social Robotics*, vol. 4, no. 1, pp. 29–51, Jan. 2012.

[32] M. Sili, M. Gira, M. Müllner-Rieder, and C. Mayer, *Interaction Model-Based User Interfaces: Two Approaches, One Goal - Comparison of Two User Interface Generation Approaches Applying Interaction Models*. Springer International Publishing, May 2015, pp. 185–197.

[33] K. Altisen, A. Clodic, F. Maraninchi, and E. Rutten, "Using controller-synthesis techniques to build property-enforcing layers," in *Programming Languages and Systems: 12th European Symposium on Programming, ESOP 2003 Held as Part of the Joint European Conferences on Theory and Practice of Software*, Apr 2003, pp. 174–188.

[34] R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and R. Chatila, "Task planning for human-robot interaction," in *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies*, Oct. 2005, pp. 81–85.

[35] ——, "Toward human-aware robot task planning," in *AAAI Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before*, Jul. 2006, pp. 39–46.

[36] A. Clodic, H. Cao, S. Alili, V. Montreuil, R. Alami, and R. Chatila, *SHARY: A Supervision System Adapted to Human-Robot Interaction*. Springer-Verlag Berlin Heidelberg, Jul. 2009, vol. 54, pp. 229–238.

[37] S. Devin, A. Clodic, and R. Alami, "About decisions during human-robot shared plan achievement: Who should act and how?" in *International Conference on Social Robotics*. Springer, 2017, pp. 453–463.

[38] S. Lemaignan, M. Warnier, E. A. Sisbot, A. Clodic, and R. Alami, "Artificial cognition for social human–robot interaction: An implementation," *Artificial Intelligence*, vol. 247, pp. 45–69, 2017.

[39] A. K. Pandey, L. de Silva, and R. Alami, "A human-robot competition: Towards evaluating robots' reasoning abilities for hri," in *International Conference on Social Robotics*. Springer, 2016, pp. 138–147.

[40] A. Clodic, E. Pacherie, R. Alami, and R. Chatila, "Key elements for human-robot joint action," in *Sociality and Normativity for Robots*. Springer, 2017, pp. 159–177.

[41] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Workshop on Mobile Computing Systems and Applications*, Dec. 1994, pp. 85–90.

[42] J. Pascoe, "Adding generic contextual capabilities to wearable computers," in *Proceedings of the 2nd IEEE International Symposium on Wearable Computers (ISWC98)*, Oct. 1998, p. 92.

[43] A. K. Dey, "Understanding and using context," *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7, Feb. 2001.

[44] H. G. Kim, J. Y. Yang, and D. S. Kwon, "Experience based domestic environment and user adaptive cleaning algorithm of a robot cleaner," in *11th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI 2014*, Nov. 2015, pp. 176–178.

[45] C.-m. Huang and B. Mutlu, "Anticipatory Robot Control for Efficient Human-Robot Collaboration," *Human-Robot Interaction*, no. Section V, 2016.

[46] I. Duque, K. Dautenhahn, K. L. Koay, L. Willcock, and B. Christianson, "A different approach of using Personas in human-robot interaction: Integrating Personas as computational models to modify robot companions' behaviour," in *22nd IEEE International Symposium on Robot and Human Interactive Communication*, Aug. 2013, pp. 424–429.

[47] A. Aly and A. Tapus, "A model for synthesizing a combined verbal and nonverbal behavior based on personality traits in human-robot interaction," *ACM/IEEE International Conference on Human-Robot Interaction*, pp. 325–332, Mar. 2013.

[48] S. D. Klee, B. Q. Ferreira, R. Silva, P. Costeira, F. S. Melo, and M. Veloso, "Personalized Assistance for Dressing Users," in *7th International Conference on Social Robotics (ICSR)*. Springer International Publishing, Oct. 2015.

[49] A. Madureira, B. Cunha, J. P. Pereira, S. Gomes, I. Pereira, J. M. Santos, and A. Abraham, "Using Personas for Supporting User Modeling on Scheduling Systems," in *14th International Conference on Hybrid Intelligent Systems (HIS)*. IEEE, Dec. 2014, pp. 279–284.

[50] G. Gordon and C. Breazeal, "Bayesian Active Learning-based Robot Tutor for Children ' s Word-Reading Skills," Jan. 2015, pp. 1343–1349.

[51] A. B. Karami, K. Sehaba, and B. Encelle, "Adaptive artificial companions learning from users feedback," *Adaptive Behavior*, vol. 24, no. 2, pp. 69–86, 2016.

[52] G. H. Lim, S. W. Hong, I. Lee, I. H. Suh, and M. Beetz, "Robot recommender system using affection-based episode ontology for personalization," *22nd IEEE International Symposium on Robot and Human Interactive Communication*, pp. 155–160, Aug. 2013.

[53] J. A. Konstan and J. Riedl, "Recommender systems: From algorithms to user experience," *User Modeling and User-Adapted Interaction*, vol. 22, no. 1-2, pp. 101–123, 2012.

[54] Y. Kim and W. C. Yoon, "Generating task-oriented interactions of service robots," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 8, pp. 981–994, Aug 2014.

[55] H. Zhang, W. Ni, X. Li, and Y. Yang, "Modeling the heterogeneous duration of user interest in time-dependent recommendation: A hidden semi-markov approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 2, pp. 177–194, Feb 2018.

[56] B. Cheng, S. Zhong, and J. Chen, "Context ontology-based reasoning service for multimedia conferencing process intelligence," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 12, pp. 3219–3232, Dec 2017.

[57] J.-y. Hong, E.-h. Suh, and S.-J. Kim, "Context-aware systems: A literature review and classification," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8509–8522, May 2009.

[58] R. Murphy and J. Burke, "Up from the rubble: Lessons learned about hri from search and rescue," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 49, no. 3, 2005, pp. 437–441.

[59] O. Brdiczka, M. Langet, J. Maisonnasse, and J. L. Crowley, "Detecting human behaviour models from multimodal observation in a smart home," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 4, pp. 588–597, 2009.

[60] O. Brdiczka, J. L. Crowley, and P. Reignier, "Learning situation models in a smart home," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 1, pp. 56–63, Feb. 2009.

[61] H. Sharp, Y. Rogers, and J. Preece, *Interaction Design: Beyond Human-Computer Interaction*, 2nd ed. Wiley, Mar. 2007.

[62] R. Molich and J. Nielsen, "Improving a human-computer dialogue," *Communications of the ACM*, vol. 33, pp. 338–348, Mar. 1990.

[63] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *Proceedings of ACM CHI'90 Conference*, Apr. 1990, pp. 249–256.

[64] J. Nielsen, "Enhancing the explanatory power of usability heuristics," in *Proceedings of ACM CHI'94 Conference*, Apr. 1994, pp. 152–158.

[65] D. Liddle, "Design of the conceptual model," *Bringing Design to Software*, pp. 17–31, 1996.

[66] A. Dix, J. Finlay, G. Abowd, and R. Beale, *Human-Computer Interaction*, 3rd ed. Pearson, 2004.

[67] R. Atkinson and R. Shiffrin, "Chapter: Human memory: A proposed system and its control processes," *The psychology of learning and motivation*, vol. 2, pp. 89–195, 1968.

[68] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychological Review*, vol. 63, no. 2, pp. 81–97, 1956.

[69] P. N. Johnson-Laird, "Mental models and human reasoning," *Proceedings of the National Academy of Sciences*, vol. 107, no. 43, pp. 18 243–18 250, 2010.

[70] J. M. Carroll, *The Nurnberg Funnel*. Cambridge, MA: MIT Press., 1990.

[71] W. James, "What is an emotion?" *Mind*, vol. 9, pp. 188–205, 1884.

[72] W. B. Cannon, "The James-Lange Theory of emotions: A critical examination and an alternative theory," *American Journal of Psychology*, vol. 39, pp. 106–124, 1927.

[73] S. Schacher and J. Singer, "Cognitive, social and psychological determinants of emotional state," *Physiological Review*, vol. 69, pp. 379–399, 1962.

[74] B. Troadec and C. Martinot, *Le développement cognitif: Théories actuelles de la pensée en contextes*. Belin, 2003.

[75] E. C. Ferstl, "Context effects in syntactic ambiguity resolution: the location of presuppositional phrase attachment," in *16th Annual Conference of the Cognitive Science Society*, Aug. 1994, pp. 295–300.

[76] T. Holtgraves, "Communication in context: effects of speaker status on the comprehension of indirect requests," *Journal of Experimental Psychology*, vol. 20, no. 5, pp. 1205–1218, Sep. 1994.

[77] R. M. Turner, "Context-mediated behavior for intelligent agents," *International Journal of Human-Computer Studies*, vol. 48, no. 3, pp. 307–330, Mar. 1998.

[78] A. Boytsov, A. Zaslavsky, E. Eryilmaz, and S. Albayrak, "Situation awareness meets ontologies: A context spaces case study," in *Modeling and Using Context: 9th International and Interdisciplinary Conference, CONTEXT 2015*, Nov 2015, pp. 3–17.

[79] M. Bazire and P. Brézillon, “Understanding Context Before Using It,” in *Modeling and Using Context: 5th International and Interdisciplinary Conference, CONTEXT 2005*, Jul 2005, pp. 29–40.

[80] H.-w. G. Albrecht Schmidt, Michael Beigl, “There is more to context than location,” *Computers and Graphics*, vol. 23, no. 6, pp. 893–091, Dec. 1998.

[81] G. D. Abowd, A. K. Dey, P. Brown, N. Davies, M. Smith, and P. Steggles, “Towards a better understanding of context and context-awareness,” in *HUC99 Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, Sep. 1999, pp. 304–307.

[82] J. Bisgaard, M. Heise, and C. Steffensen, “How is context and context-awareness defined and applied? a survey of context-awareness,” *Aalborg University*, pp. 31–40, Jan. 2004.

[83] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, “A survey of context modelling and reasoning techniques,” *Pervasive and Mobile Computing*, vol. 6, no. 2, pp. 161–180, Apr 2010.

[84] M. Ivory and M. Hears, “The state of the art in automating usability evaluation of user interfaces,” in *J. ACM Comput. Surv*, vol. 33, no. 4, 2001, pp. 470–516.

[85] A. I. Martins, A. Queirós, N. P. Rocha, and B. S. Santos, “Avaliação de usabilidade: Uma revisão sistemática da literatura,” in *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, vol. 11, no. 1, 2013, pp. 31–44.

[86] B. Hanington and B. Martin, *Universal Methods of Design: 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions.* Beverly, MA: Rockport Publishers, Feb. 2012.

[87] J. Brooke, “Sus: A quick and dirty usability scale,” *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, Nov. 1995.

[88] J. R. Lewis, “Psychometric evaluation of the post-study system usability questionaire: The pssuq,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 36, no. 16, Oct. 1992, p. 1259–1260.

[89] A. M. Lund, “Measuring usability with the use questionnaire,” in *Usability Interface*, vol. 8, no. 2, Jan. 2001, p. 3–6.

[90] C. Huijnen, A. Badii, H. van den Heuvel, P. Caleb-Solly, and D. Thiemert, “Maybe it becomes a buddy, but do not call it a robot - seamless cooperation between companion robots and smart homes,” in *Ambient Intelligence, AmI 2011. Lecture Notes in Computer Science*, vol. 7040. Springer, Berlin, Heidelberg, Nov. 2011.

[91] E. Guizzo and E. Ackerman, “The rise of the robot worker,” *IEEE Spectrum*, vol. 49, no. 10, pp. 34–41, Oct. 2012.

[92] J. Forlizzi and C. DiSalvo, “Service robots in the domestic environment: A study of the roomba vacuum in the home,” in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, Mar. 2006, pp. 258–265.

[93] T. W. Bickmore, D. Schulman, and C. L. Sidner, “A reusable framework for health counseling dialogue systems based on a behavioral medicine ontology,” *Journal of Biomedical Informatics*, vol. 44, no. 2, pp. 183–197, 2011.

[94] A. G. Bickmore and R. W. Picard, “Establishing the computer-patient working alliance in automated health behavior change interventions,” *Patient Education Counseling*, vol. 59, no. 1, pp. 21–30, October 2005.

[95] C. Lisetti, U. Yasavur, C. de Leon, R. Amini, U. Visser, and N. Rishe, “Building an on-demand avatar-based health intervention for behavior change,” in *Proc. Twenty-Fifth International Florida Artificial Intelligence Research Society Conference*, May 2012, p. 175–180.

[96] F. de Rosis, N. Novielli, V. Carofiglio, A. Cavalluzzi, and B. D. Carolis, “User modeling and adaptation in health promotion dialogs with an animated character,” *Journal Biomedical Informatics*, vol. 39, no. 5, pp. 514–531, 2006.

[97] D. Schulman, “Embodied agents for long-term interaction,” Ph.D. dissertation, Northeastern University, 2013.

[98] T. W. Bickmore, “Relational agents: Effecting change through human-computer relationships,” Ph.D. dissertation, Massachusetts Institute of Technology, 2003.

[99] J. Cassell, “More than just another pretty face: Embodied conversational interface agents,” *Communications of the ACM*, vol. 43, pp. 70–78, 2000.

[100] M. Kramer, R. Yaghoubzadeh, S. Kopp, and K. Pitsch, “A conversational virtual human as autonomous assistant for elderly and cognitively impaired users? social acceptability and design considerations,” in *INFORMATIK 2013. Informatik angepasst an Mensch, Organisation und Umwelt, Lecture Notes in Informatics*. Springer, 2013.

[101] R. Yaghoubzadeh, M. Kramer, K. Pitsch, and S. Kopp, “Virtual agents as daily assistants for elderly or cognitively impaired people,” in *Intelligent Virtual Agents. IVA 2013. Lecture Notes in Computer Science*, vol. 8108, Aug. 2013, pp. 79–91.

[102] J. Klein, Y. Moon, and R. W. Picard, "This computer responds to user frustration," in *Proceedings of CHI '99 Extended Abstracts on Human Factors in Computing Systems*, May 1999, pp. 242–243.

[103] ——, "This computer responds to user frustration: Theory, design, and results," *Interacting with computers*, vol. 14, no. 2, pp. 119–140, 2002.

[104] A. Nijholt, "Disappearing computers, social actors and embodied agents," in *Cyberworlds, 2003. Proceedings. 2003 International Conference on*, Dec. 2003, pp. 128–134.

[105] J. F. Kelley, "An iterative design methodology for user-friendly natural language office information applications," *ACM Transactions on Information Systems*, vol. 2, no. 1, pp. 26–41, 1984.

[106] L. P. Vardoulakis, L. Ring, B. Barry, C. L. Sidner, and T. Bickmore, "Designing relational agents as long term social companions for older adults." in *Intelligent Virtual Agents. IVA 2012. Lecture Notes in Computer Science*, vol. 7502. Springer, Berlin, Heidelberg, Sep. 2012, pp. 289–302.

[107] Y. Sakai, Y. Nonaka, K. Yasuda, and Y. I. Nakano, "Listener agent for elderly people with dementia," in *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, Mar. 2012, pp. 199–200.

[108] L. Ring, B. Barry, K. Totzke, and T. Bickmore, "Addressing loneliness and isolation in older adults: Proactive affective agents provide better support." in *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, Sep. 2013, pp. 61–66.

[109] T. W. Bickmore, R. A. Silliman, K. Nelson, D. M. Cheng, M. Winter, L. Henault, and M. K. Paasche-Orlow, "A randomized controlled trial of an automated exercise coach for older adults," *Journal of the American Geriatrics Society*, vol. 61, no. 10, pp. 1676–83, October 2013.

[110] R. H. Campbell, M. N. Grimshaw, and G. M. Green, "Relational agents: A critical review," *The Open Virtual Reality Journal*, vol. 1, no. 1, pp. 1–7, April 2009.

[111] T. W. Bickmore, L. Caruso, K. Clough-Gorr, and T. Heeren, "'It's just like you talk to a friend' relational agents for older adults," *Interacting with Computers*, vol. 17, no. 6, pp. 711–735, December 2005.

[112] Z. Kasap, M. B. Moussa, P. Chaudhuri, and N. Magnenat-Thalmann, "Making them remember: Emotional virtual characters with memory," *IEEE Computer Graphics Applications*, vol. 29, no. 2, pp. 20–29, March 2009.

[113] Z. Kasap and N. Magnenat-Thalmann, “Building long-term relationships with virtual and robotic characters: the role of remembering,” *The Visual Computer*, vol. 28, no. 1, pp. 87–97, January 2012.

[114] L. Ring, L. Shi, K. Totzke, and T. Bickmore, “Social support agents for older adults: longitudinal affective computing in the home,” *Journal on Multimodal User Interfaces*, vol. 9, no. 1, pp. 79–88, 2015.

[115] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand, “An architecture for autonomy,” *The International Journal of Robotics Research*, vol. 17, no. 4, pp. 315–337, Apr. 1998.

[116] R. Alami, A. Clodic, R. Chatila, and S. Lemaignan, “Reasoning about humans and its use in a cognitive control architecture for a collaborative robot,” in *Cognitive Architectures for Human-Robot Interaction Workshop@ HRI'14*, Mar. 2014.

[117] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore *et al.*, “SPENCER: A socially aware service robot for passenger guidance and help in busy airports,” in *Field and Service Robotics. Springer Tracts in Advanced Robotics*, vol. 113, Mar. 2016, pp. 607–622.

[118] S. Devin, G. Milliez, M. Fiore, A. Clodic, and R. Alami, “Some essential skills and their combination in an architecture for a cognitive and interactive robot,” *2nd Workshop on Cognitive Architectures for Social Human-Robot Interaction 2016*, Mar. 2016.

[119] M. Morandell, A. Hochgatterer, B. Wöckl, S. Dittenberger, and S. Fagel, “Avatars@home interFACEing the Smart Home for Elderly People,” in *HCI and Usability for e-Inclusion. USAB 2009. Lecture Notes in Computer Science*, vol. 5889. Springer, Berlin, Heidelberg, Nov. 2009, pp. 353–365.

[120] N. Galdona, J. Yanguas, I. Etxeberria, J. Millan-Calenti, C. Huijnen, S. Dittenberger, A. Rigaud, and A. Hochgatterer, “Character user interface acceptation and pleasantness in persons with mild cognitive impairment, informal and formal caregivers.” in *GERONTOLOGIST, 62nd Annual Scientific Meeting Gerontological Society of America*, vol. 49, no. S2, Jan. 2009, p. 281.

[121] E. Carrasco, G. Epelde, A. Moreno, A. Ortiz, I. Garcia, A. Arruti, C. Buiza, E. Urdaneta, A. Etxaniz, and M. González, “Natural interaction between virtual characters and persons with alzheimer's disease,” in *Computers Helping People with Special Needs. ICCHP 2008. Lecture Notes in Computer Science*, vol. 5105. Springer, Berlin, Heidelberg, Jul. 2008, pp. 38–45.

[122] M. Porta, "Vision-based user interfaces: methods and applications," *International Journal of Human-Computer Studies*, vol. 57, no. 1, pp. 27 – 73, 2002.

[123] R. Kelley, M. Nicolescu, A. Tavakkoli, M. Nicolescu, C. King, and G. Bebis, "Understanding Human Intentions via Hidden Markov Models in Autonomous Mobile Robots," in *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, Mar. 2008, pp. 367–374.

[124] S. Waldherr, R. Romero, and S. Thrun, "A gesture based interface for human-robot interaction," *Autonomous Robots*, vol. 9, no. 2, pp. 151–173, Sep. 2000.

[125] C. Clavel, I. Vasilescu, L. Devillers, G. Richard, and T. Ehrette, "Fear-type emotion recognition for future audio-based surveillance systems," *Speech Communication*, vol. 50, no. 6, pp. 487–503, Jun. 2008.

[126] K. Yoshii, M. Goto, and H. G. Okuno, "Drum sound recognition for polyphonic audio signals by adaptation and matching of spectogram templates with harmonic structure suppression," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 333–345, Jan. 2007.

[127] N. Sawhney and C. Schmandt, "Nomadic radio: Speech and audio interaction for contextual messaging in nomadic environments," *ACM Trans. Comput.-Hum. Interact.*, vol. 7, no. 3, pp. 353–383, Sep. 2000.

[128] L. Stifelman, B. Arons, and C. Schmandt, "The audio notebook: Paper and pen interaction with structured speech," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, Apr. 2001, pp. 182–189.

[129] K. Salisbury, F. Conti, and F. Barbagli, "Haptic rendering: introductory concepts," *IEEE Computer Graphics and Applications*, vol. 24, no. 2, pp. 24–32, Mar. 2004.

[130] M. Bordegoni, U. Cugini, P. Belluco, and M. Aliverti, *Evaluation of a Haptic-Based Interaction System for Virtual Manual Assembly*. Berlin, Heidelberg: Springer Berlin Heidelberg, Jul. 2009, pp. 303–312.

[131] A. F. Abate, M. Guida, P. Leoncini, M. Nappi, and S. Ricciardi, "A haptic-based approach to virtual training for aerospace industry," *Journal of Visual Languages & Computing*, vol. 20, no. 5, pp. 318 – 325, Oct. 2009.

[132] T. Salter, F. Michaud, D. Létourneau, D. C. Lee, and I. P. Werry, "Using proprioceptive sensors for categorizing human-robot interactions," in *Human-Robot Interaction (HRI), 2007 2nd ACM/IEEE International Conference on*, Mar. 2007, pp. 105–112.

[133] R. F. Boian, J. E. Deutsch, C. S. Lee, G. C. Burdea, and J. Lewis, “Haptic effects for virtual reality-based post-stroke rehabilitation,” in *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings.*, Mar. 2003, pp. 247–253.

[134] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, and T. M. Vaughan, “Brain-computer interface technology: a review of the first international meeting,” *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 2, pp. 164–173, Jun. 2000.

[135] U. Chaudhary, B. Xia, S. Silvoni, L. G. Cohen, and N. Birbaumer, “Brain–computer interface–based communication in the completely locked-in state,” *PLOS Biology*, vol. 15, no. 1, pp. 1–25, Jan. 2017.

[136] H. Gurkok, A. Nijholt, and M. Poel, *Brain-Computer Interface Games: Towards a Framework*. Singapore: Springer Singapore, Aug. 2016, pp. 133–150.

[137] M. Kryger, B. Wester, E. A. Pohlmeyer, M. Rich, B. John, J. Beaty, M. McLoughlin, M. Boninger, and E. C. Tyler-Kabara, “Flight simulation using a brain-computer interface: A pilot, pilot study,” *Experimental Neurology*, vol. 287, Part 4, pp. 473 – 478, Jan. 2017.

[138] C. Neti, G. Iyengar, G. Potamianos, A. W. Senior, and B. Maison, “Perceptual interfaces for information interaction: joint processing of audio and visual information for human-computer interaction.” in *INTERSPEECH*, Oct. 2000, pp. 11–14.

[139] S. Lang, M. Kleinehagenbrock, S. Hohenner, J. Fritsch, G. A. Fink, and G. Sagerer, “Providing the basis for human-robot-interaction: A multi-modal attention system for a mobile robot,” in *Proceedings of the 5th international conference on Multimodal interfaces*, Nov. 2003, pp. 28–35.

[140] T. Stivers and J. Sidnell, “Introduction: multimodal interaction,” *Semiotica*, vol. 2005, no. 156, pp. 1–20, Dec. 2005.

[141] M. Serrano, L. Nigay, J.-Y. L. Lawson, A. Ramsay, R. Murray-Smith, and S. Denef, “The openinterface framework: A tool for multimodal interaction.” in *CHI ’08 Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA: ACM, Apr. 2008, pp. 3501–3506.

[142] E. S. John, S. J. Rigo, and J. Barbosa, “Assistive robotics: Adaptive multimodal interaction improving people with communication disorders,” *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 175–180, Dec. 2016.

[143] N. Almeida, S. Silva, B. S. Santos, and A. Teixeira, “Interactive, multi-device visualization supported by a multimodal interaction framework: Proof of concept,” in *International Conference on Human Aspects of IT for the Aged Population*. Springer, Jun. 2016, pp. 279–289.

[144] C. Munteanu, P. Irani, S. Oviatt, M. Aylett, G. Penn, S. Pan, N. Sharma, F. Rudzicz, R. Gomez, B. Cowan, and K. Nakamura, “Designing speech, acoustic and multimodal interactions,” in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA: ACM, May 2017, pp. 601–608.

[145] M. Terek, “Decision trees and Influence diagrams in Decision analysis,” *Journal of the Applied Mathematics, Statistics and Informatics (JAMSI)*, vol. 1, no. 1, pp. 121–135, Apr. 2005.

[146] M. Diehl and Y. Y. Haimes, “Influence diagrams with multiple objectives and tradeoff analysis,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 34, no. 3, pp. 293–304, May 2004.

[147] H. Fazlollahtabar and M. Saidi-Mehrabad, “Optimizing multi-objective decision making having qualitative evaluation,” *Journal of Industrial and Management Optimization*, vol. 11, no. 3, pp. 747–762, Jul. 2015.

[148] G. Su, T. Chen, Y. Feng, D. S. Rosenblum, and P. S. Thiagarajan, “An iterative decision-making scheme for markov decision processes and its application to self-adaptive systems,” in *Fundamental Approaches to Software Engineering: 19th International Conference, FASE 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016*, Apr. 2016, pp. 269–286.

[149] P. Norvig and S. Russell, *Artificial Intelligence: A Modern Approach*, ser. Prentice Hall Series In Artificial Intelligence. Prentice Hall, 2003.

[150] T. R. Gruber, “Toward principles for the design of ontologies used for knowledge sharing,” *International Journal of Human-Computer Studies*, vol. 43, no. 4-5, pp. 907–928, 1995.

[151] A. Maedche and S. Staab, “Semi-automatic engineering of ontologies from text,” in *Proceedings of the 12th Internal Conference on Software and Knowledge Engineering*, Jul. 2000, pp. 231–239.

[152] D. Calvanese, G. De Giacomo, and M. Lenzerini, “Ontology of integration and integration of ontologies,” *Proceedings of the 2001 International Description Logics Workshop (DL2001)*, p. 30, Aug. 2001.

[153] A. Kiryakov, K. I. Simov, and M. Dimitrov, "Ontomap: Portal for upper-level ontologies," in *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*. ACM, Oct. 2001, pp. 47–58.

[154] J. Fernandez-Breis and R. Martinez-Bajar, "A cooperative framework for integrating ontologies," *International Journal of Human-Computer Studies*, vol. 56, no. 6, pp. 665–720, 2002.

[155] J. Madhavan, P. Bernstein, P. Domingos, and A. Halevy, "Representing and reasoning about mappings between domain models," in *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI'02)*, Jul. 2002, pp. 80–86.

[156] C. Schlenoff and E. Messina, "A robot ontology for urban search and rescue," in *Proceedings of the 2005 ACM Workshop on Research in Knowledge Representation for Autonomous Systems*. ACM, Nov. 2005, pp. 27–34.

[157] Y. Weihong, "Research on maritime search and rescue decision-making ontology model," in *Environmental Science and Information Application Technology, 2009. ESIAT 2009. International Conference on*, vol. 2, Jul. 2009, pp. 140–142.

[158] M. Tenorth, L. Kunze, D. Jain, , and M. Beetz, "Knowrob-map - knowledge linked semantic object maps," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, Dec. 2010, pp. 430–435.

[159] M. Tenorth, "Knowledge processing for autonomous robots," Ph.D. dissertation, Technische Universität München, Nov. 2011.

[160] S. Keshavdas and G.-j. M. Kruijff, "Functional mapping for human-robot collaborative exploration," *International Journal of Computers and Applications*, vol. 35, no. 3, pp. 125–135, Jan. 2013.

[161] E. Prestes, J. L. Carbonera, S. R. Fiorini, V. A. Jorge, M. Abel, R. Madhavan, A. Locoro, P. Goncalves, M. E. Barreto, M. Habib *et al.*, "Towards a core ontology for robotics and automation," *Robotics and Autonomous Systems*, vol. 61, no. 11, pp. 1193–1204, 2013.

[162] T. Strang and C. Linnhoff-Popien, "A Context Modeling Survey," in *First International Workshop on Advanced Context Modelling, Reasoning And Management at UbiComp 2004*, no. 4, Sep. 2004, pp. 1–8.

[163] J. Quintas, P. Menezes, and J. Dias, "Cloud robotics: Towards context aware robotic networks," in *Proceedings of IASTED, The 16th IASTED International Conference on Robotics (Robo 2011)*, Nov. 2011, p. 420–427.

[164] ——, "Auto-adaptive interactive systems for active and assisted living applications," in *Technological Innovation for Cyber-Physical Systems. DoCEIS 2016. IFIP Advances in Information and Communication Technology*, Apr. 2016, pp. 161–168.

[165] ——, "Context-based decision system for human-machine interaction applications," in *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, Oct. 2016, pp. 3906–3911.

[166] ——, "Information model and architecture specification for context awareness interaction decision support in cyber-physical human-machine systems," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 323–331, Dec. 2016.

[167] C. Tsiourti, E. Joly, C. Wings, M. B. Moussa, and K. Wac, "Virtual assistive companions for older adults: Qualitative field study and design implications," in *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*, May 2014, pp. 57–64.

[168] J. Cassell, T. Bickmore, L. Campbell, H. Vilhjálmsson, and H. Yan, "Human conversation as a system framework," *Designing embodied conversational agents*, pp. 29–63, Dec. 2001.

[169] K. Khoshhal, H. Aliakbarpour, J. Quintas, P. Drews, and J. Dias, "Probabilistic Ima-based classification of human behaviour understanding using power spectrum technique," in *Information Fusion (FUSION), 2010 13th Conference on*. IEEE, Jul. 2010, pp. 1–7.

[170] H. V. Kuilenburg, M. Wiering, and M. D. Uyl, "A model based method for automatic facial expression recognition," in *Machine Learning: ECML 2005. ECML 2005. Lecture Notes in Computer Science*, vol. 3720. Springer, Oct. 2005, pp. 194–205.

[171] P. Ekman and D. Keltner, "Universal facial expressions of emotion," *California mental health research digest*, vol. 8, no. 4, pp. 151–158, 1970.

[172] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze, "A global hypotheses verification method for 3d object recognition," *Computer Vision–ECCV 2012*, pp. 511–524, Oct. 2012.

[173] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and automation (ICRA), 2011 IEEE International Conference on*. IEEE, May 2011, pp. 1–4.

[174] M. Thiebaux, S. Marsella, A. N. Marshall, and M. Kallmann, "Smartbody: behavior realization for embodied conversational agents," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, Jan. 2008, pp. 151–158.

[175] S. Rose, M. Wynne, and A. Hellesoy, *The Cucumber for Java Book: Behaviour-Driven Development for Testers and Developers*, ser. Pragmatic programmers. Pragmatic Bookshelf, 2015.

[176] D. Brickley and L. Miller, "Foaf vocabulary specification 0.91," Technical report, ILRT, Tech. Rep., 2007.

[177] E. Christodoulou, C. Christophorou, G. Samaras, P. Andreou, J. M. Dias, J. A. S. Prado, and P. Alvito, "Socialrobot d1.1 specification of user needs and requirements," SocialRobot, Tech. Rep., 2012.

[178] ——, "Socialrobot d1.2 use case scenarios development and ethical privacy considerations," SocialRobot, Tech. Rep., 2012.

[179] B. Rosner, *Fundamentals of Biostatistics*. Cengage Learning, 2010.

[180] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection. corr abs/1506.02640 (2015)," 2015.

[181] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*, Aug. 2008, pp. 5–13.

[182] T. Erl, *Service-Oriented Architecture: Concepts and Technology and Design*. Prentice Hall, 2005.

[183] A. Cangelosi, G. Metta, G. Sagerer, S. Nolfi, C. Nehaniv, K. Fischer, J. Tani, T. Belpaeme, G. Sandini, F. Nori, L. Fadiga, B. Wrede, Katharina, Rohlfing, E. Tuci, K. Dautenhahn, J. Saunders, and A. Zechel, "Integration of action and language knowledge: A roadmap for developmental robotics," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pp. 167–195, Jun. 2010.

[184] J. J. Kuffner, "Cloud-enabled humanoid robots," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, Dec. 2010.

[185] C. Costea, "Applications and trends in mobile cloud computing," *Carpathian Journal of Electronic and Computer Engineering*, vol. 5, p. 57, Jan. 2012.

[186] B. K. Kim, M. Miyazaki, K. Ohba, S. Hirai, and K. Tanie, "Web services based robot control platform for ubiquitous functions," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, Apr. 2005, pp. 691–696.

[187] L. Vasiliu, B. Sakpota, and H.-G. Kim, "A semantic web services driven application on humanoid robots," in *Software Technologies for Future Embedded and Ubiquitous Systems, 2006 and the 2006 Second International Workshop on Collaborative Computing, Integration, and Assurance. SEUS 2006/WCCIA 2006. The Fourth IEEE Workshop on*, Apr. 2006.

[188] Y. Chen, Z. Du, and M. García-Acosta, "Robot as a service in cloud computing," in *Service Oriented System Engineering (SOSE), 2010 Fifth IEEE International Symposium on*, Jun. 2010, pp. 151–158.

[189] M. Waibel, M. Beetz, J. Civera, R. d'Andrea, J. Elfring, D. Galvez-Lopez, K. Häussermann, R. Janssen, J. Montiel, A. Perzylo *et al.*, "Roboearth," in *IEEE Robotics & Automation Magazine*, vol. 18, no. 2. IEEE, Jun. 2011, pp. 69–82.

[190] R. Arumugam, V. R. Enti, L. Bingbing, W. Xiaojun, K. Baskaran, F. F. Kong, A. S. Kumar, K. D. Meng, and G. W. Kit, "Davinci: A cloud computing framework for service robots," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 3084–3089.

[191] J. L. Crowley, J. Coutaz, G. Rey, and P. Reignier, "Perceptual components for context aware computing," in *International conference on ubiquitous computing*, Sep. 2002, pp. 117–134.

[192] J. L. Crowley, P. Reignier, and R. Barranquand, "Situation models: A tool for observing and understanding activity," in *Workshop on people detection and tracking, 2009 IEEE international conference on robotics and automation (ICRA)*, May 2009.

[193] P. Pawar, H. Boros, F. Liu, G. Heijenk, and B.-J. van Beijnum, "Bridging context management systems in the ad hoc and mobile environments," in *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, Jul. 2009, pp. 882–888.

[194] Y. Kalfoglou and M. Schorlemmer, "Ontology mapping: the state of the art," *The knowledge engineering review*, vol. 18, no. 1, pp. 1–31, Jan. 2003.

[195] M. Ehrig and S. Staab, "Qom-quick ontology mapping," in *International Semantic Web Conference*, vol. 3298. Springer, Oct. 2004, pp. 683–697.

[196] N. F. Noy and M. A. Musen, "Smart: Automated support for ontology merging and alignment," in *Proc. of the 12th Workshop on Knowledge Acquisition, Modelling, and Management (KAW'99), Banf, Canada*, Oct. 1999.

[197] N. F. Noy, M. A. Musen *et al.*, "Algorithm and tool for automated ontology merging and alignment," in *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00). Available as SMI technical report SMI-2000-0831*, Aug. 2000.

[198] ——, "Promptdiff: A fixed-point algorithm for comparing ontology versions," *AAAI/IAAI*, vol. 2002, pp. 744–750, Jul. 2002.

[199] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder, "An environment for merging and testing large ontologies," in *KR*, Apr. 2000, pp. 483–493.

[200] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, "Learning to map between ontologies on the semantic web," in *Proceedings of the 11th international conference on World Wide Web*. ACM, May 2002, pp. 662–673.

[201] M. S. Lacher and G. Groh, "Facilitating the exchange of explicit knowledge through ontology mappings." in *FLAIRS conference*, May 2001, pp. 305–309.

[202] S. Prasad, Y. Peng, and T. Finin, "Using explicit information to map between two ontologies," in *Proc. of the AAMAS 2002 Workshop on Ontologies in Agent Systems (OAS'02), Bologna, Italy*, Jul. 2002.

[203] P. Mitra and G. Wiederhold, "Resolving terminological heterogeneity in ontologies," in *Proceedings of the ECAI workshop on Ontologies and Semantic Interoperability*, Jul. 2002.

[204] E. Compatangelo and H. Meisel, "Intelligent support to knowledge sharing through the articulation of class schemas," in *Proceedings of the 6th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES'02), Crema, Italy*, Sep. 2002.

[205] Z. Michalewicz and D. B. Fogel, *How to solve it: modern heuristics*. Springer Science & Business Media, 1999.

[206] J. Wang, Z. Ding, and C. Jiang, "Gaom: Genetic algorithm based ontology matching," in *Services Computing, 2006. APSCC'06. IEEE Asia-Pacific Conference on*, Dec. 2006, pp. 617–620.

[207] A. Tversky, "Features of similarity." *Psychological review*, vol. 84, no. 4, p. 327, Jul. 1977.

[208] C. J. V. Rijsbergen, *Information Retrieval*, 2nd ed. Newton, MA, USA: Butterworth-Heinemann, 1979.

[209] M. A. Rodríguez and M. J. Egenhofer, "Determining semantic similarity among entity classes from different ontologies," *IEEE transactions on knowledge and data engineering*, vol. 15, no. 2, pp. 442–456, Mar. 2003.

[210] R. N. Shepard and P. Arabie, "Additive clustering: Representation of similarities as combinations of discrete overlapping properties." *Psychological Review*, vol. 86, no. 2, p. 87, Mar. 1979.

[211] J. Euzenat, H. Stuckenschmidt, and M. Yatskevich, "Introduction to the ontology alignment evaluation 2005," in *Proc. K-Cap 2005 workshop on Integrating ontology*, Oct. 2005, pp. 61–71.

[212] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[213] Y. Li and D. Schuurmans, "Mapreduce for parallel reinforcement learning," in *Recent Advances in Reinforcement Learning: 9th European Workshop, EWRL 2011*. Springer Berlin Heidelberg, Sep. 2012, pp. 309–320.

[214] F. A. Oliehoek, "Decentralized pomdps," *Reinforcement Learning: State-of-the-Art*, vol. 12, pp. 471–503, 2012.

[215] M. Khedr, "An infrastructure for managing context information in pervasive computing environments," Ph.D. dissertation, University of Ottawa, Nov. 2004.

[216] N. F. Noy and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology," in *Knowledge Systems Laboratory, Standford University*, Mar. 2001.

[217] w3c. w3crdf. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/#section-Concepts. Visited on 2017-08-07.

[218] ——. w3cowl. http://www.w3.org/TR/owl-features/. Visited on 2017-08-07.

[219] ——. w3cswrl. http://www.w3.org/Submission/SWRL/. Visited on 2017-08-07.