UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

PH.D. THESIS

IN

INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

HUMAN-COMPUTER INTERACTION IN INTELLIGENT ENVIRONMENTS

DARIO DI MAURO

TUTOR: PROF. FRANCESCO CUTUGNO

COORDINATOR: PROF. DANIELE RICCIO

XXX CICLO

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

Declaration of Authorship

I, Dario Di Mauro, declare that this thesis titled "Human-Computer Interaction in Intelligent Environments", and the work presented in it, are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at the University "Federico II" of Naples.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

“Real progress happens only when advantages of a new technology become available to everybody”

Henry Ford

Abstract

Nowadays, smart devices populate our environments, providing services and being more and more interactive and *user-friendly*. However, they usually require a centralised unit that processes all the dialogues to produce an answer. On the other hand, ubiquitous and pervasive solutions are a valid alternative, but it is hard to arrange them in a well-organised environment. In this thesis, I question if a ubiquitous infrastructure can be reactive, flexible and scalable without disadvantaging a uniform environment. Reactivity defines rapid interactions; flexibility concerns both network issues and interactions with users, through customised interfaces; scalability, instead, ensures that the adopted model does not have constrained networks' size. This investigation focuses on Human-Computer Interaction studies, because people without a required technological background will be the final users of the system.

I propose a novel distributed model where each node is a device that can independently interact with users through natural interfaces; in addition, nodes collaborate with other similar devices to support people. Nodes' intelligence is limited to their own context. In order to improve the collaboration, devices share partial knowledge and have a common strategy to forward requests they are not able to accept. The resulting network is an Intelligent Environment where the *intelligence* comes from a composition of connected interactive behaviours. I investigated the best approach to navigate requests, proposing a routing algorithm and considering also security and consistency issues. I contextualised this work in both a smart house and a smart museum. With the devised process, I paid specific attention to professionals involved in the design steps. I identified actors with different roles and needs; in order to meet their requirements, I proposed a designing process, with automated solutions that simplify the implementation of the presented model.

The system has been tested in simulated scenarios in order to evaluate all the novel parts. Results showed that the designed model is reactive, flexible and scalable. Furthermore, in order to enhance the final outcome, I characterised design patterns to design the network. Future improvements are oriented to the initialisation of the network, that now requires an expert; In addition, a more complex interaction is under investigation to support users in museum visits.

Contents

List of Figures

Dedicated to Naples, where I met my future Wife...

Chapter 1

Introduction

> *Ubiquitous computing names the third wave in computing, just now beginning. First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop. Next comes ubiquitous computing, or the age of calm technology, when technology recedes into the background of our lives.*
> – Mark Weiser

Actually, although technology already supports our activities, and it works in background, I would say that we are living a transitional process where computing is ubiquitous and it is going to be more and more pervasive. This statement is driven by the availability of commercial devices, based on Artificial Intelligence, that are starting to populate our spaces, but they are still separated entities or refer to a centralised system in the most cases. Cloud computing made this possible, by moving operations very demanding on resources online. In cloud-based architectures devices are connected, but they can be controlled through *hubs* that act as interfaces. This is the current approach of products such as Amaxon Echo[1], Google Home[2] and Apple HomeKit[3]. An alternative paradigm decentralises communication, encouraging autonomy of single entities. They establish communications, often relying on strict policies and rules, and treated from different points of view and communities [3–5]. The Internet of Things paradigm is a clear example of that; sensors are disseminated within the environment, often sending huge amount of data towards central unit that process them.

A growing need concerns what Weiser called *"calm computing"* [6]; it is considered one of the key challenges of a world crowded of smart objects. The focus is that technology

[1] https://en.wikipedia.org/wiki/Amazon_Echo retrieved on October 2017
[2] https://en.wikipedia.org/wiki/Google_Home retrieved on October 2017
[3] https://developer.apple.com/homekit/ retrieved on October 2017

should inform users, without being obtrusive [7]. This arises Human-Computer Interaction issues, reformulating goals that are not just related to the users' perception of a single point of interaction, but concerns a profound change passing from *proactive computing* to *proactive people* [8]. This aspect is also often considered in complementary research fields, such as Intelligent Environments [9], where technological solutions are defined to promote an engaged living, enabling people to do what they want, and not the opposite. Of course, people with special needs can be supported and guided, but in any case, solutions cannot limit creativity and flexibility.

1.1 Motivation

This thesis addresses research questions arisen from considerations seen in the beginning: is it possible to directly connect devices that are limited to their own contexts to support people in a ubiquitous and pervasive approach? Can such a model interact with users customising interfaces and behaviours?

With this orientation, in this thesis I propose a distributed model, where devices constitute nodes of a network. These nodes understand a limited number of requests, but they expose accepted inputs to the neighbours. A common strategy - adopted by all the nodes in the network - relies on this shared knowledge to route a received request if the local intelligence is not able to process it. Each node performs a single step in the resulting path. A particular attention is spent towards Human-Computer Interaction (HCI) issues; in this work, indeed, I highlight the importance of how the interface can be customised on each node and the consequence on the resulting Intelligent Environment. HCI as discussed here are in the theoretical frame of Natural User Interfaces, preferring interaction channels that are uncommon for typical users, such as gestures and voice, over the more classical mouse and keyboards.

The problems discussed in this thesis are not related to a single domain; the proposed model, indeed, uses an ontology that represents how the physical space is organised; the adopted methods use that structure to retrieve knowledge and to perform context-aware operations. Two environments are used as cases of study: a smart house and a smart museum. Experiments have been conducted to test many parts of my system. They concern the assessment of the infrastructure, of the routing strategies and evaluations of the used interfaces.

1.2 Outline of the thesis

This dissertation addresses different perspectives of Human-Computer Interaction and Intelligent Environments. Chapter 2 provides a literature review of studies and approaches pertaining to my work. Chapter 3 presents the model, which all the investigations of this thesis refer to. The definition of a single node - also referred as *entity* - is given; I will discuss its features and how it can belong a network of similar entities. The *"Navigation problem"* is eventually defined and described. Chapter 4 is focused on the navigation with a different point of view: a closer look on linguistic analyses performed to route the request is provided; additionally a method to resolve conflicts in the network is investigated. In Chapter 5 the discussion concentrates on how to set an Intelligent Environment up, describing actors involved in the designing steps and tools provided to support them. Chapter 6 proposes experiments to assess the presented elements.

During these years, a number of papers has been published. The contributions closely related to my model are listed below:

1. Dario Di Mauro. *"A Framework to Support Multiple Levels of Interaction"*. DC@ CHItaly. 2015.
2. Dario Di Mauro and Francesco Cutugno. *"A framework for interaction design in intelligent environments"*. 12th International Conference on Intelligent Environments (IE). 2016.
3. Dario Di Mauro, Juan C. Augusto, Antonio Origlia and Francesco Cutugno *"A framework for distributed interaction in intelligent environments"*. European Conference on Ambient Intelligence. 2017.
4. Dario Di Mauro, Antonio Origlia and Francesco Cutugno *"Distributed Processes for Spoken Questions and Commands Understanding"*. 4th Italian Conference on Computational Linguistics. 2017. (accepted)

Other papers, which complete the study here presented, are listed below:

1. Barile, Francesco, et al. *"ICT solutions for the Or.C.He.S.T.R.A. project: From personalized selection to enhanced fruition of cultural heritage data"*. 10th International Conference on Signal-Image Technology and Internet-Based Systems (SITIS). 2014.
2. D'Auria et al., *"A 3D Audio Augmented Reality System for a Cultural Heritage Management and Fruition"*. Journal of Digital Information Management 13.4 (2015).

3. Calandra et al., *"EYECU: an Emotional eYe trackEr for Cultural heritage sUpport"*. Empowering Organizations. 2016.

4. Di Mauro et al., *"PaSt: Human Tracking and Gestures Recognition for Flexible Virtual Environments Management"*. International Conference on Augmented Reality, Virtual Reality and Computer Graphics. 2016.

5. Origlia et al., *"Why so Serious? Raising Curiosity Towards Cultural Heritage with Playful Games"*. AI* CH@ AI* IA. 2016.

6. Cutugno et al., *"Augmented Reality Without Barriers: Dematerializing Interfaces in Cultural Heritage Applications"*. Artificial Intelligence for Cultural Heritage. 2016.

7. Nurgaliyev et al., *"Improved multi-user interaction in a smart environment through a preference-based conflict resolution virtual assistant"*. 13th International Conference on Intelligent Environments (IE). 2017.

8. Origlia et al., *"Establishing a theoretical background for a museum-centric entertainment system"*. 1st Workshop on Games-Human Interaction. 2017.

Other studies, presented with preliminary results or labelled as future works at the end of this thesis, are in progress.

Chapter 2

Background

In this Chapter I review the recent literature about Human-Computer Interaction (HCI) and how it is changing. An extended and complete explanation of HCI will not be presented here. For details, the reader can refer to [10]. An overview of the evolution of Human-Computer Interaction is recalled to motivate how my work is immersed in this context. The focus then moves on Intelligent Environments and their nuances that aims at supporting people in their daily life. The goal of this overview is a discussion about different models and directions. In this thesis I will introduce details of the proposed model. Backgrounds related to those systems will be contextually provided.

2.1 Human-Computer Interaction

Human-Computer Interaction researches how computers and technological solutions should be designed in communicating with users. HCI people both observe the ways in which humans interact with computers and design technologies that let humans interact with computers in novel and natural ways. The term Human-Computer Interaction has been popularised by Card et al. in their book *"The Psychology of Human-Computer Interaction"* [11] although it has firstly used about some years earlier [12]. In the evolution of computer science, researchers reviewed opinions and importance related to this field; HCI is gaining, indeed, attention from both academic and industrial actors to propose new models and products. As a research field, HCI involves computer scientist, as well as psychologists, designers, behavioural scientists and many other profiles.

Human-Computer Interaction is particularly focused on interfaces, but communication and interaction involve other aspects as well. For this reason, HCI is one of the bricks of a bigger wall called Interaction Design, widely oriented in creating a user experience

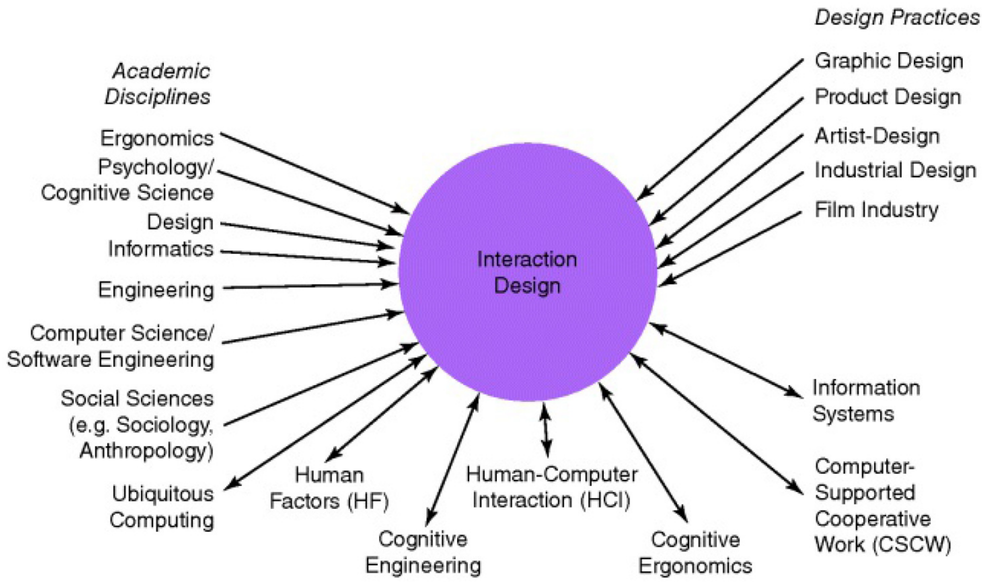


FIGURE 2.1: Roles involved in Interaction Design. Source: [1]

that enhance people's life. Figure 2.1 summarises roles involved in Interaction Design according to Rogers et al. [1]. Winograd defines it as *"designing spaces for human communication and interaction"* [13, p. 160], while Saffer views it from a social point of view, describing Interaction Design as *"the art of facilitating interactions between humans through products and services"* [14, p. 5].

People interact with devices in many ways. Interfaces actually realise the interaction which is crucial to facilitate the whole process. With the introduction of Graphical User Interfaces (GUI), interaction passed from buttons and switches, passing through command-line interfaces, to mouse, keyboards and displays. With the emerging of multimodal interaction [15, Chap. 21], Voice User Interfaces and enhanced GUIs, humans have been allowed to interact with embodied characters agents. Haptic interfaces have been investigated and improved the experience as well [16].

In using a product - technological or not - each user creates a *mental model*. *Mental models* have been initially introduced by Craik in 1943, as stated by Nersessian [17] and researchers hypothesised they play a major role in cognition, reasoning and decision-making. Users build mental models of the world around themselves; people daily use these representations in working life. However, since users do not actually need to fully understand internal and complex mechanisms, interaction designers should *"manipulate"* the perceived mental model, by providing *Represented model* through interfaces. They aim at suggesting behaviours that could support people in using a product. Represented models have been discussed in [18].

2.1.1 What are Natural User Interfaces?

In interacting with objects such as technological devices, people perform cognitive processes that *extend* physical boundaries of their body, assuming that the object is part of themselves and, therefore, it can be used in typical problem-solving operations. This process goes under the *cognitive extension* [19, 20]. This concept is part of HCI because designers propose solutions that have an effect to cognitive processes of users.

By introducing interaction with unconventional devices, the cognitive extension gains importance, since the interaction assumes a different connotation. Natural User Interfaces (NUI) are an example of this aspect. Steve Ballmer, CEO Microsoft, claimed:

> *I believe we will look back on 2010 as the year we expanded beyond the mouse and keyboard and started incorporating more natural forms of interaction such as touch, speech, gestures, handwriting, and vision what computer scientists call the "NUI" or natural user interface.*

Actually NUI let a debate to emerge within the related scientific community. *Natural refers to the users' behaviour and feeling during the experience rather than the interface being the product of some organic process* [21, Chap. 2]; the concept of naturalness for this type of interaction has been questioned [22]. Nevertheless, NUIs surely stimulated researchers in proposing innovative interfaces that bring forward concepts of traditional computing. The main idea behind NUI is that technology must adapt on users, but NUIs are not just a "veneer over a GUI". Doubts about naturalness come from many factors, including how much learning is required and how much effort in doing it: is it easier to shake hands to change a TV channel? Or pressing a button would be easier? The former approach apparently complies with the NUI definition, while the latter should generate a quicker action. *Sometimes a gesture is worth a thousand words. Other times, a word is worth a thousand gestures. It depends on how many functions the system supports* [1]; anyway, a product with a natural user interface should mirror users' capabilities, where users are *experts* that already have experience with similar systems.

2.1.2 How Human-Computer Interaction is changing?

As emerged from the previous Section, HCI is changing and the daily introduction of new products extends horizons of computer science and reviews roles of computers in our lives. In this Section I will briefly discuss advanced interfaces investigated in the last years.

Virtual Reality (VR) recurs to a graphical simulation to create *"the illusion of participation in a synthetic environment rather than external observation of such an environment"* [23, p. 3]. VR is not recent field but the evolution of the technological support is providing the opportunity for unexpected developments. VR is opposed to Augmented and Mixed reality, where the representation is not totally simulated, but computer-generated virtual shapes are combined with a real world. In Augmented Reality (AR) virtual representations are superimposed on physical devices and objects, while in Mixed Reality (MR) views of both real world and virtual one are combined [24]. The main difference between AR and MR is that in AR real world is observed through a device - usually the camera of a smartphone -, while in MR the real and virtual worlds are at the "same level".

Smartphones and wearable devices opened new possibility since they began a personal and constant interface towards internet. Smartphones are more and more involved in our lives. The same is not true, yet, for wearable such as Google glasses, Microsoft Hololens, etc. but they are in overwhelming progress. Diverse uses of smartphones can be found in literature. D'Auria et al. proposed Caruso [25], an interactive audio-guide based on dynamic 3D sounds in the Cultural Heritage domain. The interaction aimed at being totally based on sound and voice so the smartphone was not an obstacle between the listener and real life. In order to reach an Audio Augmented Reality, the adopted headphones were not acoustically isolating, the 3D soundscape added a virtual layer to the real one. Spatialised sounds are not the only interface possible in HCI: studied alternatives are based on non-speech audio such as auditory icons [26], deeply adopted in Operating Systems, or *earcons* [27].

An innovative interface that hides technology within real world was based on tangible interaction. In tangible interfaces, physical objects are coupled with digital representations. When a person manipulates such objects, sensors, hidden inside, detect and analyse the signals through a computer system. The object itself presents an output; it is usually a vibration, sound or animation [28]. Tangible interfaces has been adopted in many fields, but they have recently adopted in cultural heritage (CH) scenarios [29]. Tangible interfaces find a fertile ground in CH because they are *fully integrated into an exhibition, and they extend and complement its materiality and design identity* [29–32].

Touch-based or touchless-based interaction stimulated researchers and produced interesting projects in the last years. However, they sometimes lack of controllers or interfaces that would provide a physical feedback. This is the reason behind very unusual approaches based on levitating displays that dynamically compose tangible shapes [33]. These objects can be manipulated and warped through touch and the system will provide a tactile, visual and audio feedback.

Interfaces are designed for users, and design approaches are focused on this aspect. The role of the user has been made central with the User-Centred Design (UCD), firstly introduced by Norman [34]. A sample of final users is involved in many phases of the design process of a system. Interfaces are just a part of the goal, but it gives extensive attention to usability, considering the user characteristics, environment and tasks. UCD differentiates from other similar approaches in phases that involve final users. Participatory Design [35] - or Cooperative Design - involves designers and users on an equal footing; while other classical System-Centred Design models consider users just at the initial - requirements analysis -.

In order to cope with this highly evolutionary layer, an innovative and stable system should abstract by the channels adopted for interaction. In this thesis I propose a model that abstract from the adopted interface. An input manager receives data from separated input devices and the infrastructure can be configured to choose the right devices and fusion process. Details can be found in Chapter 3.2. My solution is inspired to already existing frameworks [36], used to receive and analyse human signals.

2.2 Intelligent Environments

> *Slowly and silently technology is becoming interwoven in our lives in the form of a variety of devices which are starting to be used by people of all ages and as part of their daily routine*

This has been stated by Augusto et al. [9]. The fast growing of technological availability required the confluence of more similar and complementary fields. Many of them concern technological advancements; others, as seen above, refer to various domains. The technological part comprises pervasive/ubiquitous computing [37], smart environments (SE) [38, 39] and Ambient Intelligence (AmI) [40]. In this background, Intelligent Environments (IE) represent a synthesis of all of them. Sensing devices (from SE) populate the environment and *"a digital environment [...] supports people in their daily lives in a nonintrusive way"* [41], through intelligent software products (from AmI concepts) [42]. The whole system bases on the pervasive and ubiquitous availability of resources.

Despite the name of the field, IE solutions have been proposed in recent years to improve people' lives, considering technology as *support*, subordinated to users' needs. Ambient Assisted Living (AAL) systems [43] are an example of these studies. They have enormous potential to compensate for the cognitive and physical deficits of older people [44] without being an obstacle to those who do not have these deficits.

In this thesis I will present a ubiquitous model that creates an Intelligent Environment, by connecting independent devices. They interact with users and have common solutions to support them. This Section proposes a literature review; different relevant approaches are explained compared with goals of my system. Since my model aims at working in multiple domains, I chose to focus on two of them: smart houses and smart museums. Section 2.2.1 will present architectures and methods found in both the environments; in Section 2.2.2 I will disclose details about my proposal and how similar issues has been studied. Section 2.2.3 completes the comparison focusing on complementary issues.

2.2.1 Architectures

Intelligent Environments have usually divided in two groups: one includes approaches based on an integrated infrastructure [45] and another that distributes the interaction on multiple devices [37]. This Section proposes a review of distributed models that copes with ambient intelligence or intelligent environments. These issues have been investigated in different fields, introducing techniques usually adopted in other disciplines. Ubiquitous computing [37] (ubicomp) is the first paradigm that introduced new concepts and points of view. In contrast to desktop computing, ubicomp spreads and diffuses technological support within the environment; users interact by means many devices connected through internet connections. Ubicomp paradigm has been treated with many aspects. It is referred as pervasive computing, internet-of-things (IoT), *"everyware"*. However, their difference is not just in the name, but also in their goal. Ubicomp is usually oriented to Human-Computer Interaction aspects, while Pervasive computing is oriented to networking and processing of data they produce [9].

IE inspired many studies related to both closed and open spaces. Among closed areas, smart homes occupy a relevant position. *"These environments are usually rich, complex, unpredictable, possibly generating substantial 'noisy' data, unstructured and sometimes highly dynamic (i.e., they change continuously or at least often)"* [9]. Houses are interesting fields of investigation for many goals: controlling, monitoring and automating [46–48], safety [49], entertainment [50] that all contribute to the well-being improvement.

Recent studies [51–53] applied Multi-Agent System (MAS) strategies to control and plan [54, 55] intelligent environments. MAS are based on the *belief-desire-intention* paradigm and are a proper compromise between centralised and distributed solutions: agents provide services, and they are invoked, according defined policies, to bring intelligence and responsivity to the environment. The agents *"have characteristics such as autonomy, reasoning, reactivity, social abilities and pro-activity which make them appropriate for developing dynamic and distributed systems based on AmI"* [56]. On the other hand, MAS solutions often require a centralised system to process computational tasks or interactive steps: *yellow-pages* providers or the *master-agent* are example of this idea. The adoption of MAS techniques and approaches help to reuse consolidated studies and tools [57, 58].

Cultural Heritage (CH) represents a worldwide resource which attracts millions of visitors every year to museums, exhibitions, monuments and historical centres. In CH, IoT has been adopted in different situations, from preservation solutions [59] to learning setups [60]. Technology is disseminated throughout an area of interest and users interact with each device by means of their personal device - usually an improved version of a mobile guide -. IoT points are used as monitoring hubs [61] or interactive Points of Interest (POIs). Chianese and Piccialli [62] proposed an environment setup where each work of art interacts with users by means of a smartphone. Devices gather information from a remote central unit, retrieving settings and cultural data for the visitors; then respond to the user. The system has been then extended [63], enhancing the infrastructure and including a story generator. In Alletto et al. [64] the authors described an IoT infrastructure based on a set of wearable devices acting as audio-guides. Each personal device was able to communicate with other devices. The system locally recognised the target work of art by recurring to image processing. In addition, dedicated distributed modules detected anonymous activities in known areas to manage lighting and heating.

Visitors' behaviour in CH has been always an object of interest for researchers. Technology introduction extended horizons because it enables novel and cheap devices to propose a new experience to users. However, the presented solutions offer novel points, still focused on single users; other studies [65, 66] are going beyond a single user interaction to support groups in museums. Korzun et al. [67] presented a dynamic knowledge base, where devices that interact with visitors and museum professionals compose an IoT infrastructure where historical information enriched the knowledge base itself. New acquired knowledge is then presented to future users and this approach stimulate visitors' interaction.

Technology in Cultural Heritage motivated many research projects in last decades. Part of them aimed at proposing personalised experiences; this is the case of PEACH (Personal Experience with Active Cultural Heritage) [68], where the authors modelled users to adapt interfaces and propose personalised contents. A similar goal inspired the CHESS project [69] (Cultural Heritage Experiences through Socio-personal interactions and Storytelling), where digital storytelling aimed to adapt contents on visitors' needs. The interfaces were mainly mobile-based and this aspect led visitors to see the museum through a display. The meSch project [70] (Material EncounterS with digital Cultural Heritage) proposed an architecture for tangible interaction, bridging the gap between visitors' on-site and on-line experience. Physical objects were enriched by digital content and collaborate to present information to users in a personalised way and collaborating with other objects to check particular conditions. Interesting is the use of replicas [29] as means of controlling an interactive experience.

2.2.2 Routing

The basic idea behind this thesis is to provide a distributed network of entities, where each node interacts with the user through multimodal interaction following NUI guidelines. Knowledge is local to the node and limited to its own provided services. If the node is not able to produce an output coherent with the request, it sends the received message to the others, without a prior determined target node. Entities operate with partial knowledge about the others and their accepted requests, so nodes in the final path do not always understand the request.

In the introduced configuration, one of the biggest challenges concerns the routing of requests in uncertain situations. The routing problem has a long history, but it gained a strong interest with IoT. If hundreds of sensors would use a domestic network infrastructure, then a congestion is highly probable [71, 72]. For this reason many studies proposed solutions for load balancing and network optimisation. Many routing solutions are also based on biological behaviours such as ant colonies [73, 74]. Although my system does not theoretically limit the number of available devices, the expected workflow is different: devices are independent and, in forwarding requests, cannot calculate the best path without knowing the target node.

The situation proposed in this thesis can be compared to ad hoc networks, also known as MANET (Mobile Ad hoc NETworks) [75], where connections are dynamic, communication does not rely on a central unit acting as routers. MANETs have been used in different scenarios, and studies are also focused on safety, where a physical infrastructure misses due disasters and a network is built upon available smartphones [76]. Another similarity is with a particular MANET, mesh networking, where each peer receives and relays data for the network. However, like in MANETs, these approaches use *flooding* of *routing* techniques to relay packets: in the former case, data are transmitted to all the links, except the one the packet arrived on; in the latter case, instead, the packet is transmitted to a single peer, in order to be closer to the destination.

The system proposed in this thesis builds upon the previous works by enalrging their scopes: without knowing the target which should receive the navigating request, current approaches cannot be directly applied; on the other hand, a broadcast is not preferred because it would easily overload the network in an IoT infrastructure. Mesh networks are closer to my proposal, but they still rely on finding a path towards a *known* target. The solution I propose in this thesis considers contextual information and moves the routing problem to a higher level, where the message, the environment and previous interactions affect the final navigation.

2.2.3 Support for designers

An Intelligent Environment requires many efforts to be set up. It is not only a job performed by technicians and all the involved actors need support both at design time and at run-time with monitoring. While this seems to be a common direction [7, 77], in Cultural Heritage some solution has been proposed. In museums, an open issue is how to realise a visualisation tool, useful for curators [78], to simulate and/or understand how visitors are experiencing the exhibits and contents provided to them [79]. Measures taken into account are: total time on an area or particular exhibits, paths and visitors' behaviours and level of engagement [80–82]. Other important aspects are attraction and holding powers of each item [83]. By introducing interactive technology in museums, researchers detected visitors' impressions with questionnaires after the visit [29]. This process has been automated by adopting dedicated devices [84]. Personal mobile guides changed the visitors' behaviour, making the users to be more attracted from exhibits with more information, but they reduced social interactions [85, 86].

In IoT, a scenario where exhibits independently interact with users, technology offers a different solution to monitor visitors' behaviour, fostering the positive aspects of a mobile guide, and reducing negative side effects. Each node in a ubiquitous environment monitors part of the museum in background and communicates updates to a remote unit. Chianese and Piccialli [87] explored this approach. Each device in the environment can be constantly controlled and monitored and, as a consequence, a museum practitioner has a responsive status about the museum.

In this thesis I will propose a more structured approach. I identified a set of actors that, with different roles, collaborate to create a smart environment. Their work is supported by tools that help them in all the relevant processes and give the possibility of a run-time monitoring.

2.3 Summary

In this Section I presented some background works behind the idea of this thesis. I presented a review of Human-Computer Interaction and how this motivated part of my work. The proposed idea builds upon the need of a novel architecture for ubiquitous computing with interfaces easy to customise.

My system has been tested in smart house and smart museum contexts, where the Intelligence perceived by the users is built upon a collection of partial nodes' intelligence. One of the main contribution is on the general architecture, organised to be easily customised for multiple needs: domains, behaviours, roles, etc.

One of the most challenging issues is in the routing problem, where nodes adopt a common strategy to deliver the request to an undetermined target node with partial knowledge about both the environment and the request. A comparison with existing and recent approaches with IoT solutions has been proposed. The Chapter ended with an overview of tools and approaches to support practitioners of Intelligent Environments.

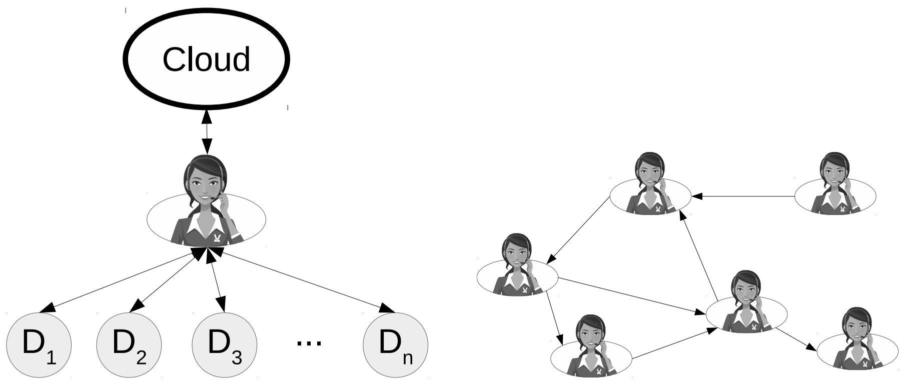
The next Chapters will extensively discuss my model, details of enhancement performed and how designers and professionals are supported in setting an Intelligent Environment up.

Chapter 3

Distributed model based on partial knowledge sharing

What does "intelligence" mean? In Intelligent Environment (IE) this term usually refers to Artificial Intelligence applied to environments, where technology offers something more than static rooms [9]. In Chapter 2 I discussed different solutions to make an environment "intelligent", but this thesis proposes a different point of view and a model to represent it.

Currently, business products base their success on virtual assistants, able to process all requests or, at least, to be smart enough to provide a response to a large set of questions. The approach they follow is represented in Figure 3.1(a). People and devices interact with a single virtual assistant that is the main bridge between cloud and real world. This approach has clear advantages: easy to maintain, easy to update, easy to



(a) A centralised model for an IE

(b) A distributed model where each node exposes the same interface

FIGURE 3.1: Comparison of the general structures to model interaction in Intelligent Environments

control. Drawbacks are also visible: an assistant limits parallel interactions, rapidly overloads the network and allows a malicious provider to easily control all the traffic. A distributed model solves the discussed problems. It does not centralise the network on a single node, but it assumes a democratic structure where each node in the network has its own independence. Each entity is able to interact with users in the environment and collaborates with other nodes for other tasks. However, if each node exposes the same interface - i.e. an avatar, as in Figure 3.1(b) - the user perceives the network as a single interface.

The main goal of this thesis is proposing a framework, called PHASER (**P**ervasive **H**uman-centred **A**rchitecture for **S**mart **E**nvironmental **R**esponsiveness). It supports people in an environment populated of devices with a common representation of the domain. Devices are nodes of an ontology and maintain a partial knowledge about other entities. This Chapter proposes a distributed solution, providing a ubiquitous environment where global intelligence is built upon single entities that show responsive behaviours and collaborate with each other to better support the user. It starts with Section 3.1 by explaining the adopted ontology to represent the environment. In Section 3.2 we introduce PHASER, model and features of a single node of the proposed framework. The chapter continues with Section 3.3 with the explanation of a network of PHASER nodes and their strategies for communication. Implementation details will be showed in Section 3.4. The Chapter ends with a summary, where I discuss scenarios that explain common issues solved with PHASER.

3.1 Representation of the environment

PHASER nodes rely on a common ontology that represents the environment. The ontology establishes how the physical space is organised both in terms of space and of available devices. Its goal is to assign roles, class and environment to each node.

In this Section I will describe the ontology and how it is accessed. A specific attention is eventually paid to the domains in which experiments are held in this thesis, a smart house and a smart museum.

3.1.1 Ontology

The ontology adopted in PHASER reports how the physical space is organised and how physical devices, that interact with users, populate each area. Figure 3.2 reports the structure, a tree-based organisation that defines environments and devices that populate them. It is detailed as follows:

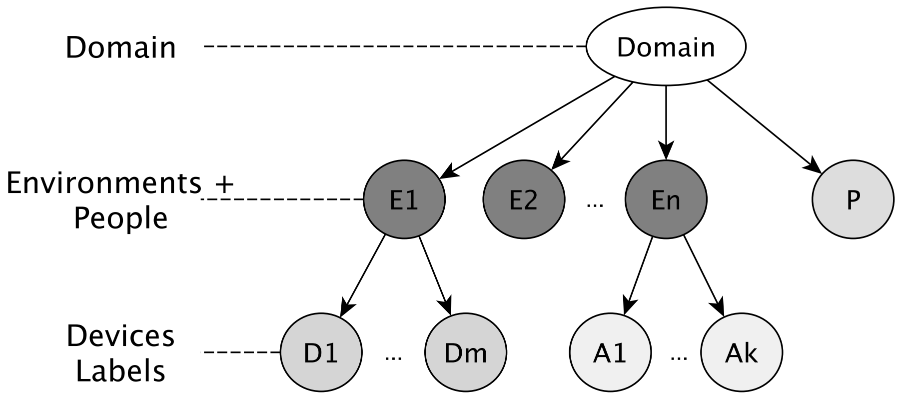


FIGURE 3.2: The structure of an ontology accepted in PHASER. Devices may have different labels. Here they are represented with different grey levels

- the root of the tree represents the domain; it defines in which context devices are operating;
- a number of environments are connected to the *domain*: they represent physical areas that compose the real world. At the same level, and connected to the domain, a particular class of entities is found; they do not have a semantic relation with a single environment. People, personal devices and devices that move through different zones are considered in this class;
- each area contains objects; they are the devices that realise the interaction with users. Objects may have multiple labels; they represent different classes of devices; similar devices working in two or more environments (i.e. lights) are specified as different devices, each belonging to its own zone.

Each instance of PHASER is an entity which belongs to a leaf of the discussed tree. They are physical or simulated devices that require a configuration including information about the adopted ontology. Details about that will be presented later in this Chapter.

By specifying a class, PHASER derives at run-time information such as the environment and other devices in the same room. Running devices use the gathered information to better support people; during the interaction, nodes discover entities and exchange messages with strategies I will in Section 3.2. The environment organisation regulates both the processes and fosters traffic among nodes in the same environment.

PHASER relies on a new developed representation of an environment, stored in a graph database. I also considered the integration with well-known ontologies in this field: soupa [88] and iot-lite[1].

[1]http://iot.ee.surrey.ac.uk/fiware/ontologies/iot-lite retrieved on September 2017

3.1.2 Ontology Manager

The ontology is accessed by an Ontology Manager that provides read-only information - i.e. the information stored in the ontology - and supports nodes during the interaction.

The Ontology Manager is intended to be remote service each node refers to. This aspect does not hinder the scalability of the network because, although each node requires a connection with the same structure, it is intended not to be often invoked.

The Ontology Manager stores running nodes and information about their reachability. This is particularly useful to maintain a global overview of the system and to ensure that interactions do not generate conflicting situations within the environment. I will extensively discuss the latter aspect in Section 4.2. In order to support conflict resolution, the Ontology Manager provides nodes' information; this approach is known in Multi-Agents Systems as *yellow-pages* service.

3.1.3 Considered environments

By adopting a common ontology, my model is not limited to a single domain, but is able to work in different contexts with different needs. Above in this Section I described how the physical space is represented. However, it is not actually related to a particular domain. In this Section, I will present the contexts considered in my thesis. I focused on two futuristic contexts, with different features and needs, to demonstrate that my framework, PHASER, is able to cope with issues and needs of different contexts.

I focused on a smart-house and a smart-museum. They have in common a physical organisation partitioned in areas, and groups of people interacting with devices. However, users' needs are very different and this reflects how they behave. The presented ontology is populated with devices that realistically can belong to the related context.

House

My concept of smart house is an environment populated of devices that provide services. Each entity is connected to others, but it does not rely on them. It means that each object is highly focused on its own context but it has a strategy to communicate with other networked similar devices. An example is depicted in Figure 3.3. I adopt a distributed model that does not require a central unit to process requests. An alternative configuration centralises all the processing and controls each detected signal. A centralised model is usually adopted in Ambient Assisted Living where all the efforts are focused on monitoring and recommending - usually for a single elderly person - [89, 90].

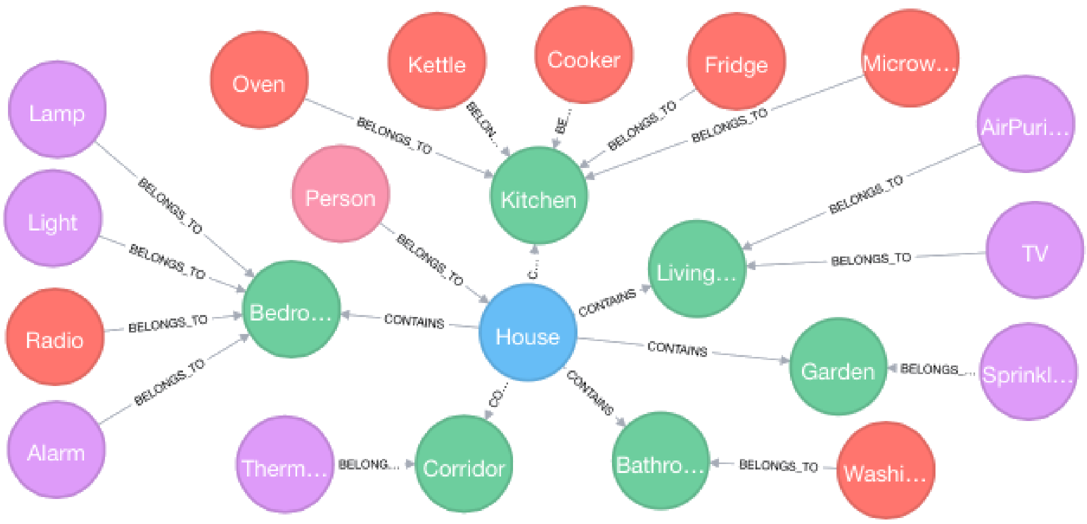


FIGURE 3.3: An example of the adopted ontology to represent a smart house

However, a distributed approach is efficient with a dynamic environment because it is scalable, modular and easily supports multi-users interaction.

The environment I consider is not limited to a single person, but takes into account a group of members with different needs. The situation I expected for a smart house is where a small group interacts with devices; the number of nodes is usually higher than the number of people. Devices operate on shared resources (lights, heating, etc.) so conflicts may arise. This scenario requires the definition of strategies for conflict resolution explained in Chapter 4.

A smart house context needs to tailor interactions on users relying on long-term interactions. These are typical in a home, because people may use the same device multiple times. Eventually, in a smart house, people are more willing to provide personal sensible information.

Museum

The second context I considered is a smart museum, where visitors interact with paintings and other artefacts. A smart museum has different needs, because visitors want to customise interaction with works of art; they usually dedicate few minutes for each item and many of them will be ignored. Figure 3.4 depicts an example of the adopted museum. It is divided in seven artistic movements each representing an area.

The main objective of each interactive work of art is to guide the user throughout the museum in a distributed way by mixing needs of both the visitor and the curator of the

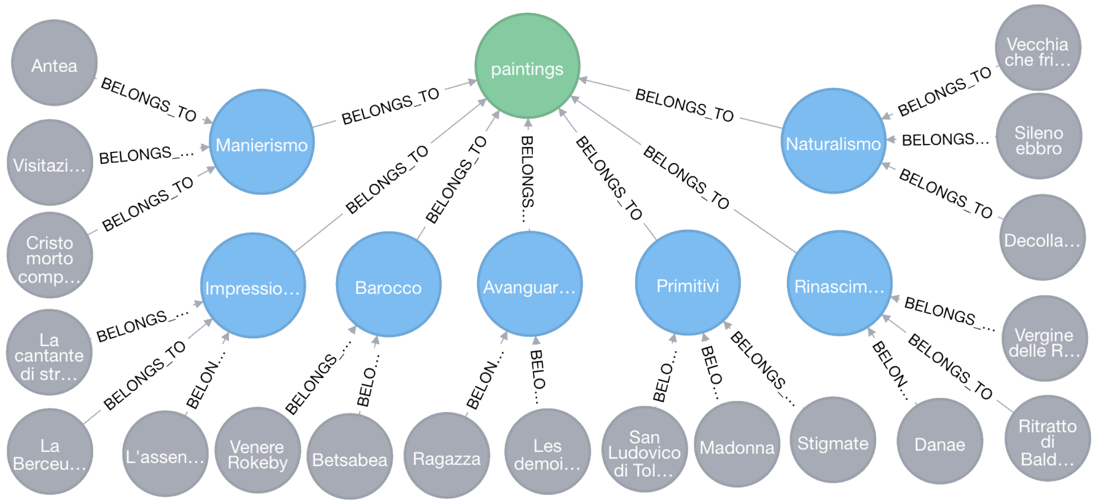


FIGURE 3.4: An example of the adopted ontology to represent a museum

museum. Nodes are connected with a topology designed by an expert and the interaction should foster these connections.

A museum may be highly populated, so redundant remote communication should be avoided, especially in a distributed context with many requests. However, differently from a smart home scenario, a smart museum usually does not modify common resources, so it does not require a conflict resolution step. However, this may be included in particular cases; some exhibitions may require lighting changes or play sounds. This affects global resources and may be controlled through dedicated procedures. A conflict resolution strategy will be discussed in Section 4.2.

3.2 PHASER

In this section I will present my model and its implementation in a framework, PHASER. In my concept, PHASER gives a role to each entity that interacts with others. Possible entities are *objects* and *people* that interact with those objects as well. While objects are devised as networked products that independently conduct dialogues, each person is identified with a personal device that acts as interface between user and objects. Objects may interact with people throughout their personal device or an I/O layer. Personal devices have not sophisticated dialogues managers; they act as a flexible interface that presents users' inputs to the network and propose outputs from objects.

I define an abstract PHASER node which includes the needs of both types of entities. Each node interacts with others, providing services and responding to requests. Some other nodes are mainly focused to a flexible natural interface.

3.2.1 Model

The core part of PHASER is a single node which represents a single entity in a network. The whole approach is based on a distributed environment so other connected entities are generically referred as peers.

By taking inspiration by Dooley et al. [91], I define a single node as a tuple:

$$N = \langle \iota, Cnf_\iota, ClosePeers_\iota, DiscoveredPeers_\iota, oBC_\iota \rangle \tag{3.1}$$

where ι is a unique identifier of the node. *ClosePeers* and *DiscoveredPeers* are sets of related nodes in the environment: ι interacts with those nodes. *ClosePeers* collects connections designed by an expert. *DiscoveredPeers*, instead, indicates links opened during the evolution of the network. oBC collects partial information about the connected peers, acting as Business Cards; they include information gathered from a common ontology and their accepted inputs.

A configuration Cnf determines the behaviour of ι in the environment. It comprises inputs, outputs and the behaviour towards other nodes. In details:

$$Cnf_\iota = \langle name_\iota, type_\iota, class_\iota, env_\iota, I_\iota, O_\iota, P_\iota \rangle \tag{3.2}$$

where *name* is a string representing a display name. *Type*, *class* and *env* determine the role of ι in the environment according an ontology; an expert defines each of those values. P is extensively discussed in Section 3.2.2. I and O represent inputs and outputs respectively; they divide data into channels as in Equation 3.3 for multi-modal interaction according

$$Ch_j = \left(c_j, RG_{c_j}\right) \tag{3.3}$$

where c_x is a channel code and $RG_{c_x} = \left\{r_{i_1}, r_{i_2}, \ldots, r_{i_{c_x}}\right\}$ is a set of accepted inputs/presented outputs. In this thesis I investigated the best structure for each r_x, considering expressiveness and security - in a networked environment -. The first choice was for r_x be regular expressions, and this Section mainly refer to them. Other elements will be discussed in Section 4.1.

If N_{i_ι} and N_{o_ι} are the number of input and output channels, I define I_ι and O_ι in Equation 3.4.

$$I_\iota / O_\iota = \bigcup_{1 \leq x \leq N_{i_\iota / o_\iota}} \{Ch_x\} \tag{3.4}$$

3.2.2 Node's Parameters

As discussed in the introduction of this Section, a node in PHASER should be able to represent different entities in the environment and their characteristics. For this reason, each node includes a set of *Parameters* in its definition. Parameters change the "appearance" each node has, specifying how a node reacts during the interaction; however, their use appears clear when a context is adopted. A designer determines parameters' values for each possible node, and they cannot be changed at run-time.

People can interact with devices by means of the I/O layer each node presents. In other cases, the network configuration can limit this possibility; in this case, people approach a PHASER network by means of their own personal devices - smart-phones, audio-guides, etc - that create a unique interface between user and PHASER network. This can be an approach in museums, for example; furthermore, this way can easily represent a group of people, referenced by the one of the members.

Parameters can be divided into two groups: the former is mainly oriented to the interaction in a network, the latter is focused on group's formation and policies in it. Their adoption is more useful in a smart museum, where roles and customisation highly change the global behaviour. However, they can be adopted in a smart house as well.

Interaction Oriented Parameters

This class of parameters are usually reserved for automatic nodes; they specify how devices should react according a global behaviour and other internal requirements.

Interaction Oriented Parameters specify if a node can *receive connections* (RC), can *respond to requests* (RR), *saves history* of visited nodes (SH), *requires history* of a node when it interacts with it (RH) or can *forward* a request (FR) if the current node does not have a response for it.

In a museum scenario, for example, a work of art (WoA) may have the following parameters P_{WoA}:

$$P_{WoA} = \{RC, RR, RH, FR, G_r, I_f\} \tag{3.5}$$

On the contrary, P_V is a possible set for a single visitor:

$$P_V = \{SH, G_p, G_c\} \tag{3.6}$$

History of explored nodes can be useful in this context to recommend the next object. However, history can be transferred - from the visitor's device V to the current WoA - just if $SH \in P_V \wedge RH \in P_{WoA}$.

Group Oriented Parameters

Group oriented parameters specify how a single device should behave with grouped with other similar nodes. They simulate a group of visitors in a museum, where internal messages are usually shared.

These parameters regulate if a node can be *part of groups* (G_p), can *create new groups* (G_c), if it can *accept new followers* (non-leader members) (G_{acc}). Other parameters specify how a node should behave when interacts with a group: for example if it can *receive requests* coming from a group (G_r) and if it can *interact with followers* (I_f).

By combining those parameters, groups of nodes in the environment will have well-established internal dynamics. For example, a group is "democratic" if the leader (who creates the group) has only the responsibility of creating the group but no privileges are assigned. A group where just *one* node accepts members and requires to control the interaction with other members is a "monarchist" group: just the leader can interact with other - external - nodes.

Members of a democratic group may adopt the set in shown Equation 3.7; a monarchist situation, instead, requires the leader to adopt Equation 3.7 and followers Equation 3.8. However, regardless the setup group, PHASER shares group's messages with all the members.

$$P_D = \{\dots, G_p, G_c, G_{acc}, \dots\} \tag{3.7}$$

$$P_f = \{\dots, G_p, !G_c, !G_{acc}, \dots\} \tag{3.8}$$

In a museum context a democratic group is usually a family or a group of friends who share the experience without any strict difference of roles; a scholarship, instead, may require a stronger hierarchy, where the teacher - or the guide - is the leader and conveys followers questions.

Inference on Parameters

It is clear that some parameters have semantic relations, and a well-formed configuration avoids conflicting features. For example, defining "$\Rightarrow$" as *implies*, "!A" the absence of property A and $\wedge$ as *"and"*, I have that $RR \Rightarrow RC$; $FR \Rightarrow RR$; $G_{acc} \Rightarrow G_p$; $G_c \Rightarrow G_{acc} \wedge G_p$. However, some parameters are optional and depend on the goal of the designer of each node. All the parameters have default values that depend on their role in the network. The role is decided by experts and designers.

3.2.3 Connections

Each node can interact with other peers and their reference is stored in $ClosePeers$; they compose the initial topology designed by a domain expert. However, sometimes new unforeseen connections can be discovered; they are included in $DiscoveredPeers$. Connections represent channels that a node can use to communicate with adjacent entities. New arcs may increase the power of the interaction for the user, because they create other bridges in the network. If a node frequently connects to a discovered peer, this is automatically promoted to $ClosePeer$.

$$DiscoveredPeers = \bigcup (\kappa, c_\kappa, T_\kappa) \tag{3.9}$$

Equation 3.9 defines the set of discovered nodes κ - I refer to them as "partially" connected nodes -, where $c_\kappa \in [0..1]$ is the probability of making this connection fixed: as $c_\kappa = 1$, κ will be included in $ClosePeers_\iota$. T_κ is the last activity of this connection. As ι interacts with κ at time t, c_κ is updated as follows:

$$c'_\kappa = UPC(c_\kappa, t - T_\kappa) \qquad \text{with } t \neq T_\kappa$$

where UPC is defined in Equation 3.10:

$$UPC(x, y) = x + \frac{x}{y\lambda(y)} \tag{3.10}$$

$$\lambda(x) = \phi_0 - (\phi_0 - \phi_1)\frac{1}{1 + e^{k_1 - x}} - (\phi_1 - \phi_2)\frac{1}{1 + e^{k_2 - x}} \tag{3.11}$$

$\phi_{0,1,2}$ and $k_{1,2}$ in Equation 3.11 are constants that derive on the activity of the interaction. According to the activity of the interaction, $\lambda(x)$ assigns a factor ϕ. $\phi > 0$ means that the connection is strengthened, but for $\phi < 0$ it is discouraged. Typical relations are $1 \geq \phi_0 > \phi_1 > \phi_2$, with $\phi_2 \leq 0$ and $k_1 \ll k_2$. By tailoring

Details about the discovery of a node are in Section 3.3.3.

Open connection

As two nodes, ι and κ, open a connection, they share part of their local information composing a personal Business Card (BC):

$$BC_{\iota/\kappa} = \langle name_{\iota/\kappa}, type_{\iota/\kappa}, class_{\iota/\kappa}, env_{\iota/\kappa}, I_{\iota/\kappa} \rangle \tag{3.12}$$

On open, ι and κ add the connection to their $oBC_{\iota/\kappa}$ as follows:

$$oBC_{\iota} = oBC_{\iota} \cup \{BC_{\kappa}\} \tag{3.13}$$

$$oBC_{\kappa} = oBC_{\kappa} \cup \{BC_{\iota}\} \tag{3.14}$$

On startup, ι asks a connection to each $p_i \in ClosePeers_{\iota}$. Partial connections will be opened following details shown in Section 3.3.3; in the latter case, collected BCs will be stored in the same way. Moreover, local information need to be updated:

$$DiscoveredPeers_{\iota} \cup \begin{cases} X_{\kappa}(t) & \text{if } (\kappa, c_{\kappa}, T_{\kappa}) \in DiscoveredPeers_{\iota} \\ \{(\kappa, 0, t)\} & otherwise \end{cases}$$

where t is the current instant and

$$X_{\kappa}(t) = \{(\kappa, UPC(c_{\kappa}, t - T_{\kappa}), t)\} \setminus \{(\kappa, c_{\kappa}, T_{\kappa})\} \tag{3.15}$$

However, $X_{\kappa}(t')$ is added on every interaction between ι and κ, where t' are the considered next instants.

3.3 Network of PHASER nodes

In order to better support the communication, two nodes ι and κ share their own Business Cards as seen before. The result is a network where each node has partial information about its local connections. Each node registers itself to the Ontology Manager, exposing networking information. However, this is a real distributed context because single nodes do not collect all the information, and the Ontology Manager is invoked just for conflict resolution. Details about conflicts will be introduces in Chapter 4.

Users interact with each node in the environment, but the actual topology is hidden for them. They, indeed, perceive the network as a compact system because each node involves other parts as in a centralised system, but PHASER is more flexible than a centralised system.

During a communication, ι may send a request to κ. A request R is a snapshot of the input for ι defined as

$$R = \left\{(c_j, r_{c_j}), 1 \leq j \leq N_{i_\iota}\right\} \tag{3.16}$$

where N_{i_ι} is the number of input channels for ι, c_j is a channel and r_{c_j} is the value of the request on c_j. κ receives R, but it is able to accept it just if it represents a valid input for κ; for an element in the request $R_x = (c_x, r_{c_x})$ this is true if $\exists(c_x, R_c) \in \{Ch_{x_\kappa} \subseteq I_\kappa \mid r_{c_x} \text{ matches on } R_c\}$. "$x$ matches on X" means that $\exists X_i \in X$ so that x complies on the format of X_i. This is wrapped in:

$$m\left(R_x, \kappa\right) = \begin{cases} 1 & \text{if } R_x \text{ is a valid input for } \kappa \\ 0 & otherwise \end{cases} \tag{3.17}$$

A request R is fully accepted by κ if $\sum_{1 \leq j \leq R} m\left(K_j, \kappa\right) = |R|$. I consider $|R|$ and not $|N_{i_\kappa}|$ because the request could not provide information for some channels. However, a fully accepted request is candidate to be manageable, but ι cannot take for granted that κ will process it successfully.

3.3.1 Navigation of a request

As a node interacts with a user, it receives requests and it tries to locally process them. If the node is not able to do it, the system could deliver an error message or share the request within the network. It may broadcast the data, being sure to reach at least one valid node, if it exists, but if multiple available nodes arrive, the starting node should be able to know which is the best one. Moreover, in large networks, many nodes that broadcast information may overload the network itself [92]. A second possible strategy is based on a more intelligent routing process [93], where the node can iteratively forward the request, and ontologies, history and context-awareness [94, 95] could help to enrich system capabilities. The approach I propose in this Section is defined the *"Navigation Problem"* and it is one of the main contributes of this thesis.

By relying on a common ontology and a dynamic topology as shown in Section 3.2.3, each node knows the business card of the adjacent ones. The current node could easily find out how others can successfully process the request and who they are. The approach I propose to solve this problem is a depth-first-search in a distributed graph where a greedy algorithm chooses the local best nodes first. Considered parameters are: current request, past interactions and context-awareness.

As a greedy method on a distributed system, the current node that is not able to locally process a request sorts its adjacent entities in a decreasing order, comparing them with Equation 3.18.

$$Comp(s,c,n,R) = M(R,n) + Toll(s,c,n) + Friend(s,n) \tag{3.18}$$

where R is the current request; s, c, n are respectively the starting, current and the next node in the path; the starting node is who received the user's request. The navigation ends if either c provides a response or too many hops have been done. Please note that, during the forwarding, c is every time different.

$M(R,n)$ is the match degree of the current request with the n's accepted inputs, and it is defined as follows:

$$M(R,n) = \sum_{0 \leq i < |R|} m(R_i, n)/|R| \tag{3.19}$$

The higher $M(R,n)$ is, the higher the probability that n can understand the request R. $M(R,n) = 1$ is a perfect match. R and m have been presented in Equations 3.16 and 3.17.

Tolls are taxes to pay in changing the environment. They have been introduced to encourage connections within the same environment. The $Toll$ function is defined as:

$$Toll(s,c,n) = (-1)^{(E_c - E_n)(E_s - E_n)} \tau\left(E_c, E_n\right) \tag{3.20}$$

where E_A is a unique integer code for the environment of node A, so $E_A - E_B = 0$ *iff* nodes A and B are in the same environment. $\tau\left(x,y\right)$ is a function representing a toll going from x to y. With a positive toll a node implicitly prefers communications within its own environment, but it is not limited to it. However, if the request changes context, it is difficult to fall into the starting environment again.

If needed, $Friends$ assigns a bonus ϕ to requests coming from similar devices. Assuming that T_A is the type of device A in the ontology, $Friend$ is defined as:

$$Friend(s,n) = \begin{cases} \phi & \text{if } T_s = T_n \\ 0 & otherwise \end{cases} \tag{3.21}$$

3.3.2 Depth-first navigation with greedy sorting

If a node x needs to send a request, it uses $Comp(s,x,n,R)$ in the Equation 3.18 to sort the connected nodes. Let s be the environment where the request started, x the current node, n one of the adjacent node and R the current request. Iteratively applying $Comp$, the system obtains a sequence as in Equation 3.22.

$$Sorted_x = \left(p_1, p_2, \cdots, p_{|oBC_x|}\right) \tag{3.22}$$

$$Comp(s, x, p_i, R) \geq Comp(s, x, p_{i+1}, R) \tag{3.23}$$

where relation in Equation 3.23 is valid $\forall i \in [1, \ldots |oBC_x| - 1]$. oBC_x is the business card of node x; it has been defined in Section 3.2.1. Each node in oBC_x will be included in $Sorted$, even if it would not be able to process R.

x will forward R to the peers in the order of $Sorted_x$. However, each single run of the algorithm proposed here is locally conducted; x cannot be sure that the selected "next node" n would be able to accept R: n could be busy or in a wrong state. In that case n will fail in processing.

At step i, peer p_i will be selected if p_{i-1} has failed at the previous step. A final *"local fail"* is arisen by x if $p_{|oBC_x|}$ fails.

A centralised reasoner is not affected by the Navigation Problem, because all the devices run in the same cluster. However, PHASER is intended to be distributed and flexible because the algorithm involves just active connections; if some internal nodes are not available and there is a path towards the final target, the algorithm will reach it.

3.3.3 What PHASER learns from interaction

During the interaction, each PHASER node analyses the messages that had been sent and the received responses. The node may operate with misunderstood commands, but internal parameters will influence next forwards to reuse "good" connections. The main goal of this step is to detect paths that worked in the past, reinforcing them in similar future situations. On forwarding, a node x selectively chooses the nodes that could reply to the user on the submitted request. x has no knowledge about the identity of the "target" t *a priori*. The target is a node able to process the request. However, PHASER optimises the navigation of the request by using *tolls* and *partial connections*.

Tolls

Tolls have been introduced in Section 3.3.1 and they represent a cost to pay if the node forwards the received question to a node in a different environment. With their use, the request follows paths within the same environment, but this behaviour can change during the evolution of the interactions. Requests like *"switch the light on"* are less ambiguous because they may refer to the environment the command starts. Tolls' value are local for each node; a peer reduces a toll if it receives a positive response from a forwarding towards the related environment. By reducing the tax, future comparison will slightly prefer that direction.

Initially, each node has the value:

$$\tau(E_x, e) = \frac{\tau_{max}}{|Env_x|} \qquad \forall e \in Envs_x$$

where τ_{max} is a defined upper limit for each single toll. E_x is the environment where x resides in. Each toll is under the constraint:

$$\sum_{e \in Envs_x} \tau(E_x, e) = \tau_{max} \tag{3.24}$$

By receiving a positive response from node t after a forward, the related toll is updated. Changes affect all the available environments in order to maintain Equation 3.24 valid. Since update functions just involve known environments, it may be possible that a new environment E_t will be explored. In that case $E_t \notin Envs_x$. Updates are formulated in Equations 3.25 and 3.26:

$$\tau'(E_x, E_t) = \begin{cases} \tau(E_x, E_t) - \mu + \max\left\{0, \tau_{max} - \frac{\mu}{|Env_x - 1|}\right\} & \text{if } E_y \in Envs_x \\ \tau_i & otherwise \end{cases} \tag{3.25}$$

$$\tau'(E_x, E_k) = \begin{cases} \min\left\{\tau_{max}, \tau(E_x, E_k) + \frac{\mu}{|Env_x - 1|}\right\} & \text{if } E_y \in Envs_x \\ \tau'(E_x, k) = \tau(E_x, k)(\tau_{max} - \tau_i)\tau_{max} & otherwise \end{cases} \tag{3.26}$$

where E_x, E_t and E_k are environments of the starting node x, the successful target node t and the other known peers k. Eventually, E_t will be added to $Envs_x$.

$\tau_i \in (0 \ldots \tau_{max})$ is the initial value for a toll. μ, τ_{max} and τ_i are constants empirically defined.

Partial connections

Partial connections are introduced to test unforeseen interactions at design time. If a request starts from a node x and reaches an unknown target t, a connection is opened from x to t. The new connection is added to $DiscoveredPeers_x$ shown in Section 3.2.1. As well as permanent links, s and t share their BCs, adding them in oBC_s and oBC_t respectively. This link is intended to be partial because it is not affected by the sorting steps, but they have the highest priority in the forwarding.

Differently from "complete" connections, a partial link lasts for the time of a session, and they are not opened as a device restarts. A partial connection has a probability to become complete that is updated every time the *discovered* node t confirms that the link is useful - the forward of the request has t as target -. This probability is updated by applying Equation 3.11 seen in Section 3.2.3.

By just opening connections, the network risks being highly connected. This is not always a desirable situation, because a lot of (useless) connections overloads the network. It is manageable in a smart house scenario, where a controlled number of devices is expected, but is not always reasonable in a smart museum. The best candidates to be erased are old connections, preferring explored connections instead of designed ones. Although strategies to limit connections have been investigated, the system currently keeps old connections as well.

3.4 Implementation

This Section reports relevant details about the implementation of PHASER. It is organised as follows: Section 3.4.1 presents the architecture of PHASER and how it set the network up. Section 3.4.2 will detail possible states for a PHASER node and their meaning. Sections 3.4.3 and 3.4.4 will talk about the chosen Interaction protocol - as structure of each message - and the adopted communication protocol with a comparison of well-known protocols for IoT.

3.4.1 Architecture

The model presented in Section 3.2.1 has been developed as a framework and a preliminary release of this design has been presented in [96].

The architecture represents the skeleton of a single PHASER node, that must be able to *(i)* interact with users, *(ii)* communicate with other similar nodes in the environment and *(iii)* locally process received commands. It has been designed as connected modules, depicted in Figure 3.5. An example of a network populated of PHASER nodes is, instead, in Figure 3.6.

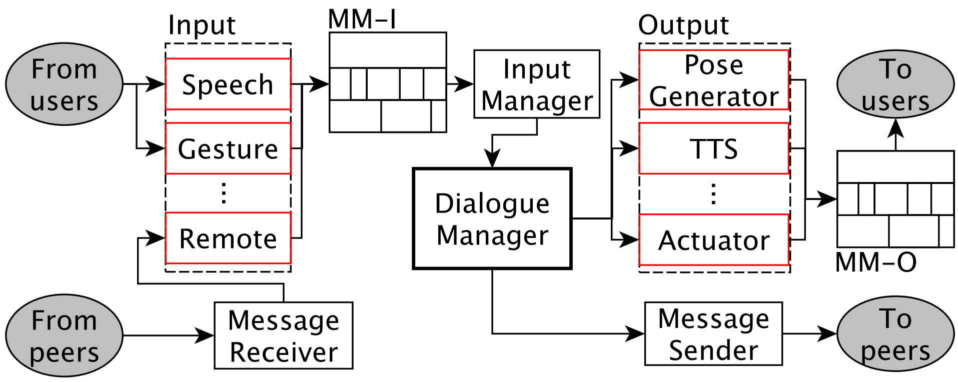


Figure 3.5: The architecture of a single PHASER node

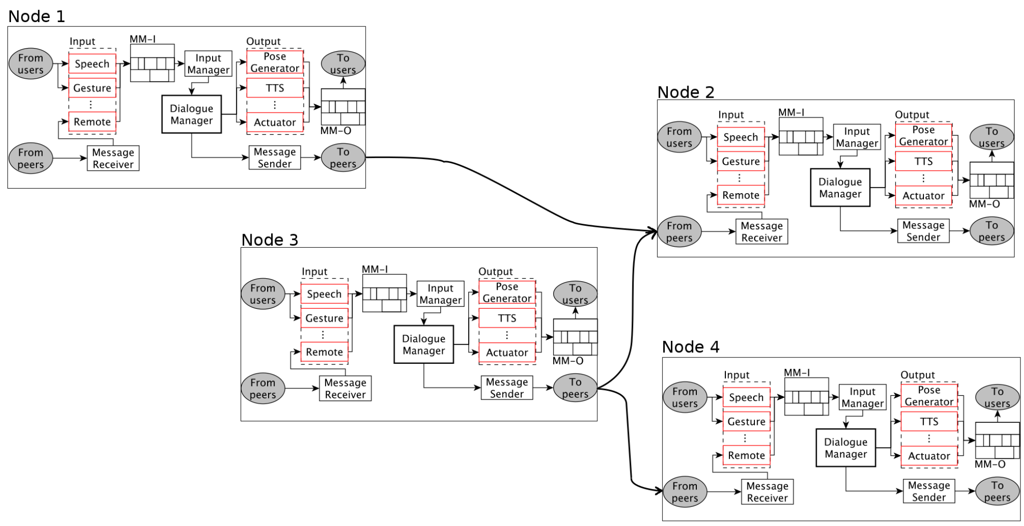


Figure 3.6: An example of PHASER network. It is composed by linked architectures

PHASER interacts with users through *Input Devices* (IDs). They are modular blocks that independently manage sensors. The same happens for *Output devices* (OMs) used to present feed-backs to the user. IDs and OMs implement an interface, in order to abstract from the used technology and represent data in a hardware-independent form. This is useful because in a ubiquitous system *"the way the system outputs information to the user and the user provides input to the system should not be fixed"* [97]. This is useful in the concept of natural user interfaces [21] and makes *Context-Sensitive I/O* possible [98].

The diagram in Figure 3.5 shows examples of IDs and ODs. Their actual activation depends on configuration each node requires and formally presented in Section 3.2.1. As IDs (OMs) produce an input (output) they specify starting and finishing timestamps. Since all the modules run on the same machine, a single clock is used and inputs (outputs) are synchronised. Dedicated structures - *MM-I* and *MM-O* - store synchronised inputs (outputs) writing them on different channels. This way, PHASER supports multi-modal dialogues. These channels usually require a semantic level fusion, as stated in similar studies [99].

An *Input Manager* (IM) manages the fusion of data taken by the MM-I structure and passes their classification to the *Dialogue Manager* (DM). The IM is an interface so it can be personalised as needed, and supports fusion at *decision* level. DM is, here, just an interface towards a real dialogue manager; its behaviour, in fact, highly depends on the particular node and cannot be included in the overall description. The DM adopted in this PhD is mainly based on OpenDial [100], included as an external tool, but other DMs may be integrated in PHASER. However, generic interaction strategies are adopted, aiming at improving the quality of a Dialogue Manager. The DM in PHASER is *user-directed*, so based on the initiative of the user. This tends to generate speech recognition and understanding errors [101], but the designer has a support to limit them.

The real DM processes the request and returns to the framework the output - that will be presented to the user - or a code in case of failure. The failure code activates internal algorithms to manage the Navigation Problem. It automatically sorts the known peers and forwards the request as explained in Section 3.3.1.

The *Remote* module is a particular ID used to communicate with non-human peers by standard protocols. This ID creates a connection from other peers. They can be robots, smart-devices, technologically enriched works of art, etc. Relationships among these entities create a PHASER network in which each node has an internal logic. By including the *Remote* module as ID, PHASER can equally interact with humans - through active IDs/ODs - and artificial entities - through *Remote* -. This aspect improves the user-centred point of view, because connections are hidden and users perceive the world as a single block, where parts of it process each request.

With the proposed architecture, PHASER offers a ubiquitous infrastructure and a comfortable framework for an Internet Of Things (IoT). The powerful aspect of this architecture is that its overall behaviour is not related to a single entity nor it is domain-specific but, with proper I/O devices and DM, it allows to easily prepare an Intelligent Environment, concentrating efforts on each entity. Furthermore, if an environment is considered as “entities providing services” and by sharing the Business Cards, each system will be able to opportunely contact nodes to solve internal tasks. This is a typical concept in AI agent-based approaches, but I propose it in a multi-domain - interaction-oriented - abstract architecture.

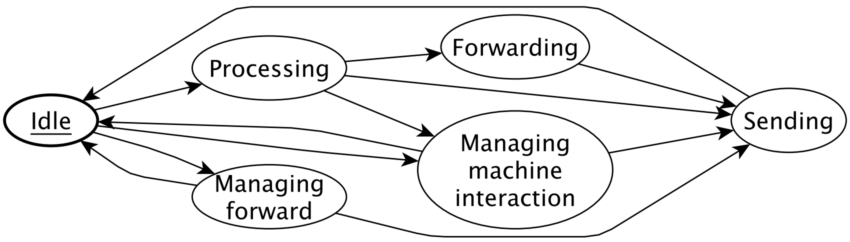


FIGURE 3.7: Possible states and Transitions for a node in PHASER

3.4.2 States

Each node in PHASER has a state that determines which kind of work a node is doing and if it can accept other requests. A node starts in the *Idle* state; Table 3.1 summarises the states, while Figure 3.7 reports all the allowed transitions. *Forwarding* and *Managing forward* are states adopted when nodes are processing the Navigation Problem.

Managing machine interaction (MMI) state has been introduced to exchange messages with other peers - other than forward - and it is useful if nodes collaborate with each other to respond to a request. The connection *MMI* $\rightarrow$ *Idle* is a "forced" transition and it happens if node ι is waiting for node κ for a *MMI*. If ι remains in *MMI* it would not be able to receive a response from κ, because it would be recognised as busy. When in *Idle*, the node is able again to accept new requests. The code of the received response will help the current node to understand which was the proper previous state.

3.4.3 Interaction protocol

A PHASER node shares messages through the network using a JSON[2] message, composed of two entries: a code that represent the type of request and a EMMA XML[3] node with the request. EMMA is designed to annotate multi-modal inputs; alternative XML-based standards exist, for example M3L [102], MMIL [103].

[2]http://www.json.org retrieved on September 2017
[3]http://www.w3.org/TR/emma/ retrieved on September 2017

State	Description
Idle	initial state
Processing	the node is processing a request
Forwarding	the node is forwarding an unknown request
Managing Forward	the node is processing a received forward
Sending	the node is replying or re-forwarding a request
Managing machine interaction	two nodes are co-working for a request

TABLE 3.1: Description of states of PHASER

Possible codes for messages are: *question*, *answer*, *[no_response]* and *forward*. The first two codes are used in interaction that involves humans: if PHASER runs on a smartphone, for example, the person could talk with it and the smartphone shares the request through the network. As the smartphone is very general purpose, its Dialogue Manager simply acts as an interface between the user and the network. It sends questions or simple messages. It changes the behaviour of the Dialogue Manager because in the former case it waits for a response, in the latter it does not. If a node is not able to process a request and needs to share it to others, it will choose *forward* code to forward a request. The node will receive *[no_response]* if an explored peer is not able to process the request itself.

3.4.4 Communication protocol

PHASER currently uses Websockets as communication protocol. Websockets are TCP-based and, differently from standard sockets, they support full-duplex communication with the possibility of sending and receive messages at the same time [104]. Although a quick interaction/communication is important in PHASER and TCP has a higher latency compared with UDP, TCP has been preferred because it is reliable and this redounds in a better efficiency in the system [105].

I investigated other standards usually adopted in IoT. MQTT (Message Queuing Telemetry Transport)[4] and CoAP (Constrained Application Protocol) [106] are well-known application-level communication protocols based on the *"publish-subscribe"* architecture [107] where sensors publish data and a broker updates clients waiting for new resources. The main difference between MQTT and CoAP is that MQTT relies on TCP and CoAP on UDP.

MQTT and CoAP adopt different strategies to ensure Quality of Service, but being based on a publish-subscribe architecture makes these protocols unsuitable to PHASER: each node should be independent and a message broker - that accepts and dispatches messages - limits the scalability. The considered protocols, indeed, have been designed for a different configuration, where Wireless Sensor Networks (WSNs) generate messages and clients receive updates [108], implicitly requiring different roles.

Another communication protocol I analysed is AMQP (Advanced Message Queuing Protocol) [109] that still supports the "publish-subscribe" approach, but defines other routing features like "point-to-point" communication. AMQP seemed to be a valid alternative to adopted Websockets and will be extensively investigated for future integration.

[4] http://mqtt.org/ retrieved on August 2017

3.5 Summary

In this Section I proposed PHASER, a framework that represents an entity able to interact with users. Each entity can establish connections towards other similar nodes, realising a network; peers are independent from the others, and they can independently interact with people, so it is a distributed model. With the discussed techniques, the envisaged work-flow of the interaction is *user* $\Rightarrow$ *node* $\Rightarrow$ *network* $\Rightarrow$ *user* . Currently, virtual assistants like Siri or Google Now propose online a query to online search engines if they are not able to understand the request. Although this approach is intentionally discarded in PHASER, a "commercial" version of this system could insert this step to avoid negative responses from the network.

The innovative aspect faced with PHASER is that each node forwards requests if it is not able to produce the expected output. The forward is based on a request-level decision process and it does not require to specify the "target" node *a priori*. The navigation of the request operates among nodes that are not able to accept it, but they all use a strategy to deliver the request to a right node.

PHASER does not aim at being a model useful in all the contexts, but there are open issues that PHASER solves. Although they concern smart houses, a comparison with smart museums would be a simple task. These issues are presented below in a scenario-based description; scenarios have been introduced in the '90s as part of User-Centred Design. Scenarios help in filling the gap from research to design. They compose the first of the following three steps: scenarios, requirements, interaction framework; all these steps are glued with narrative. Scenarios are commonly used to *describe a method of design problem solving by concretisation: making use of a specific story to both construct and illustrate design solutions* [18, Chap. 6].

Scenario 1

> *John is in the living room, watching TV is on and he is reading a book. Meanwhile, his wife is finishing the housekeeping, interacting with the washing machine and using the radio and their son is in the bedroom playing the guitar. The oven is cooking the dinner and John needs to check it, so he asks the TV by voice if the oven is still working. The TV checks and answers "no, it has finished", showing a small message in the corner. Then the family can have dinner. Later on, it's time to go to bed, so John and his wife set the alarm and fall asleep; the child does the same. The alarms can look the people in the house and can monitor the quality of their sleeping. As it is almost time to wake-up, the alarms send a message; the thermostat catches*

> *it and sets the right temperature in each bedroom. John asks the alarm to prepare a coffee, then he takes the breakfast and goes to work*

This scenario shows daily activities that can be supported by technology. Three people populate the house. They are a family and, although with different tasks, they find moment to stay all together. Since they are in different parts of the house, networked devices provide a ubiquitous support to the users, delivering commands and controlling the house from each place. In this case, PHASER-based devices independently interact with people and a distributed architecture helps in having a modular system.

Scenario 2

> *It is Sunday and Mark is taking a shower; he will reach some friends later. He wants to mow the lawn, with the just bought automatic lawnmower, but it has not been programmed to start yet and, at least on the first time, he does not want to leave it working alone. So Mark asks some device there to start the lawnmower. The lawnmower starts*

In this scenario an already existing network of smart devices is changed because the actor, Mark, bought a new device. The introduction of the new device usually requires a re-configuration of the whole network or, at least, an information exchanging with a master node in the network. With PHASER, instead, if a connection with the new device exists, all the nodes in the network are immediately able to deliver requests to the new node; the result is that the user is able to interact with it from any device, without requiring a centralised process.

PHASER aims at underlying of existing solutions and devices that are focused on specific contexts. It would be the framework to create connections and exchange messages where an Intelligent Environment is possible. A motivational background has been found in the Or.C.He.S.T.R.A. project [110], where many solutions have been proposed for Cultural Heritage. Details about related applications can be found in Appendix A.

Chapter 4

Request analysis in PHASER

Chapter 3 presented PHASER, a distributed model where independent nodes support Human-Computer Interaction in Intelligent Environments relying on partial knowledge about the context. Although people are not directly involved in most of the discussed processes - i.e. *"Navigation problem"* or the network topology - they alter these steps because the whole system aims at tailoring the PHASER network on the inferred user needs; these needs actually come from interactions with people.

As a fundamental concept, each PHASER node is focused on its own context; it is not required for an entity to understand all the networked requests. However, a rough and generic analysis of a received command can improve its navigation with low partial knowledge. This step is useful both *locally* to better sort peers in forwarding the request, and *globally* to avoid issues in the whole environment. This chapter presents some improvements in requests' analysis. It is organised in two blocks; in Section 4.1 I will propose linguistic analyses to finer calculate question matches and better sort close peers. This section improves the *"Navigation problem"* seen in Section 3.3.1. Section 4.2, instead, is based on an abstract request analysis, by introducing essential elements in the ontology to face *conflicts resolution* in a PHASER environment.

4.1 Linguistic analysis for request understanding

Each PHASER node is able to receive requests from users; however these devices are expected to have a certain amount of knowledge, circumscribed to their own domain: a fridge, for example, should understand questions about food or ingredients; commands about lighting and heating are *out-of-context* and forwarded to other peers. It means that each entity may manage commands for any other node in the network. Nodes have

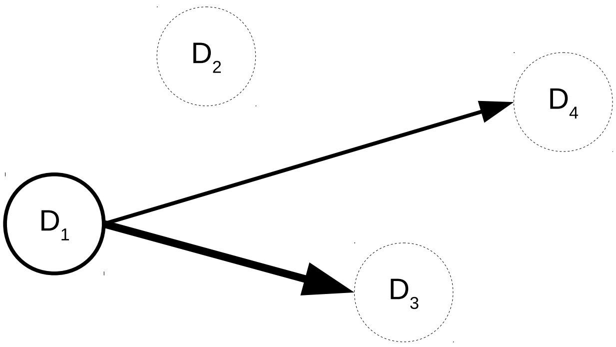


FIGURE 4.1: D_1 sorts $D_{2,3,4}$ providing a measure related to the probability to accept the received request. The thickness of each arc is proportional to this probability. D_3 will be firstly selected

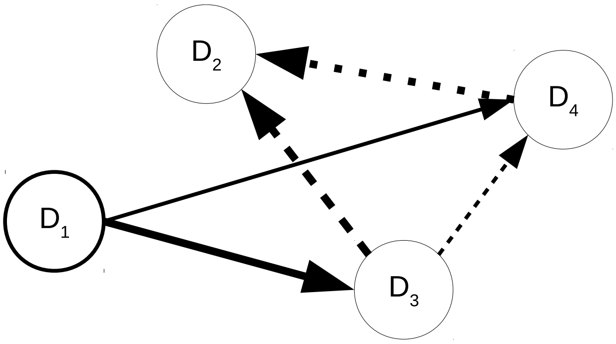


FIGURE 4.2: An example of the navigation. A command from D_1 reaches D_2 via D_3 or D_4. The thickness of each arc is the obtained match percentage of that request for the reached node; the dashed line is related to the length of the path. The algorithm must prefer shortest paths

two possible solutions with non-matching requests: *(i)* broadcast them to all connected nodes or *(ii)* just filter the most probable ones. The two approaches highly affect the network overload and should carefully adopted.

As extensively described in Section 3.3.1, PHASER nodes forward requests which were not resolvable given the local knowledge, to the best candidate with a *depth-first-search* algorithm by iterating on a sorted list of close nodes. The navigation continues until a node finds a response or all the sub-network is explored. Figure 4.1 depicts a graphical representation of the sorting phase that is performed in the navigation presented in Figure 4.2. D_1 chooses the next node in the interaction by sorting the adjacent vertices.

After a brief literature review, this Section proposes a recall of the initially adopted technique for the *"Navigation problem"* based on perfect matching and it continues with further investigated approaches that do not have this constraint, finer sorting the adjacent nodes.

4.1.1 Background

The approaches that will be discussed in this Chapter can be grouped in two parts: full match - with a perfect match on regular expressions - and partial match - based again on regular expressions or *bag-of-words* -. Although they are not new approaches in literature, I did not find any relevant work used in a context similar to PHASER's one.

The goal is to provide a strategy to better rank close nodes according to the exposed information about the accepted inputs. The *Ranking problem* is one of the fundamental problems in Information Retrieval (IR) [111], where search engines return results for a query. The Ranking problem can be solved with machine learning as summarised by Liu in her tutorial [112]. However, although the navigation problem can be seen as a *query-based* ranking problem, there is a fundamental difference. Each PHASER node relies on a small information shared with close peers. They compose the documents for IR, but they are usually composed by a few words - compared with the amount of information each document usually carries on the web -. Then, the forwarding involves many independent peers and the work each node does is limited to some operations. In IR, instead, the work is centralised on a server that collects, compares and ranks all the available documents.

By considering the Navigation problem from a Question/Answering (Q/A) point of view, PHASER could be theoretically compared with distributed Q/A systems [113]. Q/A systems do not collect entire documents, but they extract just short and relevant information to produce an answer. Since the documents are not all physically stored on the same server, a distributed Q/A system deals with parallel tasks and load balancing. Even if some similarities with PHASER can be considered, the main difference is that a node ends its own work as it delivers the message.

By adopting strategies well-known in literature, the added value of PHASER is in its application. Each node independently uses the considered approaches - that will be presented and compared in the rest of this Section - but the outcome is a finer sorting in a distributed navigation where other context-related information tailor the forwarding on observed interactions.

4.1.2 Full matching

A network of PHASER nodes starts from a topology designed by experts of the considered domain. During the interaction, the network adapts its own connections to maximise the local utilities on each arc. As the network sets up, each pair of connected nodes shares a business card. It comprises a set of active channels and, for each channel, a set of regular expressions (regexp) for the accepted inputs. By splitting the input into multiple channels, PHASER is able to support multimodal signals.

The *"Navigation problem"* is solved by sorting the adjacent nodes according their matching with the exposed regular expressions. Nodes with higher value of matching will be firstly called in forwarding. Inputs can be on multiple channels, so the matching complies with the structure. With a perfect matching, M calculates the value of matching as I have already defined in Equations 3.19 and 3.17. I re-propose them here:

$$M(R, n) = \sum_{0 \leq i < |R|} m(R_i, n)/|R|$$

$$m\left(R_x, n\right) = \begin{cases} 1 & \text{if } R_x \text{ is a valid input for } n \\ 0 & otherwise \end{cases}$$

where n is the considered device, R is the request, divided into $|R|$ channels. The expression "R_x is a valid input for n" means that exists a regular expression of n that matches with R_x. The higher is M, the more probable is that n can understand R. $M(R, n) = 1$ means a perfect match.

4.1.3 Partial matching

The approach seen in Section 4.1.2 is based on perfect matching where the outcome of each $m(x, n)$ is dichotomous. The computed value M is then normalised to the size of R - meaning the involved channels in R -. This approach highly depends on how accurate is the design of the set of regular expressions for each channel. Moreover, a generic regexp - such as *".*"* - accepts a large number of inputs. This case is not preferred: if a node accepts this input it will attract all the requests with the highly probable consequence of not being able to process all of them, acting as a *black hole*.

Another approach, supporting partial matching and based on the linguistic analysis of the received question, would be more flexible because it provides a confidence of the input. The improvement must still prefer a perfect matching, but it does not completely exclude the opposite case. In order to do that, I propose revised formulae, introducing $m_{l_x}^{v}$ as the confidence of v on channel x, and adapting M as follows:

$$M(R, n) = \prod_{0 < x \leq |R|} max\left\{\left(m_{i_1}^{R_x}, \ldots, m_{i_n}^{R_x}\right)\right\} \tag{4.1}$$

The function in Equation 4.1 still considers multiple channels and a set of possible grammars for each of them, but $m^v_{l_x}$ is now the confidence of the token v from the request on an input l_x. This is actually a probability. Two strategies have been compared to find the best choice that gathers it: regex-based and *bag-of-words*. The former approach calculates the longest sub-string that matches on each provided regular expression - filtering the channels -. The obtained length is then normalised on the total length of the request. The latter method, instead, splits both the request and the stored accepted inputs in bags of words - B_{req} and B_{input} respectively - and calculates how many words of the request match on the total set. This value is then normalised on $|B_{input}|$. Both the introduced strategies are locally calculated by each node for any received question that needs to be forwarded. No global dictionaries are saved in order to maintain a scalable distributed system where each node has partial knowledge about the environment. B_{input} can be obtained off-line, limiting required operations at run-time. A comparison of the two approaches will be given in Section 4.3.1.

This process tends to locally improve PHASER nodes, because it increments the precision of the sorting procedure.

4.1.4 Is Navigation Useful?

The *"Navigation Problem"* starts after a local miss, where the node that has received the request is not able to understand it, or to produce a useful output. Requests with a 0 match are typically forwarded by PHASER. However, non-sense phrases, out-of-domain requests and questions specifically directed to the current node - but that cannot be temporarily processed - fall in this class.

The navigation through the network could be useless in this case, because they are hardly understandable by all the nodes in the environment; the navigation would occupy nodes to process a request, and they would not produce any output. So, an additional step before the forwarding may detect these messages and avoiding the routing.

Following the approach of the linguistic analysis, I propose here a domain-independent strategy to detect if the navigation would be useful or not. It uses the following steps:

1. "neighbourhood check":
   (a) locally checks if at least one $\kappa \in ClosePeers \cup DiscoveredPeers$ could accept the received question;
   (b) In case of fail, the node checks if one of the neighbours is explicitly named in some of the inputs, by searching their name or class in the request;

2. "global check": in case of fail of previous steps, the node searches if some input contains the name of some available nodes in the adopted ontology. This step is performed by invoking the Ontology Manager.

With the presented algorithm, the current node will possibly not forward requests with spelling mistakes or out-of-domain requests. In a museum scenario, for example, user's utterances containing *"friggitrice"* (Italian word for "the deep frier"), could be considered out-of-domain, but its meaning is different if that word is specified in the ontology as part of the name of an element (*"La friggitrice"* is a painting of Diego Velázquez as well). So, according to information coming from the adopted ontology, the behaviour of the system is different. This check globally improves a PHASER network because it avoids clearly useless forwarding.

With premises introduced in this Section, the interaction between user and nodes can be tailored on the available features. In case of fail in forward, for example, the error message delivered to the user should exhort her to better format the request, providing additional information to identify the target node.

As stated in the last point, the check proposed here requires to invoke the Ontology Manager to list all the available nodes with a match on the name. For this reason this step can be excluded by the designer.

4.2 Conflict Resolution

Since a user that interacts with PHASER is waiting for a response, the navigation of the request needs to rapidly produce an output. If many people independently interact with devices in a network, then conflicting requests may arise [114]. In this scenario, conflicting situations need to be detected and avoided or resolved as they arrive.

In order to detect and resolve conflicts, the approach I propose here is based on a hybrid solution of reaction and proactivity; a PHASER node firstly reacts as receives a question and proactively invokes other nodes to avoid the conflict. This strategy is known to be one of the best working techniques [115].

In PHASER, the configuration each device has, specifies accepted requests and involved *services* for each of them. In particular, provided and influenced services may appear, according to the common ontology. The ontology collects a representation of services and features related to them. As an example, the environment may process the command *"cool the room down"* in multiple ways; with a smart-fan, cooling may be slower, with less energy absorption and may influence noise. An air-conditioning, instead, would be faster but it takes more energy and should not be directed on a person. In this example, both the devices provide *cooling* with different values of *energy absorption*, but the fan influences *noise*, while *air-conditioning* do not. As that request arrives, PHASER should deliver it to the best candidate.

At start-up each agent registers itself to the Ontology Manager (OM), exposing internal parameters: connection information and provided services and features. OM stores the registration and returns a value for each presented element. A list or registered nodes is essential for conflict resolution. OM is required to act as a *yellow-page* provider as I will deeper explain later. OM should not be intended as a Service Oriented Architecture (SOA), where a single server provides services at requests [116]. It just collects available services and their hierarchy.

Level	*Description*
internal	at least two requests delivered to a single device
request	at least two nodes can understand and process the request
service	the request operates on services and its processing may generate inconsistent situations

Table 4.1: Chosen levels for conflicts

Each actuator on a pervasive system may affect its surrounding physical environment, thus influencing the context. Although a formal definition of *context* does not exist, a commonly accepted description declaims it as *"any information that can be used to characterise the situation of an entity [...] relevant to the interaction between a user and an application, including the user and the application themselves"* [94].

The definition of conflict varies from context-aware application to application [117]. Tuttlies defines conflicts with respect to a user or application as: *"[...] a context change that leads to a state of the environment which is considered inadmissible by the application or user"* [118]. In a distributed system populated of independent entities (i.e., PHASER), the probability of a conflict is higher than in other systems, mainly due to a number of contexts and services used, and the mobility of entities [115].

PHASER adheres to the Tuttlies definition of conflict but, since the initiative of the interaction is of the user, I identify in each *command* the essential element to be analysed. Two commands may conflict with each other in multiple ways, summarised as follows:

many requests for a single device: two or more conflicting requests on the same device;

single request for many devices: the same command can be processed by many nodes in the environment;

many requests for many devices: conflicts that are not generated by the command, but by its application.

that can be classified in Table 4.1; other taxonomy of conflicts can be found in literature [114].

The first considered level is not actually a conflict: in PHASER, each node solves *internal-level* conflicts according local strategies and a complete model cannot be provided for each of them. So, *internal-level* conflicts have been introduced but PHASER detects and has instruments to solve them. Other cases need to be discovered and resolved.

The first node that receives the request and is able to process it is elected as *mediator* (hereafter *Med*); Med *(i)* detects the conflict, *(ii)* gathers the state of the world to resolve the issue, *(iii)* delivers the request to the winner. This approach maintains a distributed model and avoids overloading the same system for every possible conflict, although some steps are temporary centralised on the Med. The proposed approach for conflict resolution need to:

- prefer the mediator if it is the best candidate;
- be sure to avoid inconsistent situations within the environment;
- involve useful nodes in the process.

With these premises, the mediator is a good candidate to resolve the conflict because *(i)* it is able to understand the request, *(ii)* the shortest path reached it as first, *(iii)* it considers close agents for questions and *(iv)* it is able to compare utilities.

The mediator follows steps in collaboration with other agents and the Ontology Manager; they are explained in the following Sections.

4.2.1 Background

Conflict resolution is adopted in many topics. It is common in Multi-Agent Systems, where agents cooperate applying negotiation techniques to reach a common goal [119, 120]. Jacak and Pröll [121] presented a heuristic approach aiming at coordinate agents and find the right sequence of agents' local goals to achieve a common objective. A robotic system was discussed as a case study.

Perumal et al. [122] presented a rule-based framework to resolve conflicts. It was based on the *event-condition-action* paradigm and, through a centralised system, aimed at avoiding the generation of conflicts. An architecture based on multi-users and their own preferences, has been introduced in [49], where a virtual assistant helped people in managing daily activities and solve conflicts in a smart house. The interaction was personalised for different categories of users. A similar context has been discussed in [123].

In their work, Retkowitz et al. [124] proposed a system which assigned a service to a received resource management and used predefined dependencies of services to find the best candidate. The process was not fully automatic and based on priorities. A comparable configuration has been presented by Huerta-Canepa et al. [125], where a list of predefined metrics avoided conflicts; considered measures were users' priorities and jobs' running time.

Other studies about consistent context management have been investigated by Guerrero-Contreras et al. [126]. The authors proposed a system for consistency management in a ubiquitous environment where resources were distributed and replicated. As they claimed, a pure SOA architecture is not powerful enough for a ubiquitous system, because a SOA systems centralise requests' processes. Moreover, they are not flexible to cope with dynamic network topology [127] and rapid context changes. The proposed approach was based on a SOA system combined with replication techniques to maintain consistent replicas of the context. This issue is stronger in a mobile environment [128].

Conflict detection and management can be conducted at different levels. The works presented above mainly consider request- and service-level but higher orders are possible. Masoumzadeh et al. [129], for example, refer to authorisation conflicts, where two or more policies both permit and forbid access. However, since policies are statically defined, this kind of conflicts can be detected off-line; resolution still remains a run-time issue. Eventually, Santos et al. [130] presents a survey about normative-level conflict resolutions.

The main differences of the listed systems with my approach is that each PHASER node, as a request arrives, can become the mediator - who detects and resolves the conflict - without overloading the dedicated peers in the network. Starting from requests, PHASER intentionally involves services to better detect and avoid conflicts. This is essential for an effective conflict resolution [131].

Security issues may arise in the considered configuration, because malicious devices may expose misleading values and alter the overall process; the confidence is limited because the mediator collects bids from involved peers, but it locally applies costs before starting a quick negotiation. As revealed in Section 3.1.2, conflict resolution requires an Ontology Manager acting as a yellow-pager provider.

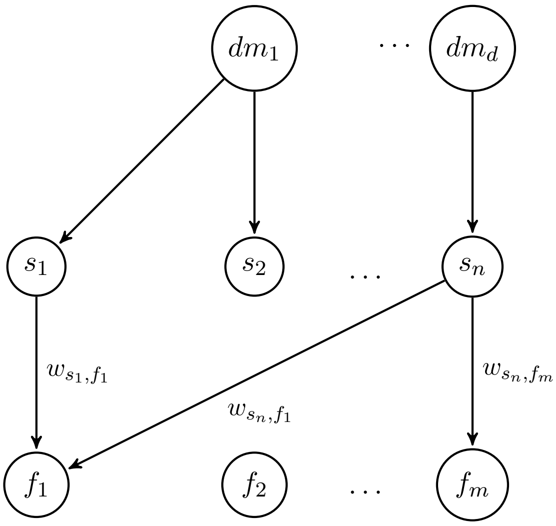


FIGURE 4.3: Description of Domains, Services and Features used for Conflict resolution

4.2.2 Model

In order to resolve conflicts in the proposed distributed environment, the Ontology Manager represents a hierarchy of elements that will be used in the process. The organisation is proposed in Figure 4.3. It is formally defined as the following tuple:

$$CR = \langle DM, S, F, DS, SF, W_S, T, SoV \rangle \tag{4.2}$$

where

$$DM = \{dm_1, dm_2, \dots, dm_d\} \tag{4.3}$$

$$S = \{s_1, s_2, \dots, s_n\} \tag{4.4}$$

$$F = \{f_1, f_2, \dots, f_m\} \tag{4.5}$$

stores all the available Domains, Services and Features respectively.

$$DS = \{(ds_{1_1}, ds_{2_1}), \dots, (ds_{1_x}, ds_{2_x})\} \tag{4.6}$$

$$SF = \{(sf_{1_1}, sf_{2_1}), \dots, (sf_{1_k}, sf_{2_k})\} \tag{4.7}$$

$$W_S = \{w_{s_1}, \dots, w_{s_n}\} \tag{4.8}$$

Equation 4.6, with $ds_{1_i} \in DM \wedge ds_{2_i} \in S$, $\forall i \in [1, 2, \dots, x]$, collects edges between domains and services. Equation 4.7, with $sf_{1_i} \in S \wedge sf_{2_i} \in F$, $\forall i \in [1, 2, \dots, k]$, stores, instead, edges between services and features. Equation 4.8 stores weights w_{s_k} of service nodes s_k, $\forall s_k \in S$. Weights denote priorities among services; they have not a meaning alone, but they gain relevance used together.

Domains used in conflict resolution are different from domains of the PHASER ontology, presented in Section 3.1, where a representation of the world was provided. However, sets of all the possible devices D and requests $\overline{R}$ are used for conflict resolution as well.

Each feature in the extended ontology has a *"Type"* and an admissible *"Sequence of Values"* (SoV). Types are in set T and they determine the length of the sequence as follows:

$$T = \{B, N, SR, MR\} \tag{4.9}$$

$$SoV_T = (v_1, v_2, \dots v_n) \tag{4.10}$$

representing the type of feature and a set of possible values. We considered four types:

Binary (B): $n = 2$, features value v can be dichotomous ($v = v_1 \oplus v = v_2$);

N-values (N): $n \in \mathbb{N}$, features value is in a set;

Single range (SR): $n = 2$, features value v in a range: $v_1 \leq v \leq v_2$;

Multiple ranges (MR): $n \in \mathbb{N}$, features value v: $v_{2i-1} \leq v \leq v_{2i}$ with $1 \leq i \leq \frac{n}{2}$.

Types and sequences of values for each feature are stored in the set SoV introduced in Equation 4.2, defined as:

$$SoV = \{(T_k, SoV_{T_k}) \mid k \in F, T_k \in T\} \tag{4.11}$$

and each SoV_{T_x} complies with Equation 4.10. For an actual use, their value is provided by experts.

At run-time a final value for a feature can be defined according to the single taken value in the valid sequence. This final value is computed by the following function:

$$v_k : SoV_{T_k} \rightarrow [0, \dots, 1] \tag{4.12}$$

defined as:

$$v_k(a) = \begin{cases} j/n & \text{if } T_k \in \{B, N\} \wedge \exists j \in [1, \dots, n] \mid SoV_{T_{k_j}} = a \\ \frac{2i(a - v_{2i-1})}{n(v_{2i} - v_{2i-1})} & \text{if } T_k \notin \{B, N\} \wedge \exists i \in [1, \dots, \frac{n}{2}] \mid v_{2i-1} \leq a \leq v_{2i} \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

where $n = |SoV_{T_k}|$.

A service has a value that can be either related to a connected feature or the weight of the service itself. It is computed by the following function:

$$V_s : S \times F \times SoV_T \to [0, \dots, 1] \tag{4.14}$$

defined as:

$$V_s(s_i, f_k, val) = \begin{cases} v_{f_k}(val) & \text{if } (s_i, f_k) \in SF \\ w_{s_i} & \text{otherwise} \end{cases} \tag{4.15}$$

Since both $v_{f_k}(val), w_{s_i} \geq 0$, then $V_s \geq 0$.

At registration time, each node specifies services and features. The OM calculates corresponding values through Equation 4.15 and returns them to nodes.

Formally, the node n exposes:

$$E_n = \left((e_1, e_2, e_3)_1, (e_1, e_2, e_3)_2, \dots, (e_1, e_2, e_3)_x \right) \tag{4.16}$$

where $p_j = (e_1, e_2, e_3)_j \in S \times F \times SoV_{T_{e_2}}$, if $p_j \in SF$, $(e_1, \emptyset, \emptyset)_j$, with $e_1 \in S$, otherwise.

OM returns the same sequence, changing each $(e_1, e_2, e_3)_j$ with $(e_1, e_2, V_s(e_1, e_2, e_3))_j$.

4.2.3 Discovery

The first required step for conflict resolution is *discovery*. A user generates a request R. The request reaches the mediator (Med) - the first device that is able to process that command -. As a first step, Med checks with known agents who is able to process the same request.

Given the request $R \in \overline{R}$, the mediator creates the set X_0 defined as follows:

$$X_0 = \{x \mid x \in ClosePeers_{med} \cup DiscoveredPeers_{med} \text{ and } M(R, x) > 0\} \tag{4.17}$$

where $ClosePeers$ and $DiscoveredPeers$ are sets of known and discovered peers respectively defined in Section 3.2.1; $M(R, x)$ is the matching function initially introduced in Equation 3.19. An improved version has been proposed in this Chapter.

Since the mediator is able to understand the request, a service s_y may be associated to the request itself. Other services may be influenced. A request does not must to provide services: requests for checking or informative questions do not affect the environment and do not actually provide a service.

An internal configuration specifies provided and influenced services and the registration phase gathered all the possible values. All the influenced services are returned by the function:

$$InfS : D \times R \to \mathcal{P}(S) \tag{4.18}$$

If s_y exists, the mediator invokes OM to gather the set $Prov_y \subseteq D$ with all the devices that, with some accepted input, may provide s_y. The mediator composes X_1, X_2 and S_X as follows:

$$X_1 = X_0 \cup Prov_y \tag{4.19}$$

$$X_2 = \{x \mid x \in X_1 \text{ and } M(R, x) > 0\} \tag{4.20}$$

$$SX = \{x \mid x \in X_1 \text{ and } x \text{ is active on } s_y\} \tag{4.21}$$

where X_2 and SX are completed by opening a connection with each node in X_1 and asking them if they can understand R or they are currently providing s_y. This step is necessary because of the distributed model without a central agent that maintains the state of the world. The Ontology Manager is able to provide a set of available devices, but it does not care about their status. Another important aspect is that $X_0 \cap X_1 \neq \emptyset$ because a node $x \in X_0$ understands the request but it can provide $s_k \neq s_y$. As an example, this is possible if the request is *"set the temperature at 25 degrees"*. A heating and a air-conditioning may both accept the command but, if the current temperature is higher, the former device may provide *"heating"* anyway, the latter would provide *"cooling"*. At discovery time both the devices will participate at the conflict management. Eventually the heating will not win.

At this step, $\forall x \in X_1$, the mediator also composes

$$InfS_x = InfS(x, R) \tag{4.22}$$

with all the influenced services for R. Then, OM is invoked again to gather $Prov_k$, $\forall k \in \bigcup_{x \in X_2} InfS_x$ and the mediator composes *items*, defined in the following Equation:

$$items = \{(d, s) \mid d \in D \text{ provides } s \in InfS_x,\ x \in X_2\} \tag{4.23}$$

Eventually, the mediator loops on each $(d, s) \in items$ to check if they are active on the specified influenced service. By all the positive responses, the mediator builds:

$$Active = \{(d, s) \mid (d, s) \in items \text{ and } d \text{ is active } s\} \tag{4.24}$$

At the end of the discovery phase, all the considered conflict levels are detected as:

single request for many devices: X_2 collects these nodes;

many requests for many devices: $Active \cup SX$ stores all the influenced nodes - through provided or influenced services -.

During the discovery phase, the mediator invokes twice OM to gather essential information to detect a conflict. By performing rules on services, the Ontology Manager checks that the provided services do not generate domain inconsistencies. In case of violated domain rules, the returned value is negative.

Domain-level conflicts may be generated if policies limit services' activation or devices operating on one or more services. In PHASER they are not completely formalised, but the Ontology Manager maintains a control on them. However, the conflict resolution considers this possibility for future improvements.

Domain-level checks are *(i)* at the beginning, so that the whole discovery process would not start in case of negative response on the provided service and *(ii)* at the end to be sure that the influenced services as well would not create other issues.

4.2.4 Resolution

The resolution phase starts if at least one conflict has been detected in the discovery. This step has the final goal to deliver the received request to the first winner that is able to process the request itself. The mediator is a valid candidate. During the discovery, OM has been invoked twice to *(i)* retrieve agents on services and *(ii)* to check domain constraints.

If $X_2 \neq \emptyset \wedge Active \cup SX \neq \emptyset$, at least two agents are generating a conflict. By collecting SX, X_2 and $Active$ the mediator retrieved a snapshot of the state of the world; in order to resolve the conflict the mediator assigns a bid - composed by utility and costs - at each node to choose a winner. However, the state of the world is not frozen, so the winner could be unable to start as it receives the command.

In order to calculate a bid, each node x *(i)* involved in a conflict resolution, *(ii)* able to understand the request R and *(iii)* that provides s_y in processing it, calculates the utility $U_{s_y}(E_x)$ by using the following function:

$$U_{s_y}(E_x) = \sum_{(s_y, e_2, e_3) \in E} e_3 \tag{4.25}$$

where E_x has been defined in Equation 4.16. Values are positively defined, so $U_{s_y} \geq 0$.

The final bid is obtained as follows:

$$B_x = U_{s_y}(E_x) - C_x \tag{4.26}$$

where C_x, not formally defined here, is a term foreseen by the node itself to save internal resources. C_x is not exposed to other peers. The mediator collects each bids by the candidates, and ranks them to choose a winner.

In the final ranking, showed in Equation 4.27, the mediator involves the received bid, a cost of the influenced services and terms to prefer *reliable* nodes and prefer less used nodes. Each node stores, for each connection towards a peer x, a success rate r_x to represent its reliability, and an age a_t updated to the last used timestamp.

$$rank_x = B_x - CI_x + \mu_r r_x - \mu_a \left(a_x - \overline{a_t}\right) \tag{4.27}$$

$$rank_{mediator} = B_{mediator} - CI_{mediator} \tag{4.28}$$

where $0 \leq \mu_r, \overline{a_t}, \mu_a \leq 1$ are constants empirically defined. By tuning them, the ranking will prefer reliable and young nodes. The former term is introduced to foster connections that successfully worked in past interactions; the latter is for load balancing.

CI_x is the cost coming from the influenced services:

$$CI_x = \sum_{i \in InfS_x} f_i \cdot d(i, Active) \tag{4.29}$$

with $d(i, Active)$ is:

$$d(i, Active) = \begin{cases} 1 & \text{if } \exists (d, i) \in Active \\ 0 & \text{otherwise} \end{cases} \tag{4.30}$$

With an approach similar to the Navigation, seen in Section 3.3.1, the final rank proposes:

$$Sorted = \left(p_1, p_2, \cdots, p_{|X_2|+1}\right) \tag{4.31}$$

with $rank_i \geq rank_{i+1}, \forall 1 \leq i \leq |X_2|-1$. The mediator is included in $Sorted$ in the right position, so $|Sorted| = |X_2| + 1$. p_1 is the winner. The resolution ends by forwarding the received request to each p_i, starting from p_1. Since the state of the world is not frozen, the processing of the delivered command may fail; for this reason, the forward continues with p_{i+1} if p_i did not successfully process the request. The question cannot be forwarded anymore. The loop terminates on the *mediator* because it is still a valid node to process the request.

The approach I proposed in this Section is an application of the *one-shot* negotiation mechanism. Each peer exposes their bids and the mediator chooses a winner; bids are hidden for all but not the mediator. I preferred this approach, avoiding longer negotiations, because the conflict resolution runs with a user waiting for a response. The presented steps ensure a consistent management of the resources and reach a final decision in a reasonable time. In Section 4.3.2 I will discuss performed experiments.

4.3 Results and Discussions

This Section proposes experiments conducted for the arguments discussed in this Chapter. A smart house with multiple devices has been considered. The first part presents results related to the linguistic analysis; the second part, instead, discusses the introduced conflict resolution strategy.

4.3.1 Linguistic analysis

The system presented in Section 4.1 has been evaluated by comparing three approaches: perfect matching, partial matching, and bag-of-words (BoW). Perfect and partial matching methods rely on regular expressions and offer a measure of how much the request matches the provided regexs. The system has been tested by simulating a smart house with 5 networked devices. The considered nodes are:

TV: (TV) provides entertainment services;

Microwave Oven: (M) is able to cook food and to expose its own cooking state;

Fridge: (F) lists its content and recommends recipes;

Kettle: (K) can prepare a tea or a coffee;

Alarm: (A) sets an alarm or wakes the user up.

I considered a star-like network with TV in the middle; requests start from the TV itself. The network configuration is adopted as an example; the centralisation on TV simulates a scenario where the user prefers to control devices from a fixed position. I

command	*perfect*	*partial*	*bag-of-words*
prepare a tea	K (1.0)	K (1.0)	K (1.0)
warm	**A (0.1)**	M (0.44)	M (0.5)
warm water	**A (0.1)**	**A (0.1)**	K (0.667)
wake me	**A (0.1)**	A (0.438)	A (0.5)

TABLE 4.2: Winner device and confidence for each request. Each node had a bag-of-word style inputs. Bold cells refer to unsuccessful evaluations; the forward in this case is random

command	*perfect*	*partial*	*bag-of-words*
prepare a tea	K (1.0)	K (1.0)	**K (0.1)**
warm	**A (0.1)**	M (0.44)	K (0.33)
warm water	**A (0.1)**	A (0.9)	K (0.667)
wake me	**A (0.1)**	A (0.778)	A (0.4)

TABLE 4.3: Winner device and confidence for each request. Each node had a regex style inputs. Bold cells refer to unsuccessful evaluations; the forward in this case is random

tested two kinds of configurations for input representation: in the former case each input was represented as a BoW style: *"give me with recipe recipes"* and *"me recipes with give recipe"* have the same value. Table 4.2 reports the gathered results. In the latter case, instead, inputs were formatted as regular expressions - i.e. *"give me recipe(s)? with .* please"* -. Table 4.3 summarises the collected data. Each node firstly sort peers with the compared strategies; then used OpenDial [132] to manage a dialogue.

Results show that a *full matching* is not always a good choice, because this strategy does not always discriminate different nodes with a consequent wrong ranking. Moreover, it requires a precise design of each regular expression and this information is exposed to other nodes: remote peers know the accepted input and its structure. *Partial matching* provides finer values and nodes can be better sorted. This approach, however, easily creates *black hole* nodes that accept many inputs because of too generic regexs. The bag-of-words model gave the best results. In addition, this solution has two benefits: *(i)* designing nodes with bag-of-words is easier; *(ii)* each node could share unstructured data, improving local security.

Other strategies have been investigated. I considered more refined approaches based on SRGS[1] (Speech Recognition Grammar Specification); however, this method requires a deep knowledge opened with adjacent nodes - in order to reach a precise evaluation each node must share all the local grammars to close nodes - and security issues may raise from this. For this reason, SRGS has been excluded. Locally each node could adopt

[1]https://www.w3.org/TR/speech-grammar/ retrieved on July 2017

grammars to test and/or to categorise received commands, but the test just concerns information exposed to close peers.

4.3.2 Conflict resolution

In this Section I describe experiments conducted to test the proposed conflict resolution method. A smart house with 5 devices, organised in 3 rooms, has been simulated. All the processes run on the same machine, in order to exclude additional delays related to network transmission. The algorithm has been tested with many topologies in order to evaluate differences in terms of quality and required time. Adopted topologies are:

star-like: the mediator is in the centre of a star-like network;

linear: the network is a chain. The mediator starts the chain; the target ends it;

layered graph: the network is layered, composing a hierarchical structure. The mediator is in the first level, while the target in lower levels;

disconnected node: at least one node is not reachable through a direct link;

random topology: connections do not follow a designed structure.

In order to validate the system, I conducted tests to evaluate the completeness and correctness of the approach. Completeness ensures that all the necessary nodes are involved in the process. Correctness, instead, assesses if the involved nodes behave proper, choosing as winner the best node.

The system has been tested by delivering at nodes in the network a question. The starting node changes in order to monitor all the nodes in the network. In the experiment I measure the following values:

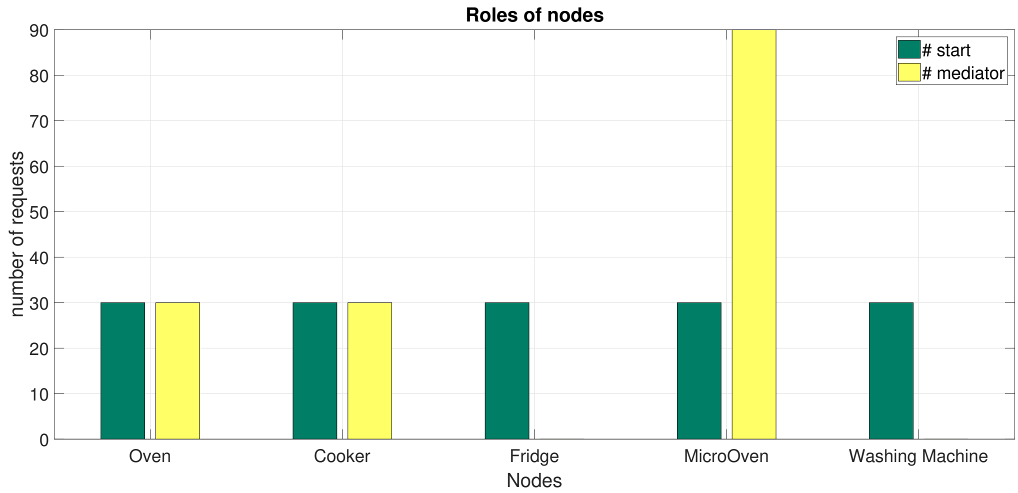
time for interaction: total time required from the delivery of the request to the presentation of the response;

time for discovery: time required for the discovery of conflicts;

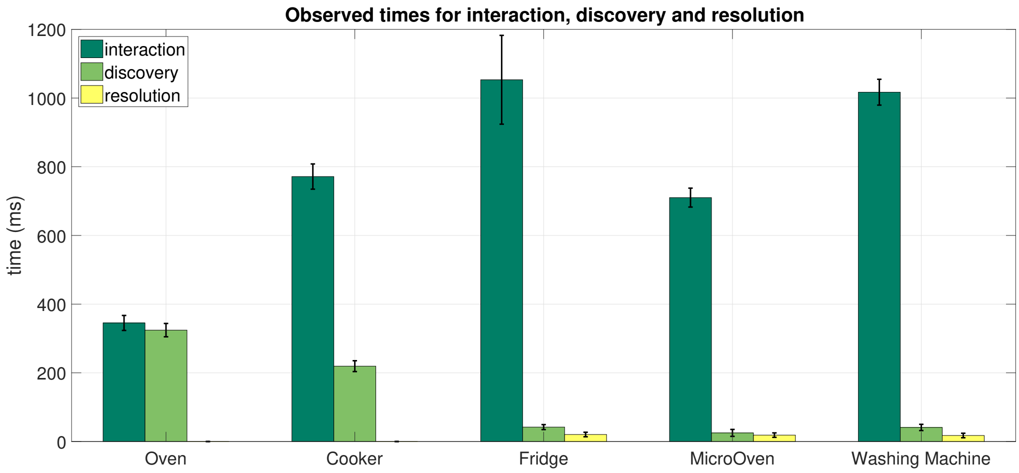
time for resolution: time required for the resolution of conflicts;

nodes for discovery: number of involved nodes. They are divided in:

- close nodes which understand the request;
- nodes that provide the same service;
- nodes that could be providing one of the influenced services;



(a) Summary of activity of each node



(b) Observed times (in milliseconds) of interaction, discovery and resolution

FIGURE 4.4: Global results observed with a star-like network. The plot summarises different configurations that alter the network behaviour

***start-mediator-winner*:** paths analysis and detection of the three main nodes.

Each request has been delivered 30 times for each node. As an example, Figure 4.4 reports conflict resolution results for a star-like network. A *Microwave Oven* is in the centre of the network and the request - *"cook food"* - may be accepted by *Oven*, *Cooker* and *Microwave Oven* itself. Both the request and the nodes' design are intentionally generic in order to generate conflicts. Figure 4.4(a) reports how nodes respond in summary. Since the *Microwave* is in the centre of the network, and it is elected many times as mediator. *Oven* and *Cooker* are mediators as they receive the request. The network was configured to let *Oven* to win the conflict. Details about correctness of the approach will be discussed later in this Section.

Figure 4.4(b) depicts observed times for interaction, discovery and resolution. Times are in milliseconds and graphs are ordered for starting nodes. As the Figure shows, discovery and resolution are fast operations. Interaction is higher but still acceptable. In particular, I observed that interaction is longer when the starting node - the node which interacts with the user - delivers the request to the mediator and waits for responses. This happens with *Fridge* and *Washing Machine*. The interaction is longer because, if the mediator does not win the conflict, the winner node will repeat discovery and resolution. This is necessary as the state of the world may change and inconsistent situations may produce undesired consequences. *Cooker* and *MicroOven* reported lower interaction time because they are mediators, so the network registers conflict resolution on just 2 active nodes - mediator and winner -.

Oven gave the lowest results because, as it receives the request from the user - starting the interaction - it is both mediator and winner, as designed for the test. For this reason, interaction and discovery+resolution times are comparable.

In order to assess correctness and completeness, the algorithm has been tested on a different setup. As a completeness evaluation, the goal is to investigate that all the relevant nodes are involved in the conflict resolution, without invoking useless peers. The system is also correct if the best node wins the conflict. The graph is composed by 10 nodes, where all of them are disconnected. The nodes seen above - *Oven*, *Cooker*, *Microwave Oven*, *Fridge* and *Washing Machine* - are part of the considered network, while the others are configured and fill the network. They appear but *(i)* do not understand the request, *(ii)* do not provide the *cooking* service, and *(iii)* do not operate on the influenced services. This means that just the listed nodes should be involved in the conflict resolution. All the nodes are configured to let *Oven* be the best node. The system has been run 5 times for *Oven*, *Cooker* and *Microwave Oven*. The same request delivered the others - in a disconnected graph - would not produce any response because the starting node is not able to assess neither the request nor the provided service with that. After delivering a request, I observed that a path among these nodes exists, while all the others are still disconnected. This demonstrates that all the relevant nodes are involved, while the others are excluded. Table 4.4 reports results about the experiment in the disconnected network.

The same experiment has been executed on 5 networks by generating random topologies among 10 nodes, with a resulting connected graph. This experiment aims to show that the topology does not affect the result. The request has been delivered to all the nodes 5 times; 3 of them - *Oven*, *Cooker* and *Microwave Oven* - have been the mediators. They have been reached with different paths according to the current topology, but all the requests have been solved. In all the cases, *Oven* - which is the best node - won the

Start	**Bid**	**Check**
Oven	Cooker Microwave Oven	Fridge Washing Machine
Cooker	Oven Microwave Oven	Fridge Washing Machine
Microwave Oven	Oven Cooker	Fridge Washing Machine

Table 4.4: Completeness assessment in a disconnected graph. The request has been delivered to nodes that *started* the interaction, first column. They become mediators as well. The second column reports nodes asked to provide their bid. The last column reports actors involved in activity checks

conflict resolution, while *Fridge* and *Washing Machine* have been invoked for activity checks.

The last two tests demonstrate that the approach is correct because the winner node is always the same, configured to be the best node. The system is complete as well, because all the relevant nodes, with different roles, are invoked in the process. Topology seems to not affect the outcome. However, if the request starts from a disconnected node that is not able to understand neither the request nor the provided service, the PHASER network will not produce a valid output. Analyses on spent times showed that all the interactions with conflict resolutions require up to 1000 ms, in the worst case.

4.4 Summary

This chapter proposed some improvements introduced in PHASER on different aspects. They both act on the analysis of the request: the former procedure classifies a received command with a linguistic process to finer sort adjacent nodes in a PHASER network without understanding the considered request. In the latter approach, instead, a conflict resolution strategy has been proposed to make a PHASER environment more robust and reliable. By processing a request, a node may provide a service, while others can be influenced. Preliminary checks ensure that actions taken by agents in the environment do not create inconsistent situations.

The improved request analysis globally improves the behaviour of a PHASER node because it enhances the precision of internal mechanisms. While the adopted *bag-of-words* approach would be useful in many contexts, the check on the usefulness of the navigation can be limited in few cases. In a museum, for example, a PHASER network would be easily overloaded, so additional checks may help in limiting useless requests' navigation.

Conflict resolution implicates an additional overhead in PHASER because many communication steps are needed. A bottleneck is the invocation of the Ontology Manager (OM) with a double goal: *(i)* as a yellow-pages provider, OM returns information about peers operating on a service; *(ii)* OM checks *domain-level* conflicts analysing the activation of services. The latter feature has not been formalised in PHASER, but it will be considered as future investigations. However, the conflict resolution is not always a *"necessary evil"*, because it is just useful where parallel updates to shared resources are expected. While it happens in smart houses, a smart museum is usually populated with devices to provide information, without actually update global resources.

Chapter 5

Setting PHASER up

The design of an Intelligent Environment requires many steps and involves in the whole process people with different roles. In this thesis I presented a model that represents a single interactive entity in a ecosystem of similar objects. All of them have some strategies to collaborate with each other and support people and their needs. In Chapter 3 I explained the core model of PHASER; in Chapter 4 additional details about request handling have been provided. This Chapter moves the focus on professionals that operate in the considered environments. They need to setup a PHASER network but a technical background usually misses. I propose here a platform including many solutions to support users that will use PHASER with different roles.

Although the process presented in this Chapter is domain-independent, some closer looks on the considered environments will be showed. After an extended discussion about the involved actors in Section 5.1, three parts will be discussed: in Section 5.3 I will present a tool supporting the job of an expert. An Interaction Designer (IxD) specifies the behaviour of each PHASER node with a proposal in Section 5.4. The designed environment is then simulated and tested by an Architect in Section 5.5.

Some of the information I will give in this Chapter are still conceptual and not fully included in the currently adopted release of PHASER. I will specify the prototypes parts.

5.1 Actors involved in the process

The whole process is composed by the initial design and realisation and continues with the monitoring of a particular snapshot of the environment. With PHASER I identified three actors involved in the setup. An *expert* of the considered domain is the general director of the whole process. The expert:

1. chooses elements that will populate the environment;
2. virtually places the smart objects on a map;
3. sets up each of them and links semantically connected entities.

The resulting graph will compose the initial topology of the PHASER network - details in Section 3.3 -. However, an expert usually has no technical expertise on interactive devices, so her work should be graphically supported as much as possible.

By following guidelines of the expert, an *Interaction Designer* defines the behaviour of each placed device, so she composes the intelligence users will perceive. It specifies input and output modules as well; this step determines how each entity responds to users. The IxD carries an essential work out in PHASER: a wrong design would harm the final perceived intelligence and interaction capabilities. Natural User Interfaces [21] must be considered here. The role of each PHASER device is defined at this step, and how it will interact with people must be determined.

The design process ends with an *Architect* that, mainly oriented to the outfitting of the physical environment, collaborates with the expert for the scenic design. The architect needs a more precise visualisation and 3D models' renderings with an accurate details management. The architect is intended as a bridge between expert's and IxD's needs; she ensures that environment changes - in terms of lighting and similar parameters - will not be damaged by using all the PHASER network together. If two or more nodes affect the lighting, by using a projector, for example, the IxD is focused on each single device, while the architect ensure that their projected areas does not overlap and that the environment lighting does not ruin the interaction.

The categorisation above should be intended as a general suggestion, but it does not mean that the different roles have to be filled by different people in any context. In a smart house, for example, the final user may act as expert because a context already exists and she may want to preserve privacy. In addition she would just need to connect devices to the already existing network. The same for architect: a PHASER node does not alter the home outfitting, so a real architect could be a useless investment. Nevertheless, in a smart house the interaction designer is who produces the device. In this case, multiple manufactures could use the PHASER framework so they do not have a single expert as reference. In this case, different IxD are just focused on a single node, working in a modelled context that simulate possible environments. This is usually done with commercial products. The approach is completely different in a museum. In that case the expert is a curator preparing a new museum or a temporary exhibit. Many objects of interest will interact with visitors and their behaviour must be tailored on the final goal - related to the *"Big Idea"* of the museum itself -. An intense communication between curator and IxD is important in this configuration. The outfitting is essential as well. Also in this part, a collaboration between expert and architect is typical.

5.2 Environment Design

Involved actors have different backgrounds and needs. IxDs provide world representation by modelling processes and users. Experts, instead, usually provide domain knowledge, referring to non-technical contexts. For this reason, the approach I propose here aims at filling this gap and providing a support for both the actors: it does not limit experts creativity but it translates their output in a structured view, essential for interaction designers. With a closer look on museums, one of the goals I have in designing a PHASER environment is the translation from a "conversational" description of visitors in a structured formalisation of their model. This is a general concept that is behind the motivation of this Chapter.

5.2.1 Categories of visitors

This Section analyse categories of people that are assumed to populate the considered environments to interact with a PHASER network. A structured representation of users is a needed starting point. However, while in museums field many studies provided categories of visitors, the same support from houses, suitable for the scopes of this thesis, has not been found in literature; it seems, indeed, that a common division misses, tailoring descriptions to validate works or, on the other side, following commonly accepted groups and roles.

In the museum field, Falk and Dierking [133] classified individuals in the following seven classes:

Explorers: curiosity-driven visitors. They have generic interest in the content of the museum.

Facilitators: socially motivated individuals. Their visit is mainly oriented to accompanying social groups.

Professionals/Hobbyists: their passion or job is closely related to museum context. They usually focus on a specific topic or content.

Experience Seekers: they consider museums as an important destination. They are motivated and their satisfaction derives from going there.

Rechargers: they look for a contemplative and/or restorative experience. They usually go to museums to take shelter from the chaos of the city.

Respectful Pilgrims: they think that a visit to the institution/memorial should be done to honour the memory of those represented there.

Affinity Seekers: individuals motivated to visit a particular museum/exhibition because it speaks to their sense of heritage and/or being.

This well-known categorisation helps in distinguishing people that decide to visit a museum. The last two, *"respectful pilgrims"* and *"affinity seekers"* are not actually fully researched categories [133, Chap. 2] but they fill gaps in label visitors motivated by national or ethnic/racial reasons.

5.2.2 Visiting styles

Different people may behave in different ways in a museum. While Falk and Dierking categorised visitors in terms of motivations, Veron and Levasseur [134] proposed four visiting styles in museums. They labelled visits with a biological inspiration: *ant*, *grasshopper*, *butterfly* and *fish*. These classes differentiate according the percentage of visited POIs and time spent towards them. Ants carefully explore POIs, seeking a lot of information from each of them. Grasshoppers seems to have a specific interest for some pre-selected exhibits and spend a lot of time observing them while they ignore other parts. Butterflies do not follow a specific path; they stop frequently to look for more details. Eventually Fishes do not show particular interest and they usually move in the centre of the room avoiding looking at POIs' details. Although other categorisation of single visitors can be found in literature [135, 136], this work is based on the Veron and Levasseur styles. A pioneering study has been argued by Kuflik and Dim [137] that proposed social behaviour patterns of pairs of visitors.

Visitors' behaviour is object of interest since ever and technological solutions have been adopted in many studies to analyse visits in museums [83, 138]. One of the most difficult goals is to define visiting styles in a non-intrusive manner [139] to propose personalised contents [68, 140] in an interactive way [70]. This aspect is essential because it may increase people visiting museums [133, Chap. 2].

Being inspired from literature, a PHASER network can be tested with models of visitors with realistic motivations. A simulation may help in monitoring *(i)* how virtual devices respond to users and *(ii)* how requests navigate in a virtual representation of the final environment. Virtual users help in simulating motivations and providing relevant inputs

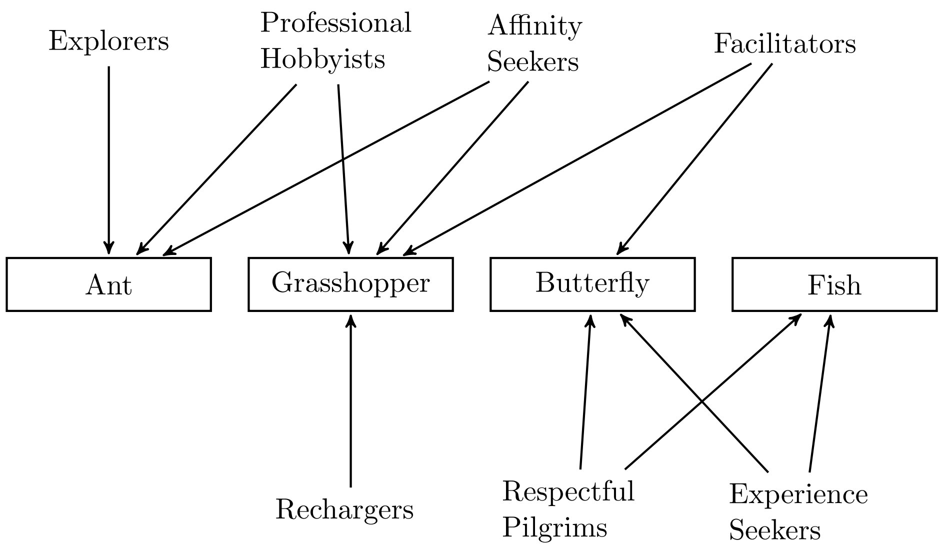


FIGURE 5.1: The proposed mapping among *(i)* Falk and Dierking - items without borders - and *(ii)* Veron and Levasseur - items with border - categories. Visitors' distinction based on motivation - *(i)* - are mapped on their expected visit strategy - *(ii)* -

for PHASER nodes. Such a simulation does not require financial investments and supplies a graphical representation of the designed world. The virtual system will be deeply discussed in Section 5.5; it provides a support for all the actors: an IxD may monitor devices behaviours; the architect focuses on the virtual outfitting and how simulation may affect it; the expert maintains a direction on it.

Although a proved correlation between motivations and visiting styles does not exist, exhibit features as well as real events stimulate interest for various groups of people that behave differently. As an example, a permanent museum may attract explorers or rechargers, that are highly motivated people and usually consider museums for leisure. Visitors change with promotions or free tickets: facilitators - e.g. families - and experience seekers may take advantage from that. Eventually, a temporary installation may attract professionals/hobbyists. These aspects are objects of investigation since many years and have been confirmed in more recent studies [133, Chap. 2]. Starting from this rough hypothesis, in this Chapter I propose a mapping from visitors' motivations to visiting styles. A validated mapping does not exist in literature and this thesis does not have this goal; in PHASER a mapping would just provide a way of communication among actors. A wrong mapping would *not* affect a PHASER outcome, but it aims at being a strategy to choose visiting styles in a more realistic than a "random" way. Figure 5.1 a mapping between Falk & Dierking and Veron & Levasseur categorisations. It is used as follows:

1. experts select - rough - estimated percentages of visitors' categories. They would

suggest expected people by their motivations, so the Falk and Dierking classification suits well. Different percentages may suggest the reason of the test - stable museum, temporary exhibition, etc. -;

2. experts' input is mapped on the Veron and Levasseur partition as shown;

3. the simulation - actually performed by the IxD - uses step 2 as input to determine types and numbers of the visiting styles. Details about simulation can be found in Section 5.5.

If IxD well designed PHASER nodes' behaviour, a wrong mapping and experts' input would not actually compromise the work-flow. Nevertheless, by tuning the discussed parameters and changing visitors' flow, a more realistic scenario can be simulated.

5.2.3 A structured view

A translated structured counterpart of the cited categories can be used modelling them. A great example are *"Personas"* [141]. The *persona* is a multipurpose design tool that helps face several problems in designing digital products. Personas are based on real-world observations, to better classify people that will use a product. A deep analysis about possible users, their needs and ethnographic information are essential for a complete modelling; ethnographic data are the key difference between personas and typical user profiles.

Personas resolve three design issues that arise during design development:

- the elastic user;
- self-referential design;
- edge cases.

the elastic user issue arises when a precise description of "user" misses; the "user" becomes what the designer needs at different steps with an imprecise consequence. The self-referential design happens if the designer consider herself as the main - and often the only one - user of the system. Eventually, the edge cases are possible, but remote, cases the system should manage: they have to be considered but they are not the focus of the design.

In PHASER, the design of the environment uses a step of simulation, especially proposed in the museum domain. It will be discussed in Section 5.5. By finding a correlation between categories of visitors and categories of their own behaviours in museums, personas may help in creating a bridge between the expert - that provides a distinction - and the IxD - that adapts models to realistic cases -. Personas are particularly useful for this scenario because they *"[. . .] must have motivations"*[18, p. 83] and, in the beginning of this Section, I listed visitors categorised by their own motivations. In addition, *"[. . .] personas are allowed to be successful because they are "personifications"[142] [. . .]"*[18, p. 81].

However, in order to indicate how devices respond to virtual users, simulation of strategies of visit can be a good approach to explore how visitors reach POIs. Personas change their way of looking for details or way of delivering requests. In Section 5.5.2 I will propose my strategy to simulate the exploration.

5.3 What a domain expert can do

The domain expert plays a fundamental work in PHASER. It defines entities composing the final PHASER network and indicates the initial topology, by connecting semantically related nodes. According the considered environment, the expert has different backgrounds: in a smart museum, for example, the expert is a curator or a museum professional in general. PHASER is going to be used to create a temporary exhibition or it is already running in a permanent museum. In this case, PHASER should not limit the creativity. In a smart house, instead, the expert intervenes in different moments of PHASER life because she may connect a new device to an existing network, for example. The introduction may require simple operations. In this case the expert is not required to have a specific background.

In this Section I will propose a tool an expert could use to set a PHASER environment up. It is oriented to people that are not familiar with technological details, but aims at creating a context populated of interacting devices. The tool is not focused on a particular context, but it often refers to smart museums, where the expert plays a more influential work.

5.3.1 What experts need

"Museum curators, exhibition designers and other professionals are increasingly involved in the design of exhibits that make use of interactive digital technologies to engage visitors in novel ways"[143]. However, although interactive exhibitions aim at enhancing visitors' experience, they are object of interest for curators as well, that consider to adopt technological solutions in their exhibitions. In both the cases, the goal is to provide

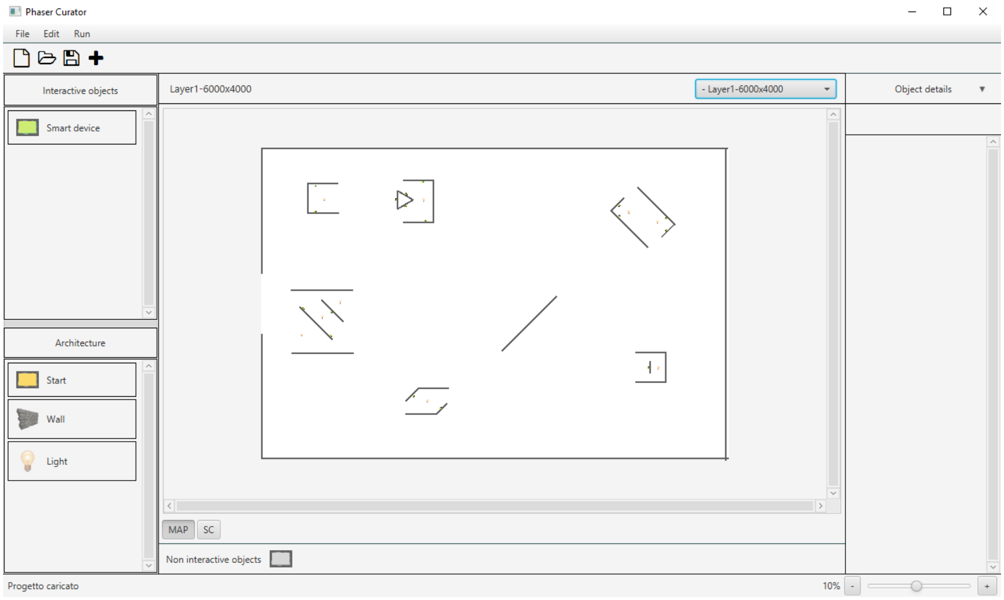


FIGURE 5.2: An example of the general design of the environment. The expert decides the involved PHASER entities, their connection and some rough physical constraints

personalised contents to visitors. While people exposed their interest in living tailored experiences [140, 144], personalised content are a desire for museum professionals that want to encourage people in repeating the visit.

Maye et al. [143] interviewed cultural heritage professionals to understand what they expect from interactive exhibits and how technology could support their work. Discussions reveal that there is not a unified opinion about technology but they all agree that focused interventions may increase interest towards museums. Some experts claim that technological solutions should not become barriers because focus should remain on the object and not on the means. This has been also discussed by Cutugno et al. [145]. Other curators have contrasting points of view: some of them would include technology just to attract people; others would actively adopt them to provide innovative ways for exploring and deepening contents. Regardless the adoption of technological expedients *during* the visit, all the professionals agree that tools may help their work. Also on this aspect needs are diverse: Content Management Systems for digitally managing of material, cataloguing, photogalleries, etc.

Since PHASER provides a technological support, experts setting a PHASER environment up should accept technology in their own museums. Differently from "typical" devices, PHASER nodes explicitly declare connections; the expert should be able to indicate both the behaviour each device expose and entities it is connected to.

Element organisation

The GUI (Graphical User Interface) of the tool is depicted in Figure 5.2. As the Figure shows, the user places elements on a region organised in levels and rooms. They compose the organisation of the environment. Elements are partitioned three categories: interactive, architecture and non-interactive items.

Interactive items compose the PHASER network. They are entities users can interact with; they build the intelligence people will perceive. Experts spatially place and link interactive items. At this level, connections just refer to a semantic level; they will compose the initial topology discussed in Section 3.3. *Architecture items* consist of walls, pillars and other architectural items. They fill the environment or complete an existing infrastructure to guide flows of visitors. *Non-interactive items* include all the items that are not provided in the previous groups. Furniture items, for example, belong to this area.

The presented categorisation is not related to a single context. Although this Section has many references to the museum context, the tool I present is not restricted to this field. In a smart house, for example, the architecture section will be less used, but interactive and non-interactive parts would have a larger assortment.

Object details

Through a side panel in the GUI - on the right side in Figure 5.2 - the expert defines parameters for the selected item. For interactive items and, in particular PHASER items, parameters are related to the configuration presented in Section 3.2.2. Most of them must be specified by the expert because they related to the domain: class and environment are example of them. However, in the current version all of them are set by the expert. Figure 5.3 depicts an example of that.

5.3.2 Monitoring

The presented tool provides a support to place and link items on a scene. Elements are divided in *architecture*, *interactive* and *non-interactive* items. Interactive items, PHASER nodes in particular, may change the initial topology during the interaction. A new link may bring additional information to the expert: a connection $A \rightarrow B$ means that a number of people required the interaction from A to B. Since the new connection will be re-used in the future but it may change the semantic of the museum. In this undesired situation, the expert may act on the museum itself, removing the link. In

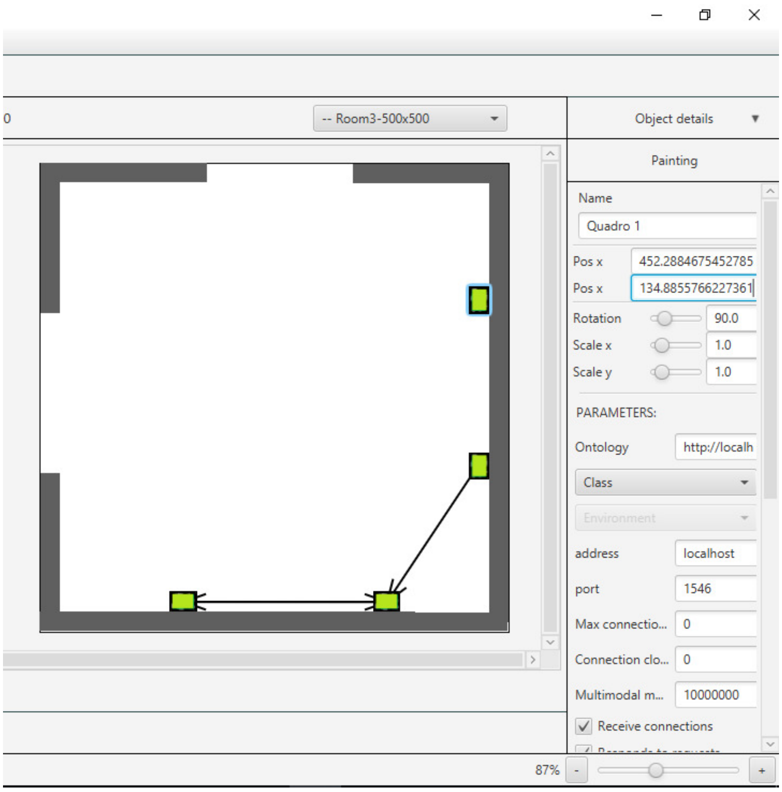


FIGURE 5.3: Details section for an interactive item

some cases, the can also decide to modify the architecture - by adding a physical barrier - to prevent the reachability of B from A. The tool provides a monitoring feature, that shows to the expert updates in the designed topology. The expert receives updates about connections so she is able to take decisions on that.

Existing IoT infrastructures already have this feature and this confirms the adoption in this tool. Examples can be found in [87, 146].

5.3.3 An XML for environment representation

The environment designed by the tool presented in Section 5.3 is then used by an Interaction Designer. In order to do that the environment is exported according a standard representation; an XML-based format has been designed. It is presented in this Section. The organisation is split in two parts: each node has its own configuration, partially designed by the expert; all the composition designed through the toll is represented in a different file.

World

This section reports details about the XML-based representation of the world designed through the tool presented in this Chapter. This part reports the environment organisation in levels and rooms and the class of the elements placed on each room. Rooms and items have scale, rotation and relative position as well. The following Listing reports an example.

```
<document>
  <layer id="1" name="Level1-1000x1000">
    <image>null</image>
    <width>1000.0</width> <height>1000.0</height>
  </layer>
  <room id="2" name="Stanza1-500x500">
    <image>null</image>
    <width>500.0</width> <height>500.0</height>
    <positionx>2.50</positionx> <positiony>5.01</positiony>
    <rotation>0.0</rotation>
    <scalex>1.0</scalex> <scaley>1.0</scaley>
  </room>
  <interactive-item id="4" class="Cooker" configuration="Conf\cooker.xml">
    <name>cooker</name>
    <positionx>56.49</positionx> <positiony>371.42</positiony>
    <width>10.0</width> <height>11.0</height> <depth>12.0</depth>
  </interactive-item>
  <architecture-item id="7" class="WallItem">
    <positionx>444.41</positionx> <positiony>59.35</positiony>
    <rotation>90.0</rotation>
    <width>400.0</width> <height>20.0</height> <depth>300.0</depth>
  </architecture-item>
  <layer-association>
    <pair source="1" target="2"/>
  </layer-association>
  <room-association>
    <pair source="3" target="4"/>
    <pair source="3" target="7"/>
  </room-association>
</document>
```

Single node configuration

Each interactive item has an external configuration. This separation is needed because independent nodes may run on different devices. Each device has its own configuration file. It is composed by settings and values specified through the tool. An example is in the following Listing. It defines a PHASER node with parameters specified in Chapter 3. However, as introduced in the preamble of this Chapter, the IxD is not currently fully supported, so part of the definition assigned to the expert will be moved to the IxD.

```xml
<configuration>
  <domain
      taxonomy="" ontology="http://localhost:12345/domain"
      remote-controller="http://localhost:12345/update">
    <node name="cooker" class="Cooker" environment="Kitchen"/>
  </domain>
  <network>
    <ip-addr value="localhost"/>
    <port-no value="1543"/>
    <max-connection-time value="0"/>
    <connection-closing-period value="0"/>
    <tolls limit="0.5" mu="0.1" entry-value="0.3">
    </tolls>
  </network>
  <behaviour>
    <setup
      multimodal-max-history="10000000" receive-connections="true"
      response-to-requests="true" save-history="false"
      require-history="false" interaction-with-followers="false"
      can-be-part-of-groups="false" receive-group-id-requests="false"
      follower-accepts-members="true" receive-group-subscription="true"
      receive-group-requests="true" forward-request="true"/>
    <connections>
      <connection
        id="con0" ip-addr="localhost" port-no="1547" required="true"
        required-at-interaction-design="true" last-operation-ts="0"
        num-successes="0" num-attempts="0"/>
    </connections>
    <inputs>
      <device id="0" mode="command-line" medium="u_u" active="true">
        <data-format expr="cook food" idx="0" />
        <service key="0" value="cooking" />
        <data-format expr="is water boiling" idx="1" />
      </device>
    </inputs>
    <outputs>
      <device id="0" mode="text" medium="u_m" active="true"/>
    </outputs>
    <dialog-manager domain-file="dummy"/>
  </behaviour>
</configuration>
```

5.4 Interaction Design

In PHASER, an Interaction Designer defines the behaviour of each interactive item specified in the XML seen in Section 5.3.3. A possible way to do it recurs to statecharts.

Statecharts have been initially proposed by Harel to formalise complex systems [147] and it was possible to reduce them to finite-state automata [148]. In subsequent studies

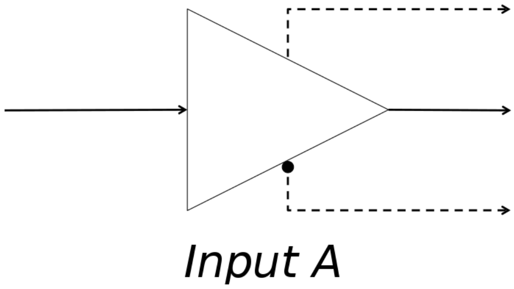


FIGURE 5.4: Input node formalism: it triggers events on input received from an input device. Solid line is for input received; dashed is for cancelled operation and dashed line with black point is for error handling

variants have been introduced and compared in [149]. Statecharts have been adopted with a new perspective to design dialogues [150] or interaction between agents [151]. I propose in this Section a modified formalism to tailor the design of interaction between a user and a physical device. The introduced model just adds graphical expedients that can be completely translated in UML-standard elements with an automatic process.

The theoretical following step is to inject the representation of the statechart in each PHASER node, in order to specify the needed behaviour. An XML representation based on SCXML [152] can be used and it has been already investigated by Di Mauro et al. [153]. However, the integration with PHASER still misses, and it is left as a future integration. Currently the behaviour of each node is defined though external Dialogue Managers; one of the adopted tools is OpenDial [132].

5.4.1 Input/Output devices

The interaction with a physical device will be based on input and output. By proposing a new formalism, I emphasise this part considering new elements to accept and present information from/to input and output modules.

Input

Input are inserted into the system by means of input modules. Graphical formalism for inputs are in Figure 5.4. This node generates events as a new input arrives. The normal flow follows the solid line, but recovery paths are possible. The input element is connected to states as shown in Figure 5.5. It means that a transition from *State X* to *State Y* will be triggered as a new input with the given *value* in brackets will arrive. *State X* and *State Y* are standard UML states.

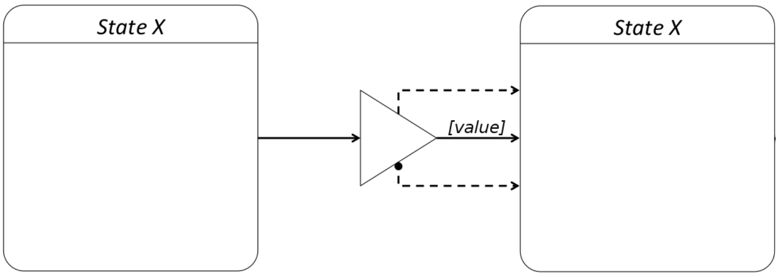


FIGURE 5.5: Input node from a state. An event from that input will trigger a transition from *State X* to *State Y*

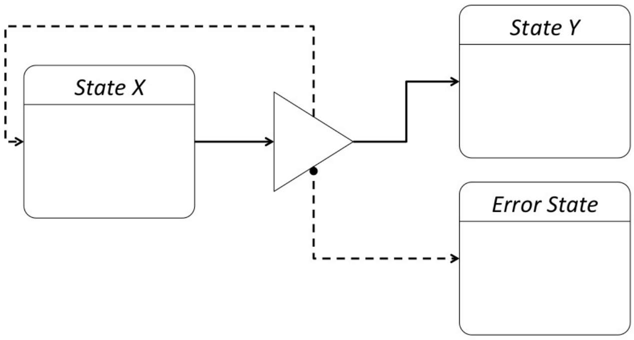


FIGURE 5.6: A transition from *State X* to *State Y* will happen on an input received: an *Error State* will manage errors

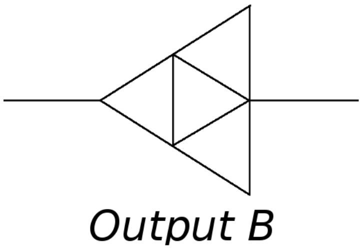


FIGURE 5.7: An output will be presented falling into an output element

Atypical flows can be handled as well; an example is depicted in Figure 5.6. The dashed line with the black circle is taken in case of errors; the dashed line will be taken is the user cancels the operation.

As an example, input modules can be ASR (automatic speech recognition) systems or physical buttons. Their activation triggers events a system uses as a designer needs.

Output

The counterpart of an input management is the output. As for the former case the new element I introduce activates the presentation to the user of a particular data. Figure

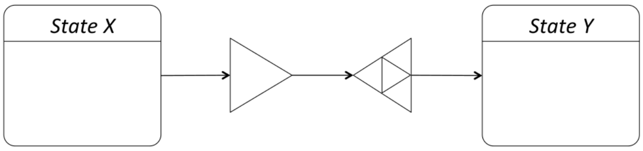


FIGURE 5.8: A transition from *State X* to *State Y* will happen as an input arrives and an output will be presented

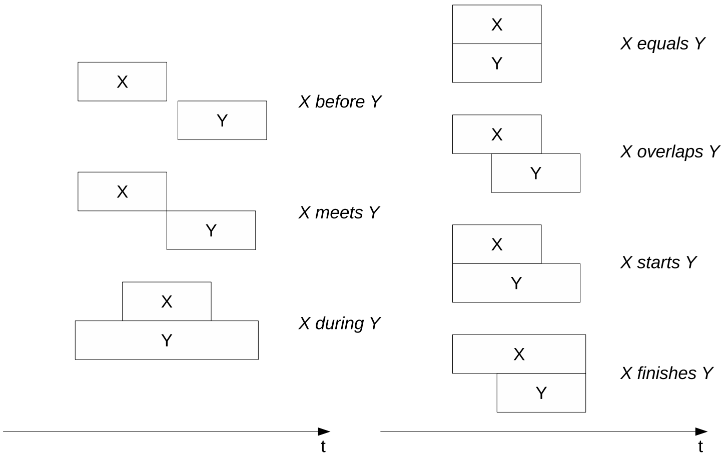


FIGURE 5.9: Allen and Ferguson temporal relations

5.7 presents the adopted graphical formalism for an output device. Output devices can be, for example, TTS (text-to-speech), displays or LEDs. The internal flow depends on the chosen output module and the designer controls it.

Figure 5.8 presents how input and output are intended to be used together; the transition is triggered by an event coming from the input and, going towards *State Y* the system presents some output.

5.4.2 Multi-modal input

The last element I introduced in the modified formalism of a statechart diagram to easier design the interaction with a physical device is an additional input element for flexible multi-modal input. Allen and Ferguson [154] categorised possible time relationships in seven classes, depicted in Figure 5.9. However, as users are involved in the interaction process, strict relations are not always functional; the utterance *"what is that?"* with a pointing gesture may be performed in many synchronisations:

- voice and then gesture

- gesture and then voice
- overlapped signals

A combined input can be performed in multiple ways, and all of them can be valid. A system that supports natural interactions should handle imprecise signals. With a proper tolerance all the listed conditions are valid. Such a tolerance depend on both the situation and the designer. Figure 5.10 presents an input element where two input modules - I_1 and I_2 - provide signals. They compose channels that can be fused and synchronised in a multi-modal block. The rule-based fusion process supports tolerance on each channel. The internal blocks represent expected duration of each channel coming from I_1 and I_2. In that particular case, tolerance means that *"I_2 is during I_1"* but *"I_2 starts I_1"* and *"I_2 overlaps I_1"* are valid as well. The designer establishes both the normal and the tolerated relations.

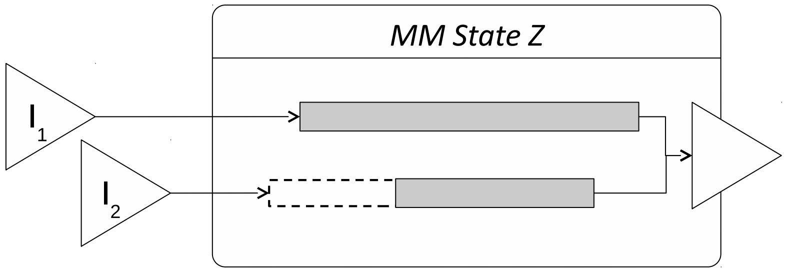


FIGURE 5.10: An input element for multi-modal interaction where the rule-based fusion process supports tolerance. The grey blocks represent mono-modal channels; tolerance is defined by the dashed rectangle. The triangle on the right depicts the fusion block

The outcome of a multimodal input node is an input that complies with a rule-based restriction. The powerful aspect is that this requirement is defined on an abstract level.

5.4.3 Translation

The new formalism I introduced in this Section include graphical elements that aim at simplifying the design of the interaction with a physical device through input/output modules. However, all the items I presented in this thesis can be transformed to standard UML statechart elements; the "translation" complicates the graphical organisation of the elements but it is formally possible. This process moves the design to a more abstract level where internal dynamics are implicitly derived. This formalism fuses information provided by other UML standards like Sequence Diagrams.

Input

As indicated in Section 5.4.1 input nodes can be translated in a set of events for transitions exiting from the connected node. The normal flow as well as errors are included - if considered by the IxD -. Each received signal has a value and starting and finishing timestamps. Filters of the received input will compose the condition of the event.

Output

As well as input items, the work of an output module is included in the state it is connected to. The presentation of information to the user will happen in the *on exit* phase of the current state.

Since the normal flow for the execution of transitions from A to B is:

1. trigger verified in A
2. on exit on A
3. transition $A \rightarrow B$
4. on entry on B
5. current state is B

inputs and outputs are not conflicting and may reside in the same state.

Multi-modal input

As well as normal input items, multi-modal inputs can be translated in events and transitions. The transformation procedure is more complicated here because it requires a more complex condition: received inputs have values and timestamps that need to be checked to verify the relations. However, since the whole process just requires timestamps comparison, the translation with a defined dictionary is possible.

5.5 Environment simulation

The tool I will present in this Section is oriented to an architect mainly focused on the outfitting of the environment. This job is particularly important for a museum, where professionals and architects deeply curate exhibits: architects collaborate with museum curators to carefully define the position of works of art, their context and the environmental lighting. Moreover, since museums are typically crowded places [79], physical barriers help in managing flows of people. Architects decide, with curators, how to regulates these flows.

As introduced in Section 5.2, a PHASER environment can be simulated by populating a virtual world of both PHASER nodes and controlled virtual visitors. The architect would analyse if:

- the environment designed by the expert meets her own needs;
- a simulated crowd of users with some prefixed goals would correctly interact with the environment.

The simulation also helps the IxD because she monitors how simulated PHASER nodes respond to users. The expert is the director that supervises the process.

Environment simulation in IoT is not a new proposal. Commercial manufacturers provide a simulation system for their products, especially if they provide a support for third-party developers. A virtual representation of the considered device is immersed in a simulated context and developers can change environmental data. This is done with Nest[1], for example.

Since PHASER nodes are software products, they can be embedded on a physical device as well as run as normal processes. The latter choice is adopted for simulation in a controlled context. In PHASER simulation has two goals: *(i)* testing each single node and *(ii)* immerse a PHASER *network* in a simulated environment with a proper context.

An additional advantage of PHASER is that nodes composing the network can be just partially executed on physical devices; this composite approach sets up a mixed reality environment [155]. In literature other studies faced the simulation problem providing solutions to emulate smart environments [156]. In emulation, behaviours are simulated in a virtual context, but sensors work in a real environment and provide noisy data. This approach requires a bridge between virtual and real environments, but it is a compromise between virtual simulations - which usually rely on idealised sensors - and costs/security - devices are still simulated and run in a controlled environment -.

This Section is organised as follows: in Section 5.5.1 I will present the proposed environment; Section 5.5.2 explains how virtual characters explore the environment.

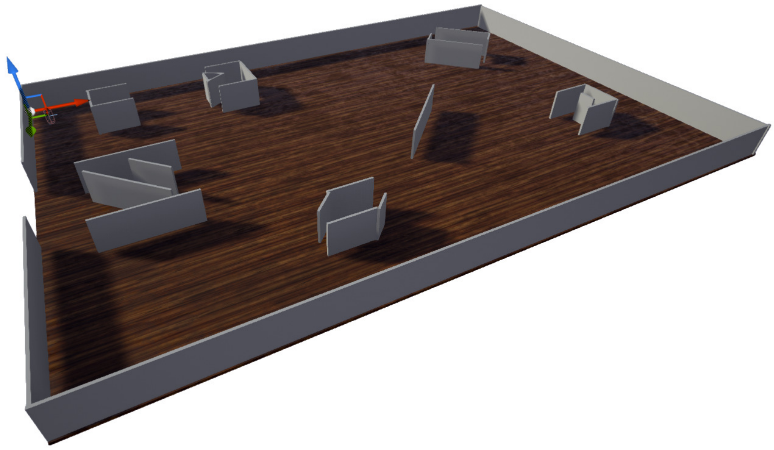


FIGURE 5.11: An example of the world generated in Unreal Engine 4

5.5.1 Simulated Environment

The tool for the architect is an application that receives as input the world created by the expert and proposes it in a 3D rendering. The Unreal Engine 4[2] (UE4) meets these needs and has been adopted as the base framework. By simulating the designed environment, both the architect and the expert have a first overview of the appearance of the exhibit. Such a simulation usually requires a sophisticated management of lighting and textures and UE4 provides a photorealistic rendering. Such a simulator helps in understanding how the environment will appear, if dimensions of paintings and works of art matches with the expert's requirements and if sunlight may improve the final return.

Figure 5.11 depicts an environment as designed by the expert. The *"sketched"* version has been showed in Figure 5.2 in Section 5.3.1. Each placed element by the expert has a corresponding item in UE4 with predefined textures and dimensions. By parsing a XML-based representation of all parts - described in Section 5.3.3 -, a procedural level is created at run-time.

5.5.2 Simulated Behaviours

By using an engine for games, the system provides stronger tools that can exploited. A graphical rendering is essential for a detailed outfitting, but the world will be populated by visitors that will potentially interact with smart devices. By spawning virtual agents controlled by artificial intelligence - known as Non-Playing Characters - and by simulating smart devices - in particular PHASER nodes - the proposed simulator provides a tool to test real case scenarios. Both PHASER nodes and visitors are actors in UE4

[1] *https://developers.nest.com/documentation/cloud/home-simulator* retrieved on September 2017
[2] https://www.unrealengine.com/en-US/what-is-unreal-engine-4 retrieved on August 2017

FIGURE 5.12: A simulation of a museum generated in Unreal Engine 4 with a random number of visitors. Each AI-controlled visitor has an internal utility function, following designed behaviours

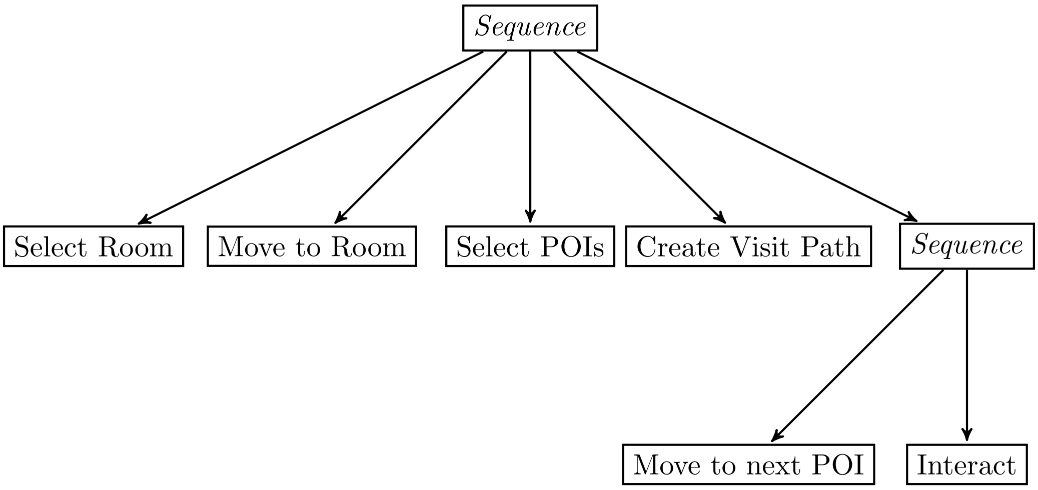


FIGURE 5.13: The behaviour tree adopted by AI-controlled visitors

and the resulting test is actually a game. Figure 5.12 reports a snapshot of a test with a museum with smart paintings and AI-controlled visitors.

By recalling visiting styles in museum discussed in Section 5.2.2, in this Section I provide details about the designed behaviour of AI-controlled visitors to simulate an environment populated of interactive entities.

Behaviour tree

AI-controller visitors populate the simulated environment and follow a behaviour structured as in Figure 5.13. Behaviour trees are well-known structures useful to model Non-Playing Characters (NPC) [157]. These NPCs move in rooms that contains POIs; rooms are not necessarily separated by walls, but are defined - invisible - areas. They have been indicated by the expert in the initial step of the design process - see Section 5.3.1. POIs are, in my case, PHASER nodes representing paintings that responds to interaction requests. In a real case, a painting - or, in general terms, an Object of Interest - would be a physical object equipped with a device where a proper instance of PHASER runs on it.

NPCs iterate on the showed tasks to *(i)* visit the museum and *(ii)* gather information from the chosen POIs. All the visitors follow the behaviour tree presented here, but the assigned style to each visitor may influence the following tasks:

Rooms selection: butterflies and fishes do not expose a particular interest so they may choose to avoid many rooms. Grasshoppers have very specific interests so are usually focused on few rooms. The outcome would be similar to butterflies and fishes but with different amount of chosen zones. Ants would visit many areas.

POIs selection: similar to the description seen for rooms selection.

Interaction: fishes avoids looking at exhibits' details and interacting with them. Butterflies usually ask for superficial information and do not spend much time. Grasshoppers have a specific interest, so a deep interaction is usually expected. Ants expose interest in museum in general, so they spend time in interaction.

The generic distinction provided here is then formalised in utility functions I will present in the rest of this Section.

The "interaction" task would require a list of defined questions and acceptable responses to actually test a PHASER network. However, this step has been left as a future investigation.

Utility functions

In order to simulate the presented categories by their visiting styles, a selection method has been provided. This work takes inspiration from [158], where a synthesis algorithm is proposed to *visualise* visitors' styles. The approach I propose here has a different goal because it provides a method to explore a virtual museum; this means is suitable to monitor simulated users in a responsive context. The object of investigation is interaction and environment setup.

Each visitor has three thresholds:

- room selection
- POIs selection

- interaction

which depend on the assigned category. Each visitor controller selects unexplored areas by generating a random value with a uniform distribution in the range $[0 \ldots 1]$. The controller will select the room/POI if the generated value is greater than the threshold. Regarding the interaction, the process is slightly different, because it requires message exchanging between the controller and the PHASER node. The controller delivers the question - selected from a predefined list - and compares the received response with a set of acceptable outcomes. If the controller accepts the response, it increases an internal parameter of an amount μ. If the value is greater than the related threshold, the interaction stops, otherwise the next question is selected.

Each threshold is empirically defined; all of them are in the range $[0 \ldots 1]$ with the following relations: *Ants* will have high values for each threshold. The opposite is for *Fishes*. *Butterflies* and *Grasshoppers* have average values, but interaction for the latter is higher.

As revealed, this approach uses empirically tuned thresholds but, by analysing real visits, these thresholds can be tailored with measured data to conduct a more realistic simulation. A new exhibition, for example, may be carefully simulated applying thresholds coming from a similar studied context. In Section 5.6 I will describe some adopted Machine Learning techniques to extract the discussed measurements.

5.6 Machine learning supporting simulation

The presented approaches work by simulating users with different visiting styles known in literature. In order to achieve a more realistic simulation, real data may help to analyse real behaviours starting from human visitors. In this Section I propose two investigations performed by analysing visits' logs collected at Hecht museum in Haifa.

At Hecht Museum, visitors interact with objects of interest (OOI) through an hand-held device acting as an audio-guide. The museum is equipped with an indoor positioning system [2] to detect people in defined areas, while the device stores activities made with museum contents. Figure 5.14 depicts an example of antenna for indoor positioning. With the adopted tools, the system monitors the visit and improves the experience of visitors because it is able to propose personalised contents [159], or analyse influences [85]. Personalised content delivery, in particular, is an active research branch and it has been investigated in many studies, as surveyed by Kray and Baus [160].

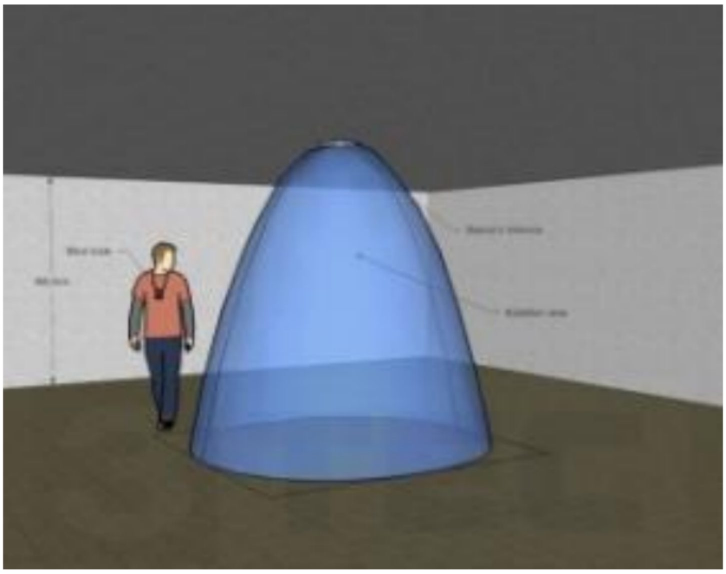


FIGURE 5.14: An antenna of the indoor positioning system used at Hecht museum, Haifa. Source: [2]

Information gathered from the presented devices are stored in logs and reports. They contain:

- explored positions
- history seen contents of each OOI

each item has beginning and ending timestamps. Logs allow to recreate the followed visit and this is very useful for off-line analyses and model productions. The dataset consists of 292 logs - i.e. visits - where visitors explored 47 exhibits organised in 7 different areas.

This Section has two goals. The main objective is finding thresholds tailored on real visitors to improve simulated behaviours; this is discussed in Section 5.6.1. The second goal is to demonstrate that known statistical models used for Process Mining [161], can be adopted as additional tools for museum-related studies. Details will be introduced in Section 5.6.2.

5.6.1 k-Means

In order to enhance the museum experience, many studies proposed techniques and investigations related to this field. These works can be grouped in two parts: the former focuses the attention on visitors and how they interact with exhibits; the latter moves on the museum itself and how architecture, outfitting and similar parameters improve the quality of the experience. These classes are semantically connected, because people will visit museums and their organisation affects visitors' perception.

Regarding museum-based analyses, Attraction Power and Holding Power are well-known measures used in literature [78, 80, 162, 163]. Attraction power measures the percentage of people that have stopped in front of an object of interest in their visit. Holding power indicates the average time spent in front of an exhibit. Both attraction and holding powers derive from observed behaviours and give an overview on the museum.

Visitor-based analyses aim at detecting the visiting style and proposing personalised contents to visitors. Approaches and techniques differentiated projects during the last decades [29, 164, 165].

Detection of visiting styles based on supervised and unsupervised models have been presented in literature [138, 139]. In this Section I propose an unsupervised approach based on k-Means [166] to cluster visitors. The main difference with the other approaches is in the adopted features: while Kuflik et al. [139] used 7-dimensional feature vectors, I use just 3 features: *(i)* number of visited POIs - position -; *(ii)* number of seen contents - activity with the personal device - and *(iii)* average time spent on each entity. *(i)* and *(ii)* were normalised on the maximum values gathered from data for POIs and seen presentations respectively.

I tried values of k in the range $[2, 3, 4, 5]$ and $k = 4$ better represented the dataset. Centroids for $k = 4$ are summarised in Table 5.1.

	C_1	C_2	C_3	C_4
visited (V)	0.325	0.449	0.455	0.520
seen (S)	0.031	0.153	0.180	0.232
avg time (T)	8.609	38.938	67.058	55.765

TABLE 5.1: Resulting centroids for each cluster

The interesting observation is that $C_{2,3,4}$ perfectly reflect the visiting styles of Veron and Levasseur. In particular:

C_1 **is Fish:** lowest values on each feature. A few time spent in visiting the museum and in taking details from each POI;

C_4 **is Ant:** highest values on each feature, but not on *avg time*. More than half museum is explored with many details received;

C_3 **is Grasshopper:** *avg time* is higher than Ant and others. Average POIs visited and seen;

C_2 **is Butterfly:** similar to Grasshopper but with a lower time asking for contents.

It is worth noting that clusters totally hide visitors' motivations: tired - but interested - visitors may have the same visiting style of a unmotivated person. Moreover the system

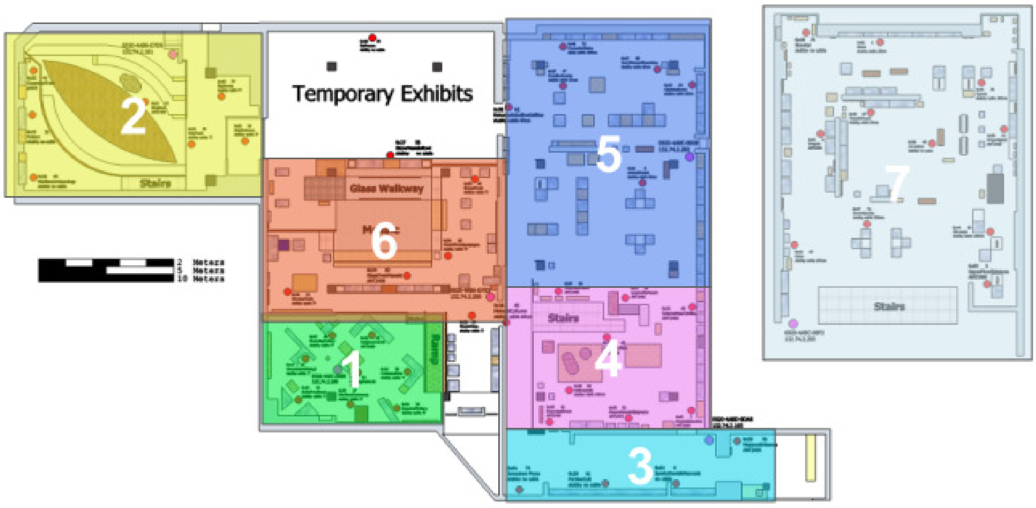


FIGURE 5.15: Hecht museum in Haifa divided in areas. All the POIs belong to a single area. Temporary exhibitions are left out from this analysis

does not guarantee that the user interacting with the audio-guide is actually *listening* for contents. This aspect confirms claims found in literature; moreover logs do not expose this information: they just report positions and activity times without any information about users background nor activity contents. With these premises motivations could be just hypotheses.

By formulating the previous analysis, I found a relation to easily detect the right visiting style. Assuming that $V_{A,G,B,F}$, $S_{A,G,B,F}$ and $T_{A,G,B,F}$ are **V**isited, **S**een and **T**ime features for **A**nt, **G**rasshopper, **B**utterfly and **F**ish respectively, the following relations result:

$$V_A > V_B \geq V_G \gg V_F \tag{5.1}$$

$$S_A > S_G \geq S_B \gg S_F \tag{5.2}$$

$$T_G > T_A > T_B \gg T_F \tag{5.3}$$

The previous result is a useful tool to label a visitor's experience according her own behaviour. However, *it might be expected that a given visitor can change her behaviour during a long visit, and it is also possible that the style is affected by the specific interests*[138]. The investigation coped with this aspect; the work continued by partitioning the POIs in areas and performing the same analyses on each of them. The museum organisation is presented in Figure 5.15. The experiment required to generate a dataset for each area. New datasets contained the same features, collected by splitting visitors' logs in 7 parts, one for each area. A 4-Means on the new datasets has been trained; details are presented in Table 5.2 and the previous relations have been observed in almost all cases. Exceptions have been observed but they may derive from museum's architecture.

	Area 1				*Area 2*			
	A	**G**	**B**	**F**	**A**	**G**	**B**	**F**
visited	.918	.750*	.831	.803*	.585	.562*	.502*	.450
seen	.614	.250*	.430*	.076	.219	.214	.145	.036
avg time	60.306	88	42.177	9	66.312*	55.824*	38.883	9.050
	Area 3				*Area 4*			
	A	**G**	**B**	**F**	**A**	**G**	**B**	**F**
visited	.548	.473	.508	.390	.415	.321	.341	.317
seen	.299	.219	.177	.019	.137	.123	.087	.0
avg time	55.792	66.489	38.921	7.133	58.764	73.667	38.071	10.571
	Area 5				*Area 6*			
	A	**G**	**B**	**F**	**A**	**G**	**B**	**F**
visited	.596	.423*	.557	.550*	.580	.452	.558	.417
seen	.271	.200	.178	.025	.272	.149*	.224*	.028
avg time	55.781	66.404	38.500	10.750	57.363	68.731	39.410	8
	Area 7							
	A	**G**	**B**	**F**				
visited	.606	.520	.529	.417				
seen	.279	.180*	.208*	.017				
avg time	57.768	68.533	40.583	8.833				

Table 5.2: Centroids with corresponding visiting style - Ant, Grasshopper, Butterfly, Fish - with the proposed features split in 7 areas. For cells with starred values relations 5.1, 5.2 and 5.3 do not hold

The approach presented in this Section can be integrated in the simulation step of PHASER by considering clusters with a complete visitor experience. Features indicate threshold values discussed in Section 5.5.2 and are now gathered from real visits. In order to simulate a longer visit with different styles, the second experiment can be used to detect the most probable changes - as a sort of probabilistic transition matrix - and replicate them on the simulation. The latter part must be intended as a future work, because the simulation just supports threshold tuning.

By assigning a visiting style to a NPC, the system does not emulate *motivations*. In order to implicitly affect them, the designer may operate on the lists of questions and related thresholds: focused questions and high interaction threshold, for example, represent a demanding user with a specific interest. Another character with general requests and low interaction level, instead, easily satisfies its own interests without deepening contents.

5.6.2 Process mining techniques

Process mining techniques are able to extract knowledge from event logs commonly available in today's information systems. These techniques provide new means to discover, monitor, and improve processes in a variety of application domains [161].

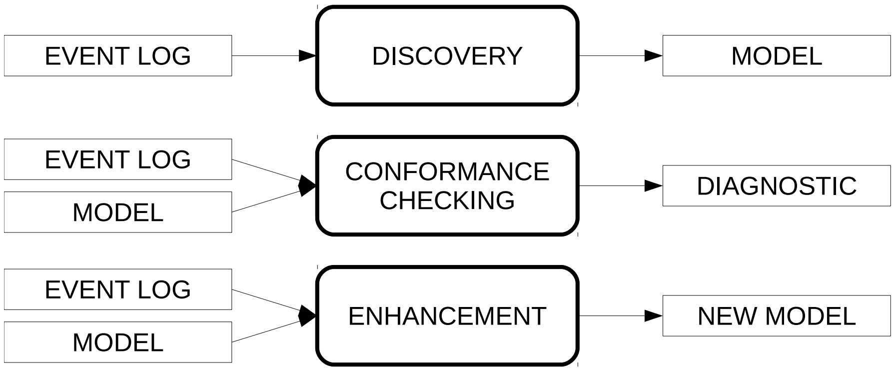


FIGURE 5.16: Three types of Process Mining techniques. Di-sco-ve-ry, Con-for-man-ce Che-ck-ing, and Enhancement compose the three pillars of this discipline

Discovery, *conformance checking* and *enhancement* are three main types of Process Mining [167]. *Discovery* extracts information from event logs, producing a model without any *a priori* knowledge. *Conformance* techniques are used to check if a model reflects reality. Checking is performed with a new event log. Eventually, *enhancement* techniques are used to improve or extend the input model by presenting new event logs. Figures 5.16 depicts a schematic summary of these techniques.

In this Section I present two well-known techniques used in Process Mining: Petri Net and Social Networks; they are usually adopted in a Business context. I propose their application in a museum context.

Petri Net

In this Section I presented three main classes of techniques usually investigated for Process Mining. Discovery often uses Petri Nets [168].

Petri Nets are a diagrammatic tool to model concurrency and synchronisation in distributed systems. They are mainly used as a visual aid to model a system behaviour. *Places*, *Transitions* and *Arcs* compose Petri Nets. Places are possible states of the system; Transitions are events or actions that which cause a change of state. Arcs connect a place and a transition or a transition and a place.

Since Petri Nets successfully model concurrency and synchronisation, they are extensively used in Work-flow Management [169], Business Process Management [170] and Process Mining. In Process Mining, the starting blocks are always event logs. Petri Nets are used to *"[...]provide an insight into the behaviour captured in the log"* [171].

By monitoring visitors in the Hecth Museum, Haifa, researchers have collected visits' logs. Logs contain information about visitors' position and activity, gathered through a technologically enriched audio-guide, as stated at the beginning of this Section. The

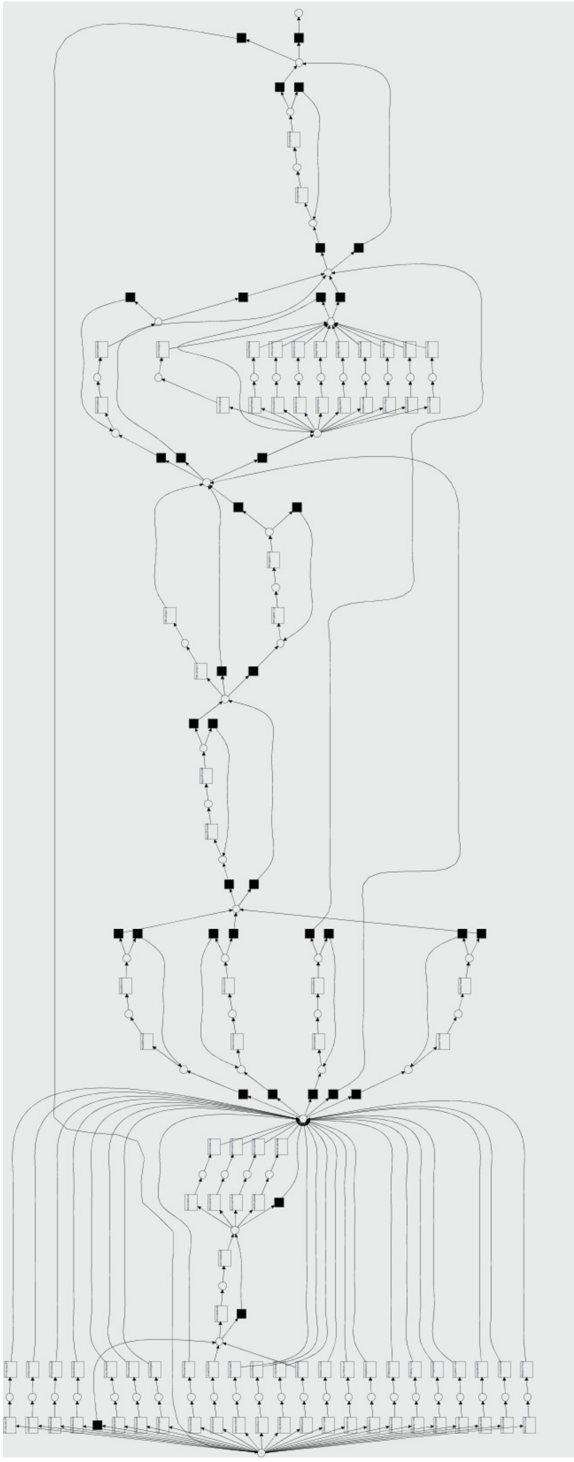


Figure 5.17: A Petri Net produced by event logs gathered by monitoring visitors' activities. Places are white rectangles, while transitions are white circles. Black squares represent operators to split or join parallel flows. The model shows relations among POIs, reported as places. The sequence *Place-Transition-Place* represents begin and end of the visit of a single POI

museum contains 47 POIs organised in 7 areas. Figure 5.17 depicts a Petri Net obtained by analysing visitors' behaviours. The net shows that many POIs are explored without a precise pattern - a number of parallel POIs in the beginning -, but relations can be found on others.

Social Network

> *Process mining techniques typically focus on performance and control-flow issues. However, event logs typically also log the performer, e.g., the person initiating or completing some activity*
> – Van der Aalst in [172]

Social networks in Process mining are usually adopted to monitor how people on tasks are connected with each others. The graph is built upon received event logs. Some people could be involved in many projects/tasks or, on the opposite side, too ignored. This analysis helps in balancing the load each person has.

By revisiting this technique for museum contexts, I propose here an approach that, starting from event logs gathered on visitors, shows how POIs are visited and connected with each others. This may help to look at the museum by a different point of view and comparing expected behaviours with real ones. The study I present here acts has not been deeply evaluated. The explanation aims at showing that well-known methods, already adopted in Process Mining, can be adopted in the museum context to assess how people approach POIs in a museum.

Figure 5.18 shows on a social network graph how POIs of Hecth museum, Haifa, are connected with each others. Links are related to visitors explorations. Size of nodes reports their degrees, in terms of the number of edges that connect to them. As the figure shows, vertexes are not circles but ellipses: width is proportional to incoming edges, where height is related to outgoing links. Although names are not very readable, some representative cases can be extracted: small nodes are remotely reached POIs; bigger nodes, instead, are important nodes. Large nodes are visited from many other places, while height and tight vertexes open paths towards many other POIs.

5.7 Summary

In this Section I presented supplementary tools for PHASER, that can be used to better define the behaviour of both a single node and the whole network. Three actors have been identified.

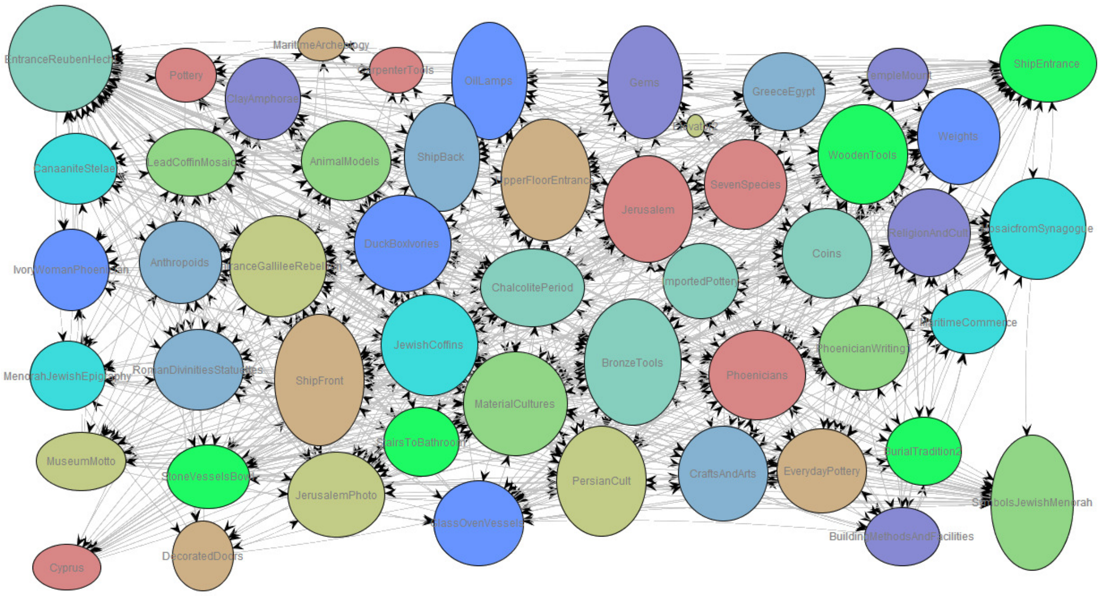


FIGURE 5.18: Social network graph usually adopted in Process mining. Nodes are POIs and their shape is related to their connections. Ellipse's width is proportional to incoming arcs, while height is proportional to outgoing links. Although names are not very readable, vertexes at the corners are representative examples

An architect, oriented to the outfitting of the environment, defines, by means of a 3D representation of the world, design details. An Interaction Designer defines the behaviour of each PHASER node; I proposed a formalism that extends classic Statecharts to better defines the interaction with a physical device. An *expert* has knowledge about the considered domain; her job is divided in three main parts: *(i)* placing of each element; *(ii)* defining knowledge and how each node should support the user, with the Interaction Designer *(iii)* establishing the outfitting of the environment with the Architect.

All the actors take advantage from a simulation - performed on museum scenarios - of the whole designed environment. AI-controlled characters simulate visitors with different visit strategies; they help in preliminary testing the system. In order to study how visitors behave in museums, I proposed studies based on Machine Learning methods to categorise visit strategies. Preliminary results of the use of Process Mining techniques in museum context has been investigated.

Although the Interaction Designer is not fully integrated with PHASER, the workflow of the whole process gave good preliminary results. Currently each behaviour is controlled by an external Dialogue Manager; one of the adopted tools is OpenDial. Both the Interaction Designer and the expert collaborate to define the intelligence.

The use of games techniques in museums has been extensively explored in the last years. However, one of the latest proposals is investigating the use of games to create interest and curiosity towards museums [173]. Narrative, scene and characters are build starting from museums' objects and historical facts. Differently from "classic" approaches, based on serious games or gamified experiences, this approach proposes a playful experience and it does not aim at suggesting people to visit museums directly. Its goal is to elicit users' curiosity with important elements of real life. Characters customised in games are then represented in museums in order to revoke parts of the game.

The whole theoretical background has been discussed by Origlia et al. [174] and is called MES (Museum-centric Entertainment System).

The Chapter ends with an introduction of methods, usually adopted in Process Mining, applied to the museum context. In particular they concern Petri-Net and Social Network analyses. The first model is useful to evaluate how POIs are connected with each other, by monitoring users' visits. If recurrent connections happen in the visit, a link appears in the Petri-Net. This information can be useful to curators and professionals to assess the status of a museum. The second approach is based on Social Network studies, by comparing visitors and POIs. The presented graph showed museum elements with different features. The discussed approaches are complementary because they move the focus on POIs and visitors as connected entities.

Chapter 6

System evaluation

In this Chapter I will present tests I conducted for PHASER. The experiments concern the core part of PHASER, that has been presented in Chapter 3. Experiments and results about linguistic analyses and conflict resolution have been already discussed in Chapters 4.

In order to test PHASER, I conducted different types of experiments. The object of evaluation is the infrastructure, aiming to assess if the proposed model can be a valid choice for ubiquitous computing; another important aspect concerns Human-Computer Interaction, investigating what people expect from a PHASER network and issues that such a model could arise.

This Chapter is organised as follows: in Section 6.1 I will presents a smart-house case study where both simulated and real networks have been considered. Similar evaluations will be exposed in Section 6.2 on a simulated smart-museum. Section 6.3 focuses on Human-Computer Interaction issues generated in PHASER, analysing how people feel in interacting with typical devices to control others. This Chapter ends with final considerations.

6.1 Home Case Study

Single PHASER node interaction has been tested in Wizard-of-Oz techniques [175], where I tested internal dynamics reproducing recorded dialogues. This was useful to assess flexibility without a completely implemented system. Flexibility is related to nodes that can be configured to represent different actors in a PHASER network. By assigning proper values to parameters discussed in Section 3.2.2, I simulated a network composed by appliances and people without a hierarchic structure.

I also conducted stress tests by simulating devices to assess the reliability of the multimodal input structure in simultaneously receiving many input data. I simulated N input devices producing nonsense data - strings, points and numbers - for each channel. I used an iMac mid 2010 Intel Core i5 a 3.6 GHz. I stressed the module up to N = 80 input devices producing data each 100ms having a CPU consuming at 8%. Although the tested has been conducted in a very unrealistic situation, the system successfully sustained such a load. Details have been published in [96].

In this Section I propose tests in smart-house scenarios. My analysis is focused on network communication with the aim of proving that the algorithm proposed in Section 3.3.1 for the *"Navigation problem"* reaches the target node in a reasonable time and steps number. Another test, in simulation, uses a more realistic network. In all the presented results, the target is the node able to provide the desired output. The Section ends with experiment and results obtained with real users.

Simulation

In the considered case of study number of required nodes rarely goes beyond tens, and searching algorithms do not work in very challenging situations, where undetermined solutions are possible. Nevertheless, in order to test convergence features, quality and processing times of our algorithm, I generated networks with 10^3 devices. The network was divided in sub-networks with random arcs: connections within the same area are more probable than connections between different sub-networks. I generated networks with $2^{0..10}$ sub-networks with a number of nodes from 22 to 4134. Since the classes of devices where limited, many nodes were assigned to the same class - n kettles, m ovens, etc - but this do not influence the test because the algorithm does not require limited numbers of devices.

By defining both the request and the target, I simulated an interaction, starting from a random node, using the algorithm of Section 3.3.1 as a measure of quality; μ, τ_{max}, τ_i and ϕ have been set to 0.1, 0.5, 0.3 and 0.05 respectively. Figure 6.1 shows the collected lengths of paths. The first and second columns contain the observed lengths at the first iteration and after 30 adaptation steps, also adding new relevant links; starting from a reference set of N nodes in a graph with $N^2 * 30\%$ connections in total, in repeated simulation, we added up to 7% of the connections. The third column shows the shortest path length on the same initial topology. Eventually, the last column reports a stability test. By analysing the paths followed during each test, nodes have been ranked according to their presence in paths. The most important node is labelled as the most visited one; start and target nodes were excluded. Stability check has been obtained by removing

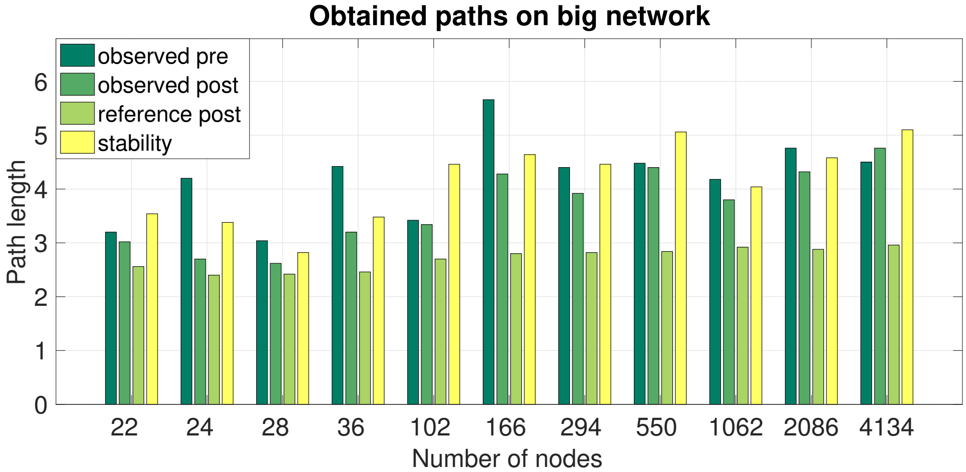


FIGURE 6.1: Observed paths pre and post the adapting phase, comparing post with the shortest path and including stability checks

both important nodes in the network - after the adaptation - and the new introduced arcs. This aims at simulating nodes that suddenly fall down in a network.

Realistic network

With the same approach, I generated a realistic network where a reasonable number of connected devices populated a smart house. The goal of this test was to assess if the quality of a connection affects the results.

The house was composed by 3 bedrooms, 1 bathroom, 1 living room and 1 kitchen. Alarm clocks, washing machines, kettles, ovens, microwave-ovens and cookers have been considered. 28 nodes composed the total network. Different network topologies have been considered. They all followed the same structure: nodes within the same area shared the 80% of connections, while inter-areas connections had a changing percentage from 5% to 30%. In order to be statistically relevant, each single experiment has been repeated 50 times; the whole process has been performed on 10 generated topologies having the same structure, but connections among nodes were potentially different. Also in this test, μ, τ_{max}, τ_i and ϕ have been set to 0.1, 0.5, 0.3 and 0.05 respectively.

The more connections nodes share, the faster could be the network in providing a response, because each node is more probably connected to the target. However, a full-connected graph is not always desirable because of infrastructure limits, especially with large networks. The considered topology aims at promoting links between nodes in the same area. This encourages intra-area interactions; however, the system is not limited to them because connections among different zones extend this possibility.

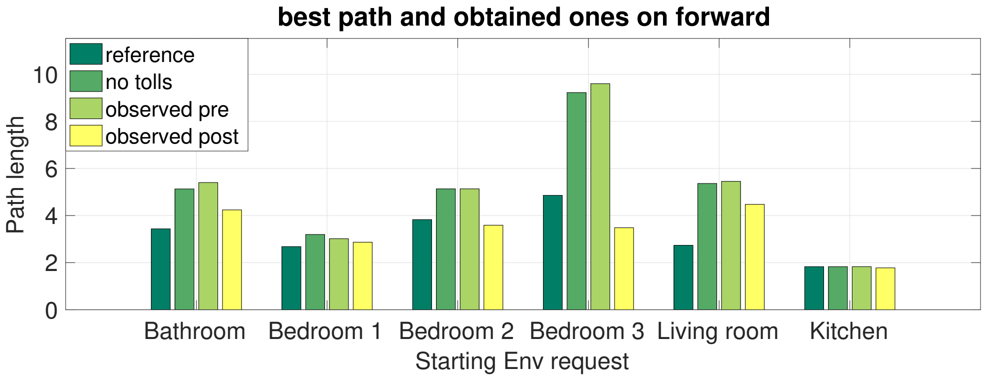


FIGURE 6.2: A comparison in realistic situation with 28 devices and reasonable connections

The system has been tested as follows: similarly as in previous test, I chose the target node T. A random node R_1 from each room was picked and then I calculated the shortest path between R_1 and T; used as reference in comparison with the forwarding algorithm without tolls, with $toll = 0.1$, and the adaptation process as explained in Section 3.3.1. In the second step, the resulting network was tested picking another node R_2 in the same area. The network was trained in seven iterations. Resulting data are showed in Figure 6.2.

Smart house

During a visiting period, I conducted these experiments at the "Smart Space Lab" at Middlesex University, London[1]. *Living labs* are houses equipped with sensors used to develop context-aware and ubiquitous systems. These labs provide naturalistic user behaviours in real life. They are more realistic than smart rooms which fail in reproducing daily behaviours and limit movements [101]. Although living labs are not *real houses*, testers may spend a longer period there than typical research lab and their feeling with the environment better simulates a realistic scenario [176].

In this test, I asked users to solve the following scenario:

> *You come back home after work in the evening, with some gym clothes to wash. You want to reach your friends for a party in an hour but you need to do something before: washing your clothes, preparing your dinner, dressing for the party. You are lucky because the smart house can help you.*

The description continued by presenting a set of available devices, with brief tutorial to control them and the three tasks in a structured organisation. The network was

[1] I thank very much Juan C. Augusto for his precious advice, support and hospitality during my stay in Middlesex.

composed of 8 devices: washing machine, radio, kettle, oven, TV, alarm clock, heating, light controller, organised in four rooms. Each user interacted with the devices through a web-page - running on tablets -, where an image clearly represented the associated intelligent device. The tablets were close to the devices itself, in the proper room. All the nodes run on three different machines, but they were connected and remotely reachable. The tests have been performed by 10 people, mainly international students recruited from the university.

I monitored behaviours in interacting with the house. It appeared that users tend to maintain their status: if they are comfortably on the sofa - as in the beginning of the test -, they prefer to remotely solve tasks. If they are standing, instead, they prefer to walk and reach the target of the next step; it has been noted for the *"washing clothes"* and *"dressing"* tasks. However, when they remotely solved the tasks, in the 90% of the cases they used the same device as interface from the beginning to the end of the task. In other cases, they did not show an evident preference between remote control and a closer interaction with physical devices, but they felt comfortable using the system in both the ways. This was the case of the *"preparing dinner"* task. It is possible that background and personal interests affect this measure.

A relevant weak point that compromised the experience of users has been detected on the voice recognition. I used an *off-the-shelf* solution, integrated in javascript, running behind the web-pages. Although it was set on British English, it failed the 62% of cases, also with British people. In one case, for example, the user had to repeat 5 times the utterance *content fridge*, with 24 seconds spent in total. In other cases, instead, the parsing of the request for the *kettle* required an hint to solve the task: the kettle did not accept *"turn on the kettle"*, but just *"turn the kettle on"*. However, the same problem would be run into a centralised system, as well.

In conclusion, all the testers perceived the system as a compact block, similarly to a centralised system. Volunteers, asked to express their preference for distributed vs. centralised systems, preferred a distributed system in 8 cases on 10.

Discussions

In the first experiment, I observed that, after the learning phase, the system reached the target in fewer steps than *"observed pre"*, also in big networks. The *t-test* confirmed the hypothesis that "*post* data are lower than *pre* ones"; this aspect has been noticed also in the biggest topology, where the *post* average distance is slightly higher than *pre*. Concerning the stability test, although the removed nodes affected the final result requiring longer paths, the system was able to reach the target anyway. The request reached the target in 25 milliseconds, even on the biggest network. I run all the nodes on the same machine, on different processes, in order to be focused on time required by the algorithm. This way I excluded transmission delays and additional time a Dialogue Manager may require in a real situation.

Concerning the quality of connections, it appears that PHASER works better when connections reflect a “semantic links” between two devices; a semantic link means that users would - unconsciously - use that connection to require actions. Alarm clock and Coffee Machine can be semantically connected, for example, because a person may ask to prepare a coffee in the morning from the bed. The test in the realistic case did not use semantically links but random connections. However, the test proved that, after some learning steps, PHASER starts using relevant links; they often match with semantic links. The test also showed that providing an optimised network is a complex job, that highly depends on the context and the available services and nodes. For this reason the initial topology should be designed from an expert for the considered domain.

Interesting results came from the last experiments with real users. Although 10 users did not completely explain when remote interaction is preferred on the close - and classic - one, the experiment is successful because people spontaneously used both the approaches. Future efforts will be spent to investigate *(i)* how much personal interests affects behaviours in the proposed context and *(ii)* if PHASER may exploit this information to adapt nodes’ behaviours on this aspect. It is possible that an extended test, conducted in real houses, would expose a clearer evidence.

Eventually, users perceived that they could control the whole house from each device, accomplishing the goal; however, they found unnatural talking with a specific device to manage everything - e.g. the fridge to switch the light on, etc. -. This aspect arose questions about which strategies are better to make the users feel the interaction as “natural”. A possible strategy they advised is to elect a unit as manager for all the devices in each room. Alternatively, one can coordinate all the devices with the same interface, hiding shared intelligence. I will present in Section 6.3 tests focused on this issue.

6.2 Museum Case Study

In this Section, I propose an experiment similar to the one shown in Section 6.1, referred to a smart museum. 10^3 nodes compose the network. The topology was automatically generated following guidelines appeared in the previous test. However, the best and optimised solution should be designed by an expert of a museum context.

Environment	Avant-garde	Impressionism	Naturalism	Baroque
p-value	0.99995	1	**1.4961e-07**	1
Environment	Mannerism	Renaissance	Primitives	
p-value	0.26782	0.10986	**0.00045991**	

TABLE 6.1: Obtained *p-values* from the Wilcoxon test with *null hypothesis* that adapted median length is lower after adaptation steps. Although the null hypothesis is refused in two cases, median values in *Primitives* are almost comparable

The environment was composed by 7 different areas, following the organisation explained in Section 3.1, where different areas corresponded to diverse artistic movements. The adopted ontology contained paintings for each considered movement. In order to reach 10^3 nodes, many of these paintings were repeated, but this did not affect the experiment because they just filled the network. The topology had the 80% of connections between nodes in the same area and connected objects in different environment with the 30% of possibility. Each node was equipped with input accepted utterances, used to rank the nodes. The objective was to reach a chosen object, which is the *target* of the interaction. Each iteration starts from a randomly selected object in the same area. After 100 attempts, the system was reset and the test started from the next area. Figure 6.3 reports the obtained paths' lengths without the adaptation step and after 40 iterations in which internal parameters are tuned. Table 6.1 reports *p-values* of the Wilcoxon test, where the null hypothesis stated that the lengths after the adaptation is lower than the starting one.

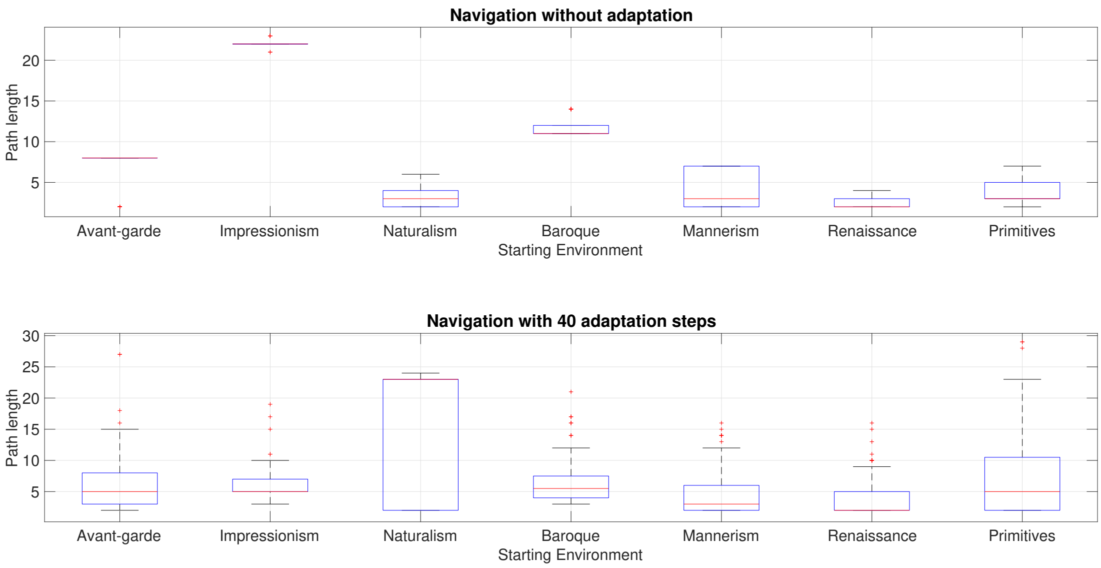


FIGURE 6.3: Box plots of the obtained paths' lengths in request navigation. Each box reports the distribution of paths of 100 requests delivered to nodes in each available environment

Discussion

In this Section I adapted PHASER to a museum, where each work of art was technologically enriched with responsive capabilities. The world was organised in seven areas, each containing paintings. The system run for 40 adaptation steps in order to tune internal parameter and evaluate if the system is able to learn from the observed requests. By knowing the target, a classical shortest path algorithm could be faster in reaching it, but in the considered case we cannot know the target *a priori*. In this condition, a shortest path search cannot be applied. The obtained results showed that the system has been able to adapt its behaviour to the received requests. In two cases it failed - *Naturalism* and *Primitives* - confirming that the initial topology is crucial to evaluate our approach. Nevertheless, in the latter case, the *Primitives*, the obtained median value is almost comparable to the initial one.

6.3 Human-Computer Interaction issues

The experiments presented and discussed above have been focused on architectural aspects of PHASER, showing that the proposed algorithm efficiently solves the Navigation Problem; however, additional steps, such as the conflict resolution discussed in Chapter 4, may require extended processing delay, and this could compromise the user experience. Observed times may reach 1.5 seconds in most cases. Real people have been involved in the smart house; although they did not disapprove required time, interesting considerations arose. In this Section I propose an experiment focused on Human-Computer Interaction issues, analysing how people feel better in interacting with networked devices.

The problem of interaction in an Intelligent Environment is not new [101], but recent studies investigated how connected devices should respond in Intelligent Environments [177]. Mennicken et al. [178], for example, questioned about how to design interaction in a pervasive system, focusing on humanised aspects of an interface. The conclusion of that work is that, although an interface can follow users hints and behave like they desire, that system would never be a friend, but a *friendly stranger*.

In this Section, I focus the attention on users, analysing if they prefer an approach of interacting with a PHASER network and assessing if the discussed delays may alter the user experience. I proposed a museum scenario, where the environment represents a museum divided in two rooms with 5 paintings; people ask information to paintings through their own smartphone; I developed an Android app for this test. This device, indeed, acts as an interface between the user and a simulated PHASER network. The interface of the smartphone can switch in two modalities: list of paintings and avatar.

In both the cases, topological details are hidden for the user. In the first modality, the user consciously selects the object of the interaction from a list with thumbnail and title; in the second, the interface does not show details, but the user needs to specify the related painting at each request. While in the former case the request is delivered to the previously selected node, in the latter version the best node is chosen to process the request. The network is composed by 5 paintings, organised in 2 categories. Users can ask generic information, such as author, description, style and age. The setup of this test is different from the one done in the smart house because here the user always accesses the network through smartphone. In the previous case, instead, the user had the possibility to change the approach. The Android interface gives the possibility to personalise the channel of interaction; the user can deliver the request by typing or pronouncing it. The response can be read - through the Text to Speech module - and the size of the written message can be adjusted.

This test has two goals: *(i)* observing if users show a clear preference in choosing the modalities; *(ii)* by intentionally introducing delays and failures during the interaction, I assess the tolerated inconveniences. Concerning the second point, incremental noise is inserted in the interactions; 4 profiles are summarised in Table 6.2.

Profile no.	1	2	3	4
delay (ms)	600	1500	2000	2000
ask repeat		once		once

TABLE 6.2: Profiles of the conducted experiments. Delays and *ask repeat* simulate times required for the propagation of the request and possible failures in the interaction

Delays are inserted between the delivery of the request, and the presentation of the response. The minimum delay (600 ms) is considered as baseline, as it has been observed in conflict resolution - details in Section 4.3.2 -. Incremental values extend the interaction and follow delays observed in simulated experiments. A failure in the interaction asks to repeat the question in both typed of spoken interaction. Other failures may arrive if the system actually does not understand the message.

The system has been tested by 20 italian people, 5 for each profile. None had experience with PHASER or similar models. 12 did not have a technical background, while 10 had experiences in laboratory activities. 16 people were in the range 20-35, while 4 over 50. The users participated to a survey after the test session; they assigned a 10-levels Likert scale, where higher values refer to a higher quality. Concerning the modalities of interaction, 14 people preferred list-based interaction with an average vote of 7.236; 5 people would choice the avatar-based one, evaluated with 6.368, while just 1 did not show preferences; 7/20 people selected the avatar-based interaction as first choice. The principal criticism was related to the low smartness of the interface; it was uncomfortable

X profile	Y profile	hypothesis		$p-value$
		null	alternative	
3	1	$X = Y$	$X \neq Y$	0.786
3	4	$X = Y$	$X > Y$	0.008
3	2	$X = Y$	$X > Y$	0.006

Table 6.3: Results of the Wilcoxon tests run on the votes gathered from users. The first and second columns represents the compared profiles values; the third and fourth refer to median values of votes; higher values relate to a better evaluation. The last column is the resulting $p-value$. Each test has been performed with $\alpha = 0.05$ significance level

to specify the name of the painting at each request. Users positively evaluated, instead, the customisation of the interface, with an average value of 7.74. I did not appreciate a difference between the different ages.

Interestingly, delays did not affect the experience. While profiles 1 and 3 gave comparable votes, the system "intelligence" deteriorates the user experience, with a final worse evaluation. Table 6.3 summarises results of the Wilcoxon test, conducted to compare the evaluations. As an explanation, $p > \alpha$ means that the null hypothesis cannot be rejected; otherwise, the alternative is valid.

6.4 Summary

In this Chapter I presented experiments conducted to assess the PHASER model. Tests have been focused on different aspects. Simulated networks have been used to test the algorithm proposed for the navigation problem. Other experiments focused on users; real scenarios have been proposed to evaluate users' judgements in interacting with a distributed model. Users interacted with devices mainly recurring to voice.

The experimental setup with real users is based on studies known in literature; Möller et al. [179] identified some important factors that influence the perceived quality of the service provided by a dialogue interface: user, agent, environment, task and contextual factors. User factors include attitude, emotions, experiences and knowledge which affect the users evaluation. Agent factors are related to the characteristics of the machine counterpart. Environment factors relate to the physical context of the interaction. Task and contextual factors relate to the non-physical context [101].

All the experiments showed that PHASER requires a careful design of the initial topology and intelligence of each device. However, since the connections are automatically adapted on the history of interactions and the topology is optimised after few steps, the proposed model is a valid alternative to currently adopted approaches. PHASER advantages concern flexibility and scalability; the network tolerates new introduced devices and is not limited to a fixed number of nodes. In addition, the algorithms are domain-independent, and they gave similar results in both the smart house and smart museum.

In the last test, a closer look on users has been taken; they clearly preferred the list-based interaction over the avatar-based one in the proposed case study. However, they tried both the modalities, and they found comfortable that the same interface can be used to access all the paintings. It is possible that a more intelligent behaviour, based on the detection of the proper painting and/or with a more friendly interaction, would affect the judgement; this will be subject of future investigations. Operation times observed in previous experiments of PHASER has been proposed to users, intentionally introducing delays in the interaction. This aspect did not affect the opinion; however, the system must be intelligent to tolerate misunderstood questions.

Conclusions and future works

In this thesis I presented PHASER, a distributed model that defines a ubiquitous infrastructure for Internet of Things scenarios in multiple domains. It defines a network of nodes - that in real cases can be both software agents and physical devices - which are grouped in environments, but they all run at the same hierarchic level. PHASER brings several innovation points, that review the currently adopted strategies in this field. PHASER has been devised for smart homes and smart museums, but it would accept any structured represented environment. With this PhD, I did not aim at proposing *just another ubiquitous model*; my goal was stating that a different direction is needed and that complementary jobs should be led, encouraging collaboration and creativity. The general idea behind PHASER is that users should be pervasively sustained, providing Natural User Interfaces as much as possible. However, designers and professionals without a deep technical background should be supported in setting an Intelligent Environment up.

One of the goals was to prove that a set of independent nodes can be arranged through a *relaxed* protocol and that they all represent an Intelligent Environment. Each node interacts with people and reports them outputs. However, if an entity is not able to produce a proper output, it shares the request to the network. Broadcasting is not a preferred solution and nodes rely on partial information about both other nodes and their knowledge. By recurring to routing solutions, based on context-awareness and network adaptation, PHASER moves the routing of the request to a more abstract level, proposing a forwarding strategy that depends on the object of the interaction and on how the network responded in previous interactions. These aspects have been discussed in Chapter 3, starting from a formal definition of PHASER elements and algorithmic solutions for the *"Navigation Problem"*.

During these three years, I revised the routing solution, proposing alternatives that improved the final result. Well-known techniques of Information Retrieval and Natural Language Processing have been adopted to enhance the ranking of nodes according to the received request. In addition, in order to ensure network consistency, conflict resolution strategies have been presented. They take into account multiple levels of description: request, provided services and influenced services. The proposed approach has resulted correct and complete; experiments showed that the system involves all the needed nodes, without overloading the network with useless interactions; the best node always won the conflicts and required times have been satisfactory. All these arguments have been presented in Chapter 4.

Intelligence and interaction need to be carefully designed, in order to tailor behaviours and interfaces on users' needs; together with PHASER, I proposed various tools to support professionals, identified in three actors covering all the needed aspects. An *expert* has knowledge about the chosen domain. She decides multiple aspects of the final network, assuming a role of director of the designing process. An *Interaction Designer* defines the behaviour of each PHASER node placed by the expert. Although dialogues are currently managed by external tools, such as OpenDial, I proposed an improved statechart formalism that simplifies the description of the interaction with a physical device. The last actor is an *Architect*; she is mainly oriented to the outfitting, ensuring that nodes do not compromise the decided scenic design. I proposed a platform to support all the actors; tools have been designed with a User-Centred Design process and a discussion about Personas and Scenarios have been used to both design and test PHASER networks.

With this PhD, I wanted to propose PHASER as an alternative approaach to current commercial products. It is not a competitor of traditional solutions, but it provides a valid alternative for many situations where independent devices can be arranged to compose an Intelligent Environment. The experiments conducted in various situations showed that PHASER is ready to be used as an infrastructure for smart devices. Future investigations will be directed on a more intelligent generation of an initial topology, by recurring to genetic algorithms to propose a network that would follow required interactions. Another important aspect I will study, is an additional support at run-time in museum contexts, where a distributed infrastructure may help visitors in exploring the museum. Current methods used to categorise visiting styles are centralised, but on-line classifiers have been explored. My idea is to propose this approach on an IoT infrastructure, where all the nodes may contribute to this categorisation; by relying on the exposed history of visit, and by orchestrating interactions among locally connected PHASER entities, a node may recommend to a user the next step in her visit following both her detected behaviour and curator expectations. In my opinion, this is an interesting support that a distributed model may provide to both professionals and visitors.

Appendix A

Applications for Smart Tourism

Over the past six decades, tourism has experienced continued expansion and diversification, becoming one of the largest and fastest-growing economic sectors in the world. Many projects have been funded to encourage research in proposing innovative technological solutions for Cultural Heritage. One of them, the Or.C.He.S.T.R.A. (ORganization of Cultural HEritage for Smart Tourism and Real-time Accessibility) project [110], aimed at providing solutions for smart cities and cultural heritage support in Naples, Italy.

The mission of Or.C.He.S.T.R.A. was to support tourists in all the steps of the visit: from the organisation, to *on-site* tours of the city and even later. In this context, for the *on-site* part three projects have been proposed: PaSt, a CAVE-like environment that provides an experience for a group of people; E.Y.E.C.U., an Eye-Tracker tailored on exploring paintings and Caruso, a smart audio-guide based on dynamic 3D soundscapes. All the projects aimed at dematerialising the interface [145], avoiding visual barriers between the user and the real world.

In this Chapter I will discuss about the cited products. Although they were isolated solutions, PHASER has been devised as a common infrastructure of all the projects and an easy bridge for an extended support of users.

A.1 PaSt

PaSt [153] stands for *"Passeggiata nella Storia"*, (in english *"walking through the history"*). It is a CAVE-like environment that tracks a small group of people on a defined area and present them multimedia contents. The leader of the group - who come first in the area - can perform gestures with the hands to interact with the proposed projections.

The system is modular, in order to easily customise PaSt: a combination of SCXML[1] and SMIL[2] are used to design the interaction. SCXML represents a statechart that defines the workflow, while SMIL is used to describe the contents to deliver.

Apparently PaSt contrasts the common philosophy, recurring to barriers. However, PaSt has been adopted to navigate information that cannot be explored in Naples: the 10 layers of the city. The floor reported the map of the city. By recurring to gestures, the leader can change the era; the system proposes a different map, and shows how the city changed. The map has some hot points. By stepping on them, the users ask to explore details. These contents are delivered on the walls, as the designer defines.

A.2 E.Y.E.C.U.

E.Y.E.C.U. [180, 181] is an Emotional eYe trackEr for Cultural heritage sUpport. It has been designed in the Or.C.He.S.T.R.A. project as an interface to navigate and explore works of art, mainly oriented to paintings. The proposed solution detects users facial features by means of a common webcam. The User Experience aims at being as natural as possible: the system, indeed, does not require to wear any device and starts working without a calibration step.

The interaction is as follows: the user explores the painting; in the while a webcam tracks her gaze. As the user watches a detail of the painting, corresponding to a region of interest, the system *"surprises"* the user with the projection of contents related to the observed detail. E.Y.E.C.U. has an emotional component based on mydriasis, pupil dilation, which is a well known event often related to emotional arousal. This is used as discussed by Calandra and Cutugno [182].

A.3 Caruso

One of the main features of the physical environment in which humans live is spatial dimensionality. Although 3D mainly refers to video, sounds as well carry an important spatial solution. It is essential in nature to localise sources in the space. However, the spatial component is not only detected, but can also be simulated to provide an improved experience to the listener.

Although spatialised sounds are common nowadays in games and movies, Caruso [25, 183] has been an innovative solution for the Cultural Heritage context. It is an Android

[1] https://www.w3.org/TR/scxml/ retrieved on October 2017
[2] https://www.w3.org/TR/smil/ retrieved on October 2017

app that provides a real-time dynamic experience by tracking users through their own smartphones and proposing an interaction based on 3D sound that follows users' movements. GPS position and orientation are detected to adapt a virtual 3D soundscape. In addition, in order to enhance the interaction, interactive headphones have been proposed. They are equipped with an inertial sensor, able to detect the orientation of the head, and communicate it to the smartphone via Bluetooth. The headphones' capsules are not completely isolating, so real sounds are added to the virtual part, reaching an Augmented Audio Reality. With this setup, the user is free to leave the smartphone in her pocket and interact with the system without any visual and acoustical barriers.

The user has the impression to be part of a vocal scene that explained, through animated real stories played by actors, details about Points of Interest (POI) in the historic centre of Naples. The description automatically starts close to the proper POI; the display of the smartphone can be consulted in needed to find POIs on a map.

Bibliography

[1] Yvonne Rogers, Helen Sharp, and Jenny Preece. *Interaction design: beyond human-computer interaction*. Interaction Design: Beyond Human-computer Interaction. John Wiley & Sons, 2011. ISBN 9780470665763. URL https://books.google.it/books?id=b-v_6BeCwwQC.

[2] Tsvi Kuflik, Joel Lanir, Eyal Dim, Alan Wecker, Michele Corra, Massimo Zancanaro, and Oliviero Stock. Indoor positioning in cultural heritage: Challenges and a solution. In *2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel*, pages 1–5. IEEE, Nov 2012. doi: 10.1109/EEEI.2012.6376935.

[3] Fabien Badeig, Emmanuel Adam, René Mandiau, and Catherine Garbay. Analyzing multi-agent approaches for the design of advanced interactive and collaborative systems. *Journal of Ambient Intelligence and Smart Environments*, 8(3):325–346, 2016. doi: 10.3233/AIS-160380.

[4] Silvia Marzal, Raul González-Medina, Robert Salas-Puente, Emilio Figueres, and Gabriel Garcerá. A novel locality algorithm and peer-to-peer communication infrastructure for optimizing network performance in smart microgrids. *Energies*, 10(9), 2017. ISSN 1996-1073. doi: 10.3390/en10091275. URL http://www.mdpi.com/1996-1073/10/9/1275.

[5] Carlos Rodríguez-Domínguez, Tomás Ruiz-López, Kawtar Benghazi, Manuel Noguera, and José Luis Garrido. A model-driven approach for the development of middleware technologies for ubiquitous systems. In *Intelligent Environments (IE), 2013 9th International Conference on*, pages 16–23. IEEE, 2013. doi: 10.1109/IE.2013.17.

[6] Mark Weiser and John Seely Brown. The coming age of calm technology. In *Beyond calculation*, pages 75–85. Springer, 1997. ISBN 978-1-4612-0685-9. doi: 10.1007/978-1-4612-0685-9_6. URL https://doi.org/10.1007/978-1-4612-0685-9_6.

[7] Thomas Kubitza, Alexandra Voit, Dominik Weber, and Albrecht Schmidt. An iot infrastructure for ubiquitous notifications in intelligent living environments.

In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, pages 1536–1541. ACM, 2016. ISBN 978-1-4503-4462-3. doi: 10.1145/2968219.2968545. URL http://doi.acm.org/10.1145/2968219.2968545.

[8] Yvonne Rogers. Moving on from weisers vision of calm computing: Engaging ubicomp experiences. In *International conference on Ubiquitous computing*, pages 404–421. Springer, 2006. ISBN 978-3-540-39635-2. doi: 10.1007/11853565_24. URL https://doi.org/10.1007/11853565_24.

[9] Juan C. Augusto, Vic Callaghan, Diane Cook, Achilles Kameas, and Ichiro Satoh. Intelligent environments: a manifesto. *Human-Centric Computing and Information Sciences*, 3(1):1–18, 2013.

[10] Alan Dix. Human-computer interaction. In *Encyclopedia of database systems*, pages 1327–1331. Springer, 2009. ISBN 978-0-387-39940-9. doi: 10.1007/978-0-387-39940-9_192. URL https://doi.org/10.1007/978-0-387-39940-9_192.

[11] Stuart K. Card, Allen Newell, and Thomas P. Moran. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1983. ISBN 0898592437.

[12] James H. Carlisle. Evaluating the impact of office automation on top management communication. In *Proceedings of the June 7-10, 1976, National Computer Conference and Exposition*, AFIPS '76, pages 611–616, New York, NY, USA, 1976. ACM. doi: 10.1145/1499799.1499885. URL http://doi.acm.org/10.1145/1499799.1499885.

[13] Terry Winograd. From computing machinery to interaction design. *Beyond calculation: The next fifty years of computing*, pages 149–162, 1997.

[14] Dan Saffer. *Designing for Interaction: Creating Innovative Applications and Devices*. Voices that matter. New Riders, 2010. ISBN 9780321643391. URL https://books.google.it/books?id=k28yVW3SEyYC.

[15] Sharon Oviatt. Multimodal interfaces. *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications*, 14:286–304, 2003.

[16] Euan Freeman, Graham Wilson, Dong-Bach Vo, Alex Ng, Ioannis Politis, and Stephen Brewster. in hci: Haptics, non-speech audio, and their applications. *The Handbook of Multimodal-Multisensor Interfaces: Foundations, User Modeling, and Common Modality Combinations*, page 277, 2017.

[17] Nancy J Nersessian. In the theoretician's laboratory: Thought experimenting as mental modeling. In *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, volume 1992, pages 291–301. Philosophy of Science Association, 1992.

[18] Alan Cooper, Robert Reimann, and David Cronin. *About face 3: the essentials of interaction design*. John Wiley & Sons, 2007. ISBN 9780470171356. URL https://books.google.it/books?id=xk1S1-wGZbAC.

[19] Andy Clark. *Supersizing the mind: Embodiment, action, and cognitive extension*. Oxford University Press, USA, 2008. ISBN 9780195333213.

[20] Andy Clark and David Chalmers. The extended mind. *analysis*, 58(1):7–19, 1998. ISSN 00032638, 14678284. URL http://www.jstor.org/stable/3328150.

[21] Daniel Wigdor and Dennis Wixon. *Brave NUI world: designing natural user interfaces for touch and gesture*. Elsevier, 2011.

[22] Donald A. Norman. Natural user interfaces are not natural. *interactions*, 17(3): 6–10, May 2010. ISSN 1072-5520. doi: 10.1145/1744161.1744163. URL http://doi.acm.org/10.1145/1744161.1744163.

[23] Michael A Gigante. Virtual reality: definitions, history and applications. *Virtual Reality Systems*, pages 3–14, 1993.

[24] David Drascic and Paul Milgram. Perceptual issues in augmented reality. In *Proceedings-Spie The International Society For Optical Engineering*, pages 123–134. SPIE INTERNATIONAL SOCIETY FOR OPTICAL, 1996.

[25] Daniela D'Auria, Dario Di Mauro, Davide Maria Calandra, and Francesco Cutugno. Caruso: Interactive headphones for a dynamic 3d audio application in the cultural heritage context. In *Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference on*, pages 525–528. IEEE, 2014. doi: 10.1109/IRI.2014.7051934.

[26] William W Gaver. Auditory icons: Using sound in computer interfaces. *Human-computer interaction*, 2(2):167–177, 1986.

[27] Meera M Blattner, Denise A Sumikawa, and Robert M Greenberg. Earcons and icons: Their structure and common design principles. *Human–Computer Interaction*, 4(1):11–44, 1989.

[28] Kenneth P Fishkin. A taxonomy for and analysis of tangible interfaces. *Personal and Ubiquitous Computing*, 8:347–358, 2004.

[29] Mark T. Marshall, Nick Dulake, Luigina Ciolfi, Daniele Duranti, Hub Kockelkorn, and Daniela Petrelli. Using tangible smart replicas as controls for an interactive museum exhibition. In *Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 159–167. ACM, 2016.

[30] Steve Benford, Andy Crabtree, Martin Flintham, Chris Greenhalgh, Boriana Koleva, Matt Adams, Nick Tandavanitj, Ju Row Farr, Gabriella Giannachi, and Irma Lindt. Creating the spectacle: Designing interactional trajectories through spectator interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 18(3):11, 2011. doi: 10.1145/1993060.1993061. URL http://doi.acm.org/10.1145/1993060.1993061.

[31] Luigina Ciolfi. Supporting affective experiences of place through interaction design. *CoDesign*, 3(S1):183–198, 2007. doi: 10.1080/15710880701309308. URL http://dx.doi.org/10.1080/15710880701309308.

[32] Robyn Taylor, John Bowers, Bettina Nissen, Gavin Wood, Qasim Chaudhry, Peter Wright, Lindsey Bruce, Sarah Glynn, Helen Mallinson, and Roy Bearpark. Making magic: designing for open interactions in museum settings. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*, pages 313–322. ACM, 2015. doi: 10.1145/2757226.2757241. URL http://doi.acm.org/10.1145/2757226.2757241.

[33] Julie R. Williamson, Euan Freeman, and Stephen Brewster. Levitate: Interaction with floating particle displays. In *Proceedings of the 6th ACM International Symposium on Pervasive Displays*, PerDis '17, pages 24:1–24:2, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5045-7. doi: 10.1145/3078810.3084347. URL http://doi.acm.org/10.1145/3078810.3084347.

[34] Donald A. Norman and Stephen W. Draper. *User Centered System Design; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1986. ISBN 0898597811.

[35] Douglas Schuler and Aki Namioka. *Participatory design: Principles and practices*. CRC Press, 1993. ISBN 9780805809510. URL https://books.google.it/books?id=pWOEk6Sk4YkC.

[36] Johannes Wagner, Florian Lingenfelser, Tobias Baur, Ionut Damian, Felix Kistler, and Elisabeth André. The social signal interpretation (ssi) framework: multimodal signal processing and recognition in real-time. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 831–834. ACM, 2013.

[37] Mark Weiser. The computer for the 21st century. *Scientific american*, 265(3): 94–104, 1991.

[38] Hideyuki Nakashima, Hamid Aghajan, and Juan Carlos Augusto. *Handbook of ambient intelligence and smart environments*. Springer Science & Business Media, 2009.

[39] Diane Cook and Sajal Das. *Smart environments: Technology, protocols and applications*, volume 43. John Wiley & Sons, 2004.

[40] Juan Carlos Augusto and Paul McCullagh. Ambient intelligence: Concepts and applications. *Computer Science and Information Systems*, 4(1):1–27, 2007.

[41] H Raffler. Other perspectives on ambient intelligence. *Password Mag*, 23:12–13, 2006.

[42] Juan Carlos Augusto. Ambient intelligence: the confluence of ubiquitous/pervasive computing and artificial intelligence. *Intelligent Computing Everywhere*, pages 213–234, 2007.

[43] Ricardo Costa, Davide Carneiro, Paulo Novais, Luís Lima, José Machado, Alberto Marques, and José Neves. Ambient assisted living. In *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*, pages 86–94, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-540-85867-6. doi: 10.1007/978-3-540-85867-6_10. URL https://doi.org/10.1007/978-3-540-85867-6_10.

[44] Davide Bacciu, Stefano Chessa, Erina Ferro, Luigi Fortunati, Claudio Gallicchio, Davide La Rosa, Miguel Llorente, Alessio Micheli, Filippo Palumbo, Oberdan Parodi, et al. Detecting socialization events in ageing people: the experience of the doremi project. In *Intelligent Environments (IE), 2016 12th International Conference on*, pages 132–135. IEEE, 2016.

[45] Michael H Coen et al. Design principles for intelligent environments. In *AAAI/IAAI*, pages 547–554, 1998.

[46] S Tomic, A Fensel, M Schwanzer, M Veljovic, and M Stefanovic. Semantics for energy efficiency in smart home environments. applied semantic technologies: Using semantics. *Intelligent Information Processing*, 2011.

[47] Shiu Kumar. Ubiquitous smart home system using android application. *International Journal of Computer Networks & Communications*, 6(1):33–43, 2014. doi: 10.5121/ijcnc.2014.6103.

[48] Rajeev Piyare. Internet of things: ubiquitous home control and monitoring system using android based smart phone. *International Journal of Internet of Things*, 2 (1):5–11, 2013.

[49] Kenzhegali Nurgaliyev, Dario Di Mauro, Nawaz Khan, and Juan Carlos Augusto. Improved multi-user interaction in a smart environment through a preference-based conflict resolution virtual assistant. In *Proceedings of 13th International Conference on Intelligent Environments (IE17)*. Institute of Electrical and Electronics Engineers (IEEE), 2017. In Press.

[50] Florian Block, Albrecht Schmidt, Nicolas Villar, and Hans Gellersen. Towards a playful user interface for home entertainment systems. In *Proceedings of the European Symposium on Ambient Intelligence*. Springer, 2004.

[51] Stefano Ferilli, Berardina De Carolis, Andrea Pazienza, Floriana Esposito, and Domenico Redavid. An agent architecture for adaptive supervision and control of smart environments. In *Pervasive and Embedded Computing and Communication Systems (PECCS), 2015 International Conference on*, pages 1–8. IEEE, 2015.

[52] Weixian Li, Thillainathan Logenthiran, WL Woo, Van-Tung Phan, and Dipti Srinivasan. Implementation of demand side management of a smart home using multi-agent system. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 2028–2035. IEEE, July 2016. doi: 10.1109/CEC.2016.7744037.

[53] Sergio Pajares Ferrando and Eva Onaindia. Context-aware multi-agent planning in intelligent environments. *Information Sciences*, 227:22–42, 2013. doi: http://dx.doi.org/10.1016/j.ins.2012.11.021. URL http://www.sciencedirect.com/science/article/pii/S0020025512007748.

[54] Houda Haiouni and Ramdane Maamri. Context-aware planning for intelligent environments. In *Advanced Aspects of Software Engineering (ICAASE), 2016 International Conference on*, pages 1–5. IEEE, Oct 2016. doi: 10.1109/ICAASE.2016.7843857.

[55] Ilche Georgievski and Marco Aiello. Automated planning for ubiquitous computing. *ACM Computing Surveys (CSUR)*, 49(4):63, 2016. doi: 10.1145/3004294.

[56] Dante I Tapia, Ajith Abraham, Juan M Corchado, and Ricardo S Alonso. Agents and ambient intelligence: case studies. *Journal of Ambient Intelligence and Humanized Computing*, 1(2):85–93, 2010. ISSN 1868-5145. doi: 10.1007/s12652-009-0006-2. URL https://doi.org/10.1007/s12652-009-0006-2.

[57] Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood. *Developing multi-agent systems with JADE*, volume 7. John Wiley & Sons, 2007.

[58] Francesco Benzi, Daniela Fogli, and Giovanni Guida. Towards natural interaction with smart homes. In *Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter*, CHItaly '17, pages 10:1–10:6. ACM, 2017. ISBN 978-1-4503-5237-6. doi: 10.1145/3125571.3125581. URL http://doi.acm.org/10.1145/3125571.3125581.

[59] De Sheng Liu, Feng Xin Yan, and Shu Dong Sun. Research on the applications of internet of things technology in cultural heritage preservation. In *Advanced Materials Research*, volume 476, pages 371–374. Trans Tech Publ, 2012.

[60] Chia-Chen Chen and Tien-Chi Huang. Learning in a u-museum: Developing a context-aware ubiquitous learning environment. *Computers & Education*, 59(3): 873–883, 2012.

[61] Stefania Costantini, Leonardo Mostarda, Arianna Tocchio, and Panagiota Tsintza. Dalica: Agent-based ambient intelligence for cultural-heritage scenarios. *IEEE Intelligent Systems*, 23(2), 2008.

[62] Angelo Chianese and Francesco Piccialli. Improving user experience of cultural environment through iot: the beauty or the truth case study. In *Intelligent Interactive Multimedia Systems and Services*, pages 11–20. Springer, 2015.

[63] Angelo Chianese, Francesco Piccialli, and Jay E Jung. The internet of cultural things: Towards a smart cultural heritage. In *2016 12th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, pages 493–496. IEEE, 2016. doi: 10.1109/SITIS.2016.83.

[64] Stefano Alletto, Rita Cucchiara, Giuseppe Del Fiore, Luca Mainetti, Vincenzo Mighali, Luigi Patrono, and Giuseppe Serra. An indoor location-aware system for an iot-based smart museum. *IEEE Internet of Things Journal*, 3(2):244–253, 2016.

[65] Alan Wecker, Tsvi Kuflik, Joel Lanir, and Oliviero Stock. Evaluating mobile projectors as a shared display option for small groups. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 2543–2548. ACM, 2012. ISBN 978-1-4503-1016-1. doi: 10.1145/2212776.2223833. URL http://doi.acm.org/10.1145/2212776.2223833.

[66] Joel Lanir, Alan J Wecker, Tsvi Kuflik, and Yasmin Felberbaum. Shared mobile displays: an exploratory study of their use in a museum setting. *Personal and Ubiquitous Computing*, 20(4):635–651, 2016.

[67] Dmitry Korzun, Aleksey Varfolomeyev, Svetlana Yalovitsyna, and Valentina Volokhova. Semantic infrastructure of a smart museum: toward making cultural

heritage knowledge usable and creatable by visitors and professionals. *Personal and Ubiquitous Computing*, 21(2):345–354, 2016. ISSN 1617-4917. doi: 10.1007/s00779-016-0996-7. URL https://doi.org/10.1007/s00779-016-0996-7.

[68] Oliviero Stock, Massimo Zancanaro, Paolo Busetta, Charles Callaway, Antonio Krüger, Michael Kruppa, Tsvi Kuflik, Elena Not, and Cesare Rocchi. Adaptive, intelligent presentation of information for the museum visitor in peach. *User Modeling and User-Adapted Interaction*, 17(3):257–304, Jul 2007. ISSN 1573-1391. doi: 10.1007/s11257-007-9029-6.

[69] Angeliki Antoniou, Akrivi Katifori, Maria Roussou, Maria Vayanou, Manolis Karvounis, Marialena Kyriakidi, and Laia Pujol-Tost. Capturing the visitor profile for a personalized mobile museum experience: an indirect approach. In *UMAP (Extended Proceedings)*, 2016.

[70] Daniela Petrelli, Mark Marshall, Elena Not, Massimo Zancanaro, Adriano Venturini, Dario Cavada, Thomas Kubitza, Albrecht Schmidt, Martin Risseeuw, and Dick van Dijk. mesch: Internet of things and cultural heritage. *SCIRES-IT-SCIentific RESearch and Information Technology*, 6(1):15–22, 2016.

[71] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010. ISSN 1389-1286. doi: https://doi.org/10.1016/j.comnet.2010.05.010. URL http://www.sciencedirect.com/science/article/pii/S1389128610001568.

[72] Yuichi Kawamoto, Hiroki Nishiyama, Nei Kato, Naoko Yoshimura, and Shinichi Yamamoto. Internet of things (iot): present state and future prospects. *IEICE TRANSACTIONS on Information and Systems*, 97(10):2568–2575, 2014. ISSN 1745-1361.

[73] Vittorio Maniezzo and Antonella Carbonaro. Ant colony optimization: An overview. In *Essays and Surveys in Metaheuristics*, pages 469–492, Boston, MA, 2002. Springer US. ISBN 978-1-4615-1507-4. doi: 10.1007/978-1-4615-1507-4_21. URL https://doi.org/10.1007/978-1-4615-1507-4_21.

[74] Sofiane Hamrioui and Pascal Lorenz. Bio inspired routing algorithm and efficient communications within iot. *IEEE Network*, 31(5):74–79, 2017. ISSN 0890-8044. doi: 10.1109/MNET.2017.1600282.

[75] C.K. Toh. *Wireless Atm and Ad-Hoc Networks: Protocols and Architectures*. Kluwer Academic Publisherb Group, 1997. ISBN 9780792398226. URL https://books.google.com.sg/books?id=0hMfAQAAIAAJ.

[76] Hiroki Nishiyama, Masaya Ito, and Nei Kato. Relay-by-smartphone: realizing multihop device-to-device communications. *IEEE Communications Magazine*, 52 (4):56–65, 2014.

[77] Thomas Kubitza and Albrecht Schmidt. Towards a toolkit for the rapid creation of smart environments. In *End-User Development: 5th International Symposium, IS-EUD 2015, Madrid, Spain, May 26-29, 2015. Proceedings*, pages 230–235, Cham, 2015. Springer International Publishing. ISBN 978-3-319-18425-8. doi: 10.1007/978-3-319-18425-8_21. URL https://doi.org/10.1007/978-3-319-18425-8_21.

[78] Joel Lanir, Tsvi Kuflik, Julia Sheidin, Nisan Yavin, Kate Leiderman, and Michael Segal. Visualizing museum visitors' behavior: Where do they go and what do they do there? *Personal and Ubiquitous Computing*, pages 1–14, 2016. ISSN 1617-4917. doi: 10.1007/s00779-016-0994-9. URL http://dx.doi.org/10.1007/s00779-016-0994-9.

[79] Yuji Yoshimura, Stanislav Sobolevsky, Carlo Ratti, Fabien Girardin, Juan Pablo Carrascal, Josep Blat, and Roberta Sinatra. An analysis of visitors' behavior in the louvre museum: A study using bluetooth data. *Environment and Planning B: Planning and Design*, 41(6):1113–1131, 2014.

[80] Beverly Serrell. Paying attention: The duration and allocation of visitors' time in museum exhibitions. *Curator: The museum journal*, 40(2):108–125, 1997. ISSN 2151-6952. doi: 10.1111/j.2151-6952.1997.tb01292.x. URL http://dx.doi.org/10.1111/j.2151-6952.1997.tb01292.x.

[81] Steven S. Yalowitz and Kerry Bronnenkant. Timing and tracking: Unlocking visitor behavior. *Visitor Studies*, 12(1):47–64, 2009.

[82] Hans-Joachim Klein. Tracking visitor circulation in museum settings. *Environment and Behavior*, 25(6):782–800, 1993.

[83] Joel Lanir, Peter Bak, and Tsvi Kuflik. Visualizing proximity-based spatiotemporal behavior of museum visitors using tangram diagrams. In *Computer Graphics Forum*, volume 33, pages 261–270, 2014.

[84] Angelo Chianese, Fiammetta Marulli, Francesco Piccialli, Paolo Benedusi, and Jai E Jung. An associative engines based approach supporting collaborative analytics in the internet of cultural things. *Future generation computer systems*, 66: 187–198, 2017.

[85] Joel Lanir, Tsvi Kuflik, Eyal Dim, Alan J. Wecker, and Oliviero Stock. The influence of a location-aware mobile guide on museum visitors' behavior. *Interacting with Computers*, 25(6):443–460, 2013. doi: 10.1093/iwc/iwt002.

[86] Gema Bello-Orgaz, Jason J Jung, and David Camacho. Social big data: Recent achievements and new challenges. *Information Fusion*, 28:45–59, 2016. doi: http://dx.doi.org/10.1016/j.inffus.2015.08.005. URL http://www.sciencedirect.com/science/article/pii/S1566253515000780.

[87] Angelo Chianese and Francesco Piccialli. Designing a smart museum: When cultural heritage joins iot. In *Next Generation Mobile Apps, Services and Technologies (NGMAST), 2014 Eighth International Conference on*, pages 300–306. IEEE, 2014. doi: 10.1109/NGMAST.2014.21.

[88] Harry Chen, Tim Finin, and Anupam Joshi. The soupa ontology for pervasive computing. In *Ontologies for agents: Theory and experiences*, pages 233–258. Springer, 2005.

[89] Nayat Sánchez-Pi and José Manuel Molina. A centralized approach to an ambient assisted living application: An intelligent home. In *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living: 10th International Work-Conference on Artificial Neural Networks, IWANN 2009 Workshops, Salamanca, Spain, June 10-12, 2009. Proceedings, Part II*, pages 706–709, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-02481-8. doi: 10.1007/978-3-642-02481-8_106.

[90] Davide Bacciu, Stefano Chessa, Claudio Gallicchio, Alessio Micheli, Erina Ferro, Luigi Fortunati, Filippo Palumbo, Oberdan Parodi, Federico Vozzi, Sten Hanke, et al. Smart environments and context-awareness for lifestyle management in a healthy active ageing framework. In *Portuguese Conference on Artificial Intelligence*, pages 54–66. Springer, 2015.

[91] James Dooley, Martin Henson, Vic Callaghan, Hani Hagras, Daniyal Al-Ghazzawi, Areej Malibari, Mohammed Al-Haddad, and Abdullah Al-Malaise Al-Ghamdi. A formal model for space based ubiquitous computing. In *Intelligent Environments (IE), 2011 7th International Conference on*, pages 74–79. IEEE, 2011.

[92] Falko Dressler and Ozgur B. Akan. A survey on bio-inspired networking. *Computer Networks*, 54(6):881–900, 2010. ISSN 1389-1286. doi: http://dx.doi.org/10.1016/j.comnet.2009.10.024. URL http://www.sciencedirect.com/science/article/pii/S1389128610000241.

[93] Oladayo Bello and Sherali Zeadally. Intelligent device-to-device communication in the internet of things. *IEEE Systems Journal*, 10(3):1172–1182, 2016. ISSN 1932-8184. doi: 10.1109/JSYST.2014.2298837.

[94] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *International Symposium on Handheld and Ubiquitous Computing*, pages 304–307. Springer, 1999.

[95] Mirco Musolesi and Cecilia Mascolo. Evaluating context information predictability for autonomic communication. In *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, pages 495–499. IEEE Computer Society, 2006.

[96] Dario Di Mauro and Francesco Cutugno. A framework for interaction design in intelligent environments. In *Proceedings of 12th International Conference on Intelligent Environments (IE)*, pages 246–249, Sept 2016. doi: 10.1109/IE.2016.56.

[97] Tomás Ruiz-López, José Luis Garrido, Sam Supakkul, and Lawrence Chung. A pattern approach to dealing with nfrs in ubiquitous systems. In *CAiSE Forum*, pages 25–32, 2013.

[98] James A Landay and Gaetano Borriello. Design patterns for ubiquitous computing. *Computer*, 36(8):93–95, 2003. ISSN 0018-9162. doi: 10.1109/MC.2003.1220589.

[99] David F McAllister, Robert D Rodman, Donald L Bitzer, and Andrew S Freeman. Lip synchronization of speech. In *Audio-Visual Speech Processing: Computational & Cognitive Science Approaches*, 1997.

[100] Pierre Lison. A hybrid approach to dialogue management based on probabilistic rules. *Computer Speech & Language*, 34(1):232–255, 2015.

[101] Ramón López-Cózar and Zoraida Callejas. Multimodal dialogue for ambient intelligence and smart environments. *Handbook of ambient intelligence and smart environments*, pages 559–579, 2010. doi: 10.1007/978-0-387-93808-0_21. URL https://doi.org/10.1007/978-0-387-93808-0_21.

[102] Wolfgang Wahlster. Smartkom: Fusion and fission of speech, gestures, and facial expressions. In *Proceedings of the 1st International Workshop on Man-Machine Symbiotic Systems*, pages 213–225, 2002.

[103] Norbert Reithinger, Christoph Lauer, and Laurent Romary. Miamm – multidimensional information access using multiple modalities. *Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems*, page 137, 2002.

[104] I. Fette and A. Melnikov. The websocket protocol, 2011. URL https://www.rfc-editor.org/info/rfc6455. [Online; RFC 6455 posted December-2011].

[105] Gabriel Guerrero-Contreras, José Luis Garrido, María José Rodríguez Fórtiz, Gregory MP O'Hare, and Sara Balderas-Díaz. Impact of transmission communication protocol on a self-adaptive architecture for dynamic network environments. In *World Conference on Information Systems and Technologies*, pages 115–124. Springer, 2017. ISBN 978-3-319-56538-5. doi: 10.1007/978-3-319-56538-5_12. URL https://doi.org/10.1007/978-3-319-56538-5_12.

[106] Carsten Bormann, Angelo P Castellani, and Zach Shelby. Coap: An application protocol for billions of tiny internet nodes. *IEEE Internet Computing*, 16(2):62–67, 2012.

[107] Patrick Th Eugster, Pascal A Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM computing surveys (CSUR)*, 35 (2):114–131, 2003.

[108] D. Thangavel, X. Ma, A. Valera, H. X. Tan, and C. K. Y. Tan. Performance evaluation of mqtt and coap via a common middleware. In *IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 1–6, April 2014. doi: 10.1109/ISSNIP.2014.6827678.

[109] John O'Hara. Toward a commodity enterprise middleware. *Queue*, 5(4):48–55, May 2007. ISSN 1542-7730. doi: 10.1145/1255421.1255424. URL http://doi.acm.org/10.1145/1255421.1255424.

[110] Francesco Barile, Davide M. Calandra, Antonio Caso, Daniela D'Auria, Dario Di Mauro, Francesco Cutugno, and Silvia Rossi. Ict solutions for the or.c.he.s.t.r.a. project: From personalized selection to enhanced fruition of cultural heritage data. In *Signal-Image Technology and Internet-Based Systems (SITIS), 2014 Tenth International Conference on*, pages 501–507. IEEE, Nov 2014. doi: 10.1109/SITIS.2014.12.

[111] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[112] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3:225–331, 2009. ISSN 1554-0669. doi: 10.1561/1500000016. URL http://dx.doi.org/10.1561/1500000016.

[113] Mihai Surdeanu, Dan I. Moldovan, and Sanda M. Harabagiu. Performance analysis of a distributed question/answering system. *IEEE Transactions on Parallel and Distributed Systems*, 13(6):579–596, 2002.

[114] Paulo Carreira, Sílvia Resendes, and André C Santos. Towards automatic conflict detection in home and building automation systems. *Pervasive and Mobile Computing*, 12:37–57, 2014. ISSN 1574-1192. doi: http://dx.doi.org/10.1016/j.pmcj.2013.06.001. URL http://www.sciencedirect.com/science/article/pii/S1574119213000783.

[115] Evi Syukur, Seng Wai Loke, and Peter Stanski. Methods for policy conflict detection and resolution in pervasive computing environments. In *Policy Management for Web workshop in conjunction with WWW2005 Conference*, 2005.

[116] Mike P Papazoglou and Willem-Jan Heuvel. Service oriented architectures: approaches, technologies and research issues. *The VLDB JournalThe International Journal on Very Large Data Bases*, 16(3):389–415, 2007. doi: 10.1007/s00778-007-0044-3. URL http://dx.doi.org/10.1007/s00778-007-0044-3.

[117] Insuk Park, Kyungmin Lee, Dongman Lee, Soon J Hyun, and Hee Yong Yoon. A dynamic context conflict resolution scheme for group-aware ubiquitous computing environments. In *Proceedings of the 1st International Workshop on Personalized Context Modeling and Management for UbiComp Applications (ubiPCMM 2005)*, pages 42–47, 2005.

[118] Verena Tuttlies, Gregor Schiele, and Christian Becker. Comity-conflict avoidance in pervasive computing environments. In *On the move to meaningful internet systems 2007: OTM 2007 workshops*, pages 763–772. Springer, 2007.

[119] W Alshabi, S Ramaswamy, M Itmi, and H Abdulrab. Coordination, cooperation and conflict resolution in multi-agent systems. *Innovations and advanced techniques in computer and information sciences and engineering*, pages 495–500, 2007. doi: 10.1007/978-1-4020-6268-1_87. URL https://doi.org/10.1007/978-1-4020-6268-1_87.

[120] Silvia Rossi, Claudia Di Napoli, Francesco Barile, and Luca Liguori. A multi-agent system for group decision support based on conflict resolution styles. In *International Workshop on Conflict Resolution in Decision Making*, pages 134–148. Springer, 2016.

[121] Witold Jacak and Karin Pröll. Heuristic approach to conflict problem solving in an intelligent multiagent system. In *International Conference on Computer Aided Systems Theory*, pages 772–779. Springer, 2007. doi: 10.1007/978-3-540-75867-9_97. URL https://doi.org/10.1007/978-3-540-75867-9_97.

[122] Thinagaran Perumal, Md Nasir Sulaiman, Soumya Kanti Datta, Thinaharan Ramachandran, and Chui Yew Leong. Rule-based conflict resolution framework for

internet of things device management in smart home environment. In *Consumer Electronics, 2016 IEEE 5th Global Conference on*, pages 1–2. IEEE, 2016.

[123] Md Kamrul Hasan, Kim Anh, Lenin Mehedy, Young-Koo Lee, and Sungyoung Lee. Conflict resolution and preference learning in ubiquitous environment. In *International Conference on Intelligent Computing*, pages 355–366. Springer, 2006.

[124] Daniel Retkowitz and Sven Kulle. Dependency management in smart homes. In *IFIP International Conference on Distributed Applications and Interoperable Systems*, pages 143–156. Springer, 2009.

[125] Gonzalo Huerta-Canepa and Dongman Lee. A multi-user ad-hoc resource manager for smart spaces. In *World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a*, pages 1–6. IEEE, 2008.

[126] Gabriel Guerrero-Contreras, José Luis Garrido, Sara Balderas-Díaz, and Carlos Rodríguez-Domínguez. Consistent management of context information in ubiquitous systems. In *Internet and Distributed Computing Systems: 7th International Conference, IDCS 2014, Calabria, Italy, September 22-24, 2014. Proceedings*, pages 184–193. Springer International Publishing, 2014. ISBN 978-3-319-11692-1. doi: 10.1007/978-3-319-11692-1_16. URL https://doi.org/10.1007/978-3-319-11692-1_16.

[127] Prasenjit Choudhury, Anirban Sarkar, and Narayan C Debnath. Deployment of service oriented architecture in manet: A research roadmap. In *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*, pages 666–670. IEEE, 2011. doi: 10.1109/INDIN.2011.6034957.

[128] Gabriel Guerrero-Contreras, Carlos Rodríguez-Domínguez, Sara Balderas-Díaz, and José Luis Garrido. Dynamic replication and deployment of services in mobile environments. In *New Contributions in Information Systems and Technologies: Volume 1*, pages 855–864. Springer International Publishing, 2015. ISBN 978-3-319-16486-1. doi: 10.1007/978-3-319-16486-1_85. URL https://doi.org/10.1007/978-3-319-16486-1_85.

[129] Amirreza Masoumzadeh, Morteza Amini, and Rasool Jalili. Conflict detection and resolution in context-aware authorization. In *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, volume 1, pages 505–511, May 2007. doi: 10.1109/AINAW.2007.127.

[130] Jéssica S. Santos, Jean O. Zahn, Eduardo A. Silvestre, Viviane T. Silva, and Wamberto W. Vasconcelos. Detection and resolution of normative conflicts in multi-agent systems: a literature survey. *Autonomous Agents and Multi-Agent*

Systems, 2017. ISSN 1573-7454. doi: 10.1007/s10458-017-9362-z. URL https://doi.org/10.1007/s10458-017-9362-z.

[131] Antonis Bikakis, Grigoris Antoniou, and Panayiotis Hasapis. Strategies for contextual reasoning with conflicts in ambient intelligence. *Knowledge and Information Systems*, 27(1):45–84, Apr 2011. ISSN 0219-3116. doi: 10.1007/s10115-010-0293-0. URL https://doi.org/10.1007/s10115-010-0293-0.

[132] Pierre Lison and Casey Kennington. Opendial: A toolkit for developing spoken dialogue systems with probabilistic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Demonstrations)*, pages 67–72, Berlin, Germany, 2016. Association for Computational Linguistics.

[133] John H Falk and Lynn D Dierking. *The museum experience revisited*. Routledge, 2016.

[134] Eliséo Véron and Martine Levasseur. *Ethnographie de l'exposition: l'espace, le corps et le sens*. Centre Georges Pompidou, Bibliothèque publique d'information, 1989.

[135] Flavia Sparacino. The museum wearable: Real-time sensor-driven understanding of visitors' interests for personalized visually-augmented museum experiences. In *Proceedings of: Museums and the Web (MW 2002)*. ERIC, 2002.

[136] Marek Hatala and Ron Wakkary. Ontology-based user modeling in an augmented audio reality system for museums. *User Modeling and User-Adapted Interaction*, 15(3):339–380, 2005.

[137] Tsvi Kuflik and Eyal Dim. Early detection of pairs of visitors by using a museum triage. In *Proceedings of the Annual Conference of Museums and the Web*, 2013. URL http://mw2013.museumsandtheweb.com/paper/early-detection-of-pairs-of-visitors-by-using-a-museum-triage.

[138] Massimo Zancanaro, Tsvi Kuflik, Zvi Boger, Dina Goren-Bar, and Dan Goldwasser. Analyzing museum visitors' behavior patterns. In *User Modeling 2007: 11th International Conference, UM 2007, Corfu, Greece, July 25-29, 2007. Proceedings*, pages 238–246, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-73078-1. doi: 10.1007/978-3-540-73078-1_27.

[139] Tsvi Kuflik, Zvi Boger, and Massimo Zancanaro. Analysis and prediction of museum visitors behavioral pattern types. In *Ubiquitous Display Environments*, pages 161–176. Springer, 2012.

[140] Liliana Ardissono, Tsvi Kuflik, and Daniela Petrelli. Personalization in cultural heritage: the road travelled and the one ahead. *User Modeling and User-Adapted Interaction*, 22(1):73–99, Apr 2012. ISSN 1573-1391. doi: 10.1007/s11257-011-9104-x.

[141] Alan Cooper. *The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity.* Sams Indianapolis, IN, USA, 2004. ISBN 0-672-32614-0.

[142] Larry L Constantine and Lucy AD Lockwood. foruse newsletter# 26, 2002.

[143] Laura A. Maye, Fiona E. McDermott, Luigina Ciolfi, and Gabriela Avram. Interactive exhibitions design: What can we learn from cultural heritage professionals? In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, pages 598–607, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2542-4. doi: 10.1145/2639189.2639259. URL http://doi.acm.org/10.1145/2639189.2639259.

[144] Fabian Bohnert and Ingrid Zukerman. Non-intrusive personalisation of the museum experience. In *UMAP*, pages 197–209. Springer, 2009.

[145] Francesco Cutugno, Davide M. Calandra, Dario Di Mauro, and Antonio Origlia. Augmented reality without barriers: Dematerializing interfaces in cultural heritage applications. *Artificial Intelligence for Cultural Heritage*, pages 85–117, 2016.

[146] Joel Lanir, Tsvi Kuflik, Nisan Yavin, Kate Leiderman, and Michael Segal. Visualizing museum visitors' behavior. In *the first Workshop on Advanced Visual Interfaces for Cultural Heritage*, 2016. URL http://avich-16.di.unito.it/pdf/AVI-CH_2016_paper_2.pdf.

[147] David Harel. Statecharts: A visual formalism for complex systems. *Science of computer programming*, 8(3):231–274, 1987.

[148] Doron Drusinsky and David Harel. Using statecharts for hardware description and synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 8(7):798–807, 1989. ISSN 0278-0070. doi: 10.1109/43.31537.

[149] Michael Von der Beeck. A comparison of statecharts variants. In *Formal techniques in real-time and fault-tolerant systems*, pages 128–148. Springer, 1994.

[150] Mathias Kühn and Peter Forbrig. Applying ui patterns for modeling dialogs. In *PAME/VOLT@ MODELS*, pages 13–17, 2016.

[151] Pedro Sánchez, Bárbara Álvarez, Ramón Martínez, and Andrés Iborra. Embedding statecharts into teleo-reactive programs to model interactions between agents. *Journal of Systems and Software*, 2017. ISSN 0164-1212. doi: https://doi.org/10.1016/j.jss.2017.05.081. URL http://www.sciencedirect.com/science/article/pii/S0164121217301024.

[152] Jim Barnett, Rahul Akolkar, RJ Auburn, Michael Bodell, Daniel C Burnett, Jerry Carter, Scott McGlashan, Torbjörn Lager, Mark Helbing, Rafah Hosn, et al. State chart xml (scxml): State machine notation for control abstraction. *W3C working draft*, 2007.

[153] Dario Di Mauro, Davide Maria Calandra, Antonio Origlia, and Francesco Cutugno. Past: Human tracking and gestures recognition for flexible virtual environments management. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pages 331–347. Springer International Publishing, 2016. ISBN 978-3-319-40651-0. doi: 10.1007/978-3-319-40651-0_27. URL https://doi.org/10.1007/978-3-319-40651-0_27.

[154] James F Allen and George Ferguson. Actions and events in interval temporal logic. *Journal of logic and computation*, 4(5):531–579, 1994.

[155] Marc Davies, Vic Callaghan, and Michael Gardner. Towards a mixed reality intelligent campus. *IET Conference Proceedings*, 2008. URL http://digital-library.theiet.org/content/conferences/10.1049/cp_20081175.

[156] Anasol Peña Ríos, Vic Callaghan, Michael Gardner, and Mohammed J Alhaddad. Using mixed-reality to develop smart environments. In *Intelligent Environments (IE), 2014 International Conference on*, pages 182–189. IEEE, 2014. doi: 10.1109/IE.2014.34.

[157] Ryan Marcotte and Howard J. Hamilton. Behavior trees for modelling artificial intelligence in games: A tutorial. *The Computer Games Journal*, Jul 2017. ISSN 2052-773X. doi: 10.1007/s40869-017-0040-9.

[158] Kingkarn Sookhanaphibarn and Ruck Thawonmas. A movement data analysis and synthesis tool for museum visitors' behaviors. In *Advances in Multimedia Information Processing - PCM 2009: 10th Pacific Rim Conference on Multimedia, Bangkok, Thailand, December 15-18, 2009 Proceedings*, pages 144–154, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-10467-1. doi: 10.1007/978-3-642-10467-1_12.

[159] Alan J. Wecker, Tsvi Kuflik, and Oliviero Stock. Dynamic personalization based on mobile behavior: From personality to personalization: a blueprint. In *Proceedings*

of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct, MobileHCI '16, pages 978–983, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4413-5. doi: 10.1145/2957265.2962645. URL http://doi.acm.org/10.1145/2957265.2962645.

[160] Christian Kray and Jörg Baus. A survey of mobile guides. In *Workshop on HCI in mobile guides, 5th Int. Symposium on HCI with Mobile Devices and Services. Udine, Italy*, 2003.

[161] Wil Van Der Aalst, Arya Adriansyah, Ana Karla Alves De Medeiros, Franco Arcieri, Thomas Baier, Tobias Blickle, Jagadeesh Chandra Bose, Peter van den Brand, Ronald Brandtjen, Joos Buijs, et al. Process mining manifesto. In *International Conference on Business Process Management*, pages 169–194. Springer, 2011. ISBN 978-3-642-28108-2. doi: 10.1007/978-3-642-28108-2_19. URL https://doi.org/10.1007/978-3-642-28108-2_19.

[162] Alessandro Bollo and Luca Dal Pozzolo. Analysis of visitor behaviour inside the museum: an empirical study. In *Proceedings of the 8th international conference on arts and cultural management*, 2005.

[163] Eilean Hooper-Greenhill. Studying visitors. *A companion to museum studies*, pages 362–376, 2006.

[164] Jörg Baus, Antonio Krüger, and Wolfgang Wahlster. A resource-adaptive mobile navigation system. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 15–22. ACM, 2002.

[165] Tsvi Kuflik, Charles Callaway, Dina Goren-Bar, Cesare Rocchi, Oliviero Stock, and Massimo Zancanaro. Non-intrusive user modeling for a multimedia museum visitors guide system. *User Modeling 2005*, pages 152–152, 2005.

[166] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967. URL https://books.google.it/books?id=IC4Ku_7dBFUC.

[167] Wil Van Der Aalst. *Process Mining Discovery, Conformance and Enhancement of Business Processes*. Springer Heidelberg Dordrecht London New York, 2011. ISBN 978-3-642-19344-6.

[168] James Lyle Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice Hall PTR, 1981. ISBN 0136619835. URL http://portal.acm.org/citation.cfm?id=539513.

[169] Wil MP Van der Aalst. The application of petri nets to workflow management. *Journal of circuits, systems, and computers*, 8(01):21–66, 1998. doi: 10.1142/S0218126698000043. URL http://www.worldscientific.com/doi/abs/10.1142/S0218126698000043.

[170] Wil van der Aalst. Making work flow: On the application of petri nets to business process management. *Application and Theory of Petri Nets 2002*, pages 1–22, 2002. doi: 10.1007/3-540-48068-4_1. URL https://doi.org/10.1007/3-540-48068-4_1.

[171] B Van Dongen, A Alves de Medeiros, and Lijie Wen. Process mining: Overview and outlook of petri net discovery algorithms. *Transactions on Petri Nets and Other Models of Concurrency II*, pages 225–242, 2009. doi: 10.1007/978-3-642-00899-3_13. URL https://doi.org/10.1007/978-3-642-00899-3_13.

[172] Wil MP Van Der Aalst, Hajo A Reijers, and Minseok Song. Discovering social networks from event logs. *Computer Supported Cooperative Work (CSCW)*, 14(6): 549–593, 2005. ISSN 1573-7551. doi: 10.1007/s10606-005-9005-9. URL https://doi.org/10.1007/s10606-005-9005-9.

[173] Antonio Origlia, Maria Laura Chiacchio, Dario Di Mauro, and Francesco Cutugno. Why so serious? raising curiosity towards cultural heritage with playful games. In *AI* CH@ AI* IA*, pages 18–25, 2016.

[174] Antonio Origlia, Dario Di Mauro, Maria Laura Chiacchio, and Francesco Cutugno. Establishing a theoretical background for a museum-centric entertainment system. In *Proceedings of the 1st Workshop on Games-Human Interaction (GHItaly '17)*, September 2017.

[175] Norman M. Fraser and G. Nigel Gilbert. Simulating speech systems. *Computer Speech & Language*, 5(1):81–99, 1991. ISSN 0885-2308. doi: http://dx.doi.org/10.1016/0885-2308(91)90019-M. URL http://www.sciencedirect.com/science/article/pii/088523089190019M.

[176] Stephen S. Intille, Kent Larson, JS Beaudin, Jason Nawyn, E. Munguia Tapia, and Pallavi Kaushik. A living laboratory for the design and evaluation of ubiquitous computing technologies. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1941–1944. ACM, 2005. ISBN 1-59593-002-7. doi: 10.1145/1056808.1057062. URL http://doi.acm.org/10.1145/1056808.1057062.

[177] Estefanía Serral, Luca Sabatucci, Chiara Leonardi, Pedro Valderas, Angelo Susi, Massimo Zancanaro, and Vicente Pelechano. Incorporating users into ami system

design: From requirements toward automation. In *Information Systems Development*, pages 499–511. Springer, 2013.

[178] Sarah Mennicken, Oliver Zihler, Frida Juldaschewa, Veronika Molnar, David Aggeler, and Elaine May Huang. "it's like living with a friendly stranger": Perceptions of personality traits in a smart home. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 120–131. ACM, 2016. ISBN 978-1-4503-4461-6. doi: 10.1145/2971648.2971757. URL http://doi.acm.org/10.1145/2971648.2971757.

[179] Sebastian Möller, Jan Krebber, and Paula Smeele. Evaluating the speech output component of a smart-home system. *Speech communication*, 48(1):1–27, 2006. ISSN 0167-6393. doi: http://dx.doi.org/10.1016/j.specom.2005.05.004. URL http://www.sciencedirect.com/science/article/pii/S016763930500124X.

[180] Davide Maria Calandra, Dario Di Mauro, Daniela D'Auria, and Francesco Cutugno. E.y.e. c. u.: an emotional eye tracker for cultural heritage support. In *Empowering Organizations: Enabling Platforms and Artefacts*, pages 161–172, Cham, 2016. Springer International Publishing. ISBN 978-3-319-23784-8. doi: 10.1007/978-3-319-23784-8_13. URL https://doi.org/10.1007/978-3-319-23784-8_13.

[181] Davide Maria Calandra, Dario Di Mauro, Francesco Cutugno, and Sergio Di Martino. Navigating wall-sized displays with the gaze: a proposal for cultural heritage. In *AVI* CH*, pages 36–43, 2016.

[182] Davide Maria Calandra and Francesco Cutugno. Automyde: A detector for pupil dilation in cognitive load measurement. In *Smart Organizations and Smart Artifacts: Fostering Interaction Between People, Technologies and Processes*, pages 135–147, Cham, 2014. Springer International Publishing. ISBN 978-3-319-07040-7. doi: 10.1007/978-3-319-07040-7_15. URL https://doi.org/10.1007/978-3-319-07040-7_15.

[183] Daniela D'Auria, Dario Di Mauro, Davide Maria Calandra, and Francesco Cutugno. A 3d audio augmented reality system for a cultural heritage management and fruition. *Journal of Digital Information Management*, 13(4), 2015. ISSN 0972-7272. URL http://dline.info/fpaper/jdim/v13i4/v13i4_1.pdf.