

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Corso di Dottorato in Ingegneria Industriale

Study and Development of Method and Tool to Support the Design of Adaptive and Adaptable User Interface

Ph.D. Dissertation of:
Francesca Gullà

Supervisor:
Prof. Michele Germani

Ph.D. Course coordinator:
Prof. F. Mandorli

XVII edition - new series

Università Politecnica delle Marche
Dipartimento di Ingegneria Industriale e Scienze Matematiche
Via Brecce Bianche — 60131 - Ancona, Italy



Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Corso di Dottorato in Ingegneria Industriale

Study and Development of Method and Tool to Support the Design of Adaptive and Adaptable User Interface

Ph.D. Dissertation of:
Francesca Gullà

Supervisor:
Prof. Michele Germani

Ph.D. Course coordinator:
Prof. F. Mandorli

XVII edition - new series

Università Politecnica delle Marche
Dipartimento di Ingegneria Industriale e Scienze Matematiche
Via Brecce Bianche — 60131 - Ancona, Italy

To my family...

Acknowledgements

This thesis is a result of my work experience, academical research and personal interests in this area. It has been a long journey to this final stage and I would like to give credits and thanks to everyone who supported me during this time.

I would like to thank my advisor, Professor Michele Germani for the provided opportunity to do this research. I wish to thank you for welcoming me to the brilliant *DT&M Research Group*, where every day I grew professionally and personally. Thanks for your daily guidance and support. Thanks for all Michele.

A heartfelt gratitude to the *Candy Company* in the persons of Alberto Strinati and Angelo Mautone for making this research work possible. Each brainstorming was an opportunity for growth and improvement, thank you very much. It is a pleasure to thank the *NTNext* and *Codermine Companies* for supporting me in the design and development of this work.

I wish to express my gratitude also to all partners of the *D4All: Design for all research project* (CTN01_00128_297089) funded by the Italian Minister of University and Research, under the National Cluster of Ambient Assisting Living Technologies (TAV).

Appreciation must also be given to the other Professors, Researchers and Collaborators involved in the *DT&M Research Group*, whose numerous suggestions have contributed to considerable improvements in this thesis.

A special thanks goes to all my colleagues who have accompanied me on this path. Thank you for enduring me, thanks for the smiles and the laughter, thanks for the suggestions, the availability and the collaboration that you gave me every day. Thanks guys!

I would also like to thank you my friends to be part of my life. When I need, I know I can count on you! Thank you all!

Thanks to the *Svampite* … because every day with you was special!!

Thanks, *Gnagna*, for understanding me and supported.

I am grateful to my family *Enzo, Simona and Marco*, who have provided me through moral and emotional support in my life. Thank you because without you I would never have done it.

I am also grateful to my aunt *Lorella* who have supported me along the way. Thank you for always been there.

To my best friend, to my life partner for sharing this experience with me day after day. Thank you for understanding and supporting me, always. Thank you for giving me the courage to experiment new ideas, to get involved and to understand that obstacles exist to be overcome. Thanks *Dani*.

To my life teachers, my grandparents *Alfredo and Ginetta*, thanks for always been there, I miss you every day..

"... people are required to conform to technology. It is time to reverse this trend, time to make technology conform to people…"

(Donald Norman, "Designing the Future" Scientific American, Sept 1995)

Abstract

In recent decades, people's daily lives have been transformed by the faster developing of information and communication technologies (ICT). The introduction of ICT in every area of human activities has made computer technology increasingly oriented towards supporting communication with users. For this reason, a growing need to understand how to design the computer systems interaction with users in order to obtain easy-to-use systems has been.

A good interface design can let the user interact and deal with the system without problem. However, the wealth of modern computer technology allows many uses of interactive systems and often becomes necessary for the user interfaces to be adapted to the context of use; as a matter of fact, the Adaptive User Interfaces (AUIs) are becoming one of the major objectives addressed by Human-Computer Interaction research.

In literature there are many articles that focus on methods to design an adaptive interface, however, there is a lack of a single effective methodology for the design of Adaptive interfaces, able to support the designer in the UI design phase. Furthermore, the adaptive interfaces research context is very heterogeneous and closely linked with each domain of application.

In this context, the goal of the present research work consists of the study and development of a method and tool to support the design of adaptive and adaptable user interface. The proposed method and tool have been used: (i) to support the development of a new adaptive system able to assist the user in performing daily life activities; (ii) to re-design the existing Hoover Wizard Application in order to make it adaptive.

Experimental results demonstrate that the proposed method and tool effectively support designing of an adaptive and adaptable user interface.

In addition, tests with users have shown that the adaptive and adaptable system was the best in terms of both performance and satisfaction. In detail, most of the users have found the benefit of using the adaptive interface.

Contents

ACKNOWLEDGEMENTS ..... **I**

ABSTRACT ..... **II**

CONTENTS ..... **III**

LIST OF FIGURES ..... **VII**

LIST OF TABLES ..... **XII**

CHAPTER 1 ..... **1**

INTRODUCTION 1

1.1 Research Context *1*

1.2 Research Scope, Objectives, Methodology and Limitations *8*

1.3 Structure of the Thesis *10*

CHAPTER 2 ..... **13**

RESEARCH BACKGROUND 13

2.1. Adaptive and Adaptable Interfaces *13*

2.2. Adaptive VS Adaptable User Interfaces *15*

2.3. Adaptive Interface Method *18*

2.4. Adaptive Interface Case Studies *25*

2.5. Adaptive Interface Evaluation *33*

CHAPTER 3 ..... **38**

ADAPTIVE INTERFACE METHODOLOGY 38

3.1. Introduction *38*

3.2. User Model *40*

3.2.1 Demographic attributes 42

3.2.2 Behavioral attributes 42

3.2.3 Health attributes 43

3.3. Environment Model *44*

3.3.1 Typology and Context of use 45

3.3.2 Devices 46

3.3.3 Physical Characteristics 51

3.4. Interaction Model *52*

3.4.1 The first level 54

3.4.2 The second level 61

3.4.3 The third level 63

CHAPTER 4 ..... **66**

ARCHITECTURE 66
4.1. Introduction *66*
4.2. Database *69*
4.3. Core *71*
4.3.1 Adaptable Engine 71
4.3.2 Adaptive Engine 71
4.4. Interface *82*
4.4.1 Concept Interface 82

CHAPTER 5 86

THE FIRST CASE STUDY: PROTOTYPING ADAPTIVE SYSTEM IN SMART ENVIRONMENT USING VIRTUAL REALITY 86
5.1. Introduction *86*
5.2. Analysis of Context of Use and Definition of Design Requirements *88*
5.2.1 Definition of Personas 88
5.2.2 Definition of System Functionalities 90
5.2.3 Definition of Use Case 91
5.3. Architecture Application *91*
5.3.1 Database Application 92
5.3.2 Core Application 94
5.3.3 Interface Application 97
5.4. Final UI Prototype *106*
5.4.1 CookBook 107
5.4.2 Household Appliances Management 108
5.4.3 Software Architecture 112
5.5. Virtual Reality Environment Design *113*
5.6. Validation *116*
5.6.1 Introduction 116
5.6.2 Bayesian Network Simulation 116
5.6.3 System Usability Evaluation 124
5.6.4 Heuristic Evaluation 130

CHAPTER 6 134

THE SECOND CASE STUDY: RE-DESIGN OF THE HOOVER WIZARD APPLICATION TO MAKE IT AN ADAPTIVE SYSTEM 134
6.1. Introduction *134*
6.2. System Specifications Definition *137*
6.2.1 Hoover Wizard Application Mapping 137
6.2.2 Architecture AS-IS 142
6.2.3 Database 143
6.3. Adaptation Goals and Action Definition *143*
6.4. Architecture Re-Design *145*
6.5. Back-end Translation Definition *147*
6.6. Adaptive Engine Design *148*

6.7. Cluster Definition *153*
6.8. DM Algorithm Test & Learning Network *155*
6.9. UI Re-Design *161*
6.9.1 Adaptive Behavior 162
6.10. User Experimentation *166*
6.10.1 Participants 167
6.10.2 Procedure 167
6.10.3 Results 168

CHAPTER 7 177

CONCLUSION 177

REFERENCES 182

List of Figures

Fig. 1: A Taxonomy of Human Computer Interaction (adapted from Section 2 of the ACM SIGCHI Curricula for HCI, ACM Press, 1992) 2
Fig. 2: The three dimensions of usability according to ISO 9241 (ISO/IEC 9241-14: 1998) 3
Fig. 3: The Norman Model (Norman, 2013) 4
Fig. 4:User-Centred Design process (Source: The Interaction Design Foundation) 6
Fig. 5: Research Methodology 10
Fig. 6: Adaptable User Interface Approach 13
Fig. 7: Adaptive User Interface Approach 14
Fig. 8: (a) Static split menu;(b) Adaptable split menu (Findlater & Mc Grenere, 2004). 15
Fig. 9: (a) The Split Interface; (b) The Moving Interface; (c) The Visual Popout Interface (Gajos, et al., 2006) 16
Fig. 10. Menu types: (a) adaptable menu (changing positions); (b) adaptive split menu; (c) adaptive highlight menu (Park, et al., 2007). 17
Fig. 11: "The participants' comments on the advantages and disadvantages of the five menu types (Park & Han 2011)." 18
Fig. 12: "Architecture for intelligent interfaces" (Benyon & Murray 1993) 20
Fig. 13:" Classic loop "user modeling - adaptation" in adaptive systems" (Brusilovsky, 1996) 20
Fig. 14: Generalized Adaptive Interface Model (Rothrock et al., 2002) 21
Fig. 15: An adaptive user interface scheme (Liu et al., 2003) 21
Fig. 16: Interface Adaptation flow chart (Yen & Acay 2009) 23
Fig. 17: The new architecture for the AUI design (Bongartz et al., 2012) 24
Fig. 18: The proposed adaptive user interface generation process (Song & Cho, 2013) 24
Fig. 19:Adaptive User Interfaces Application (since 2000) 26
Fig. 20: HEMA workstation display a) a cardiologist health record consultation session; b) a GP health record consultation session (Ghédira et al., 2002). 27
Fig. 21: A general user interface in a modern home network environment (above); the proposed adaptive user interface (Song & Cho, 2013) 28

Fig. 22: A Graphical user interface. A) Order view; b) Map view (Bongartz et al., 2012). 29
Fig. 23: The AUI workflow (Hussain et al., 2014) 31
Fig. 24: The adaptive user interface for route searching (Feng & Liu, 2015) 31
Fig. 25: The adaptive user interface presented by Plessas et al., (2017). "the list with suggested contacts (A–Z) followed by the embedded contact list (left) and the list with suggested contacts (A–Z) followed by the embedded list of recent calls (right)." 32
Fig. 26: Main variables used for the evaluation of interfaces (Van Velsen et al., 2013). 35
Fig. 27: An Evaluation Framework for Adaptive Interfaces 36
Fig. 28: The Adaptive Interface Methodology 39
Fig. 29: The User Modelling 41
Fig. 30: Demographic Attributes 42
Fig. 31: Behavioral Attributes 43
Fig. 32: Health Attributes 44
Fig. 33: The Environment Modelling 45
Fig. 34: Devices Characteristics 47
Fig. 35: Device Specifications 48
Fig. 36: Device Functionalities 50
Fig. 37: Environment Characteristics 52
Fig. 38: The Interaction Model 53
Fig. 39: Interaction Model function 54
Fig. 40: Classifications for Adaptive System methods and techniques (Brusilovsky P., 1996) 55
Fig. 41: The User Interface Design Process 63
Fig. 42: The System Adaptive Architecture 68
Fig. 43: Example of a general Bayesian Network (BN) 74
Fig. 44. The decision-making algorithm 76
Fig. 45: Example of use case scenarios simulate to create learning data 78
Fig. 46: The BBN Example 79
Fig. 47: External database related to Scenario 80
Fig. 48. The DMA algorithm tested with a thr_perc = 20. On the left the plot of the filtered nodes is shown (not subject to adaptation). On the right the nodes that satisfy the condition and are adapted are shown. 81

Fig. 49: The DMA algorithm tested with a thr_perc = 40. On the left the plot of the filtered nodes is shown (not subject to adaptation). On the right the nodes that satisfy the condition and are adapted are shown. . 81
Fig. 50. The Unified Scheme Functional 84
Fig. 51. From above Personas related to the subjects with cognitive impairment (U1), with dexterity problems (U2) and visually impaired (U3) 89
Fig. 52. The Use Case Bayesian Network..... 96
Fig. 53. The Unifed Scheme Functional (SFU) 98
Fig. 54. Paper prototype. "Cooking program setting" menu: alternative ways of presenting information 101
Fig. 55. Paper prototype: oven setting procedure "Normal Setting" mode with "tap + swipe + rotate" commands 102
Fig. 56. Paper prototype: oven setting procedure "Normal Setting" mode with "tap + typing" commands..... 103
Fig. 57. Paper prototype: oven setting procedure "Normal Setting" mode with "tap + swipe" commands..... 104
Fig. 58. Paper prototype: "Wizard" mode oven setting procedure with "tap + typing" commands 105
Fig. 59. Paper prototype: "Wizard" mode oven setting procedure with "tap + swipe" commands 105
Fig. 60. Interaction mode chosen in the alpha prototype design..... 106
Fig. 61- Recipe book: recipe search in Normal and Wizard Setting 107
Fig. 62. Recipe book: recipe display in Normal and Wizard Setting 108
Fig. 63. Oven management in Normal and Wizard Setting 109
Fig. 64. Oven management in Normal Mode (on the left: first screen, on the right: duration management) 110
Fig. 65. Oven management in Wizard Mode (help function)..... 110
Fig. 66. Dishwasher Managemnt in Normal and Wizard Setting..... 111
Fig. 67. Fridge Managment in normal and Wizard Setting..... 112
Fig. 68. Package Manger of Virtual Reality Envirnment and GUI communication 114
Fig. 69. Communication between VR and UI 115
Fig. 70. The Virtual Reality Environment..... 116
Fig. 71. The Normal Bayesian Network..... 117
Fig. 72. Simulation with Filtered nodes in the Normal Mode 118
Fig. 73. Simulation with PP Distributions in the Normal Mode 119
Fig. 74. The Wizard Bayesian Network 120

Fig. 75. Simulation with Filtered nodes in the Wizrad Mode 121
Fig. 76. Simulation with PP Distributions in the Wizard Mode 122
Fig. 77 The Adaptive Network: Normal and Wizard 123
Fig. 78. System Usability Evaluation 124
Fig. 79. The System Usability Scale (SUS) Score 127
Fig. 80. The Work Flow 136
Fig. 81: The Home Screen Wizard App 138
Fig. 82: a) The Hoover Wizard App: The Washing Start Page; b) The Hoover Wizard App: The Washer setting page 139
Fig. 83: The Hoover Wizard App: The selected Washer Program box 140
Fig. 84: The Hoover Wizard App: The Washer Options page 140
Fig. 85: The Hoover Wizard App: The delayed start setting page 141
Fig. 86: The Hoover Wixard App: The Washer Program start Page 141
Fig. 87. The AS-IS Architecture 142
Fig. 88. The Architecture re-design (in red the Adaptive Module addition) 146
Fig. 89. Cleaning and Filtering Process of the Oven Database 148
Fig. 90. Cleaning and Filtering Process of the Washer Database 148
Fig. 91. Cleaning and Filtering Process of the Dishwasher Database 148
Fig. 92. The Oven Bayesian Network 149
Fig. 93. The Dishwasher Bayesian Network 151
Fig. 94. The Washer Bayesian Network 152
Fig. 95: 4°Cluster_Washer Bayesian Networ 156
Fig. 96. Algorithm Test for the 4°Cluster_Washer with a threshold of 20% 157
Fig. 97. Algorithm Test for the 4°Cluster_Washer with a threshold of 35%: at the top the filtered nodes; at the bottom the nodes that satisfy the condition 158
Fig. 98: The two App compared: (a) the initial page with the default program of the Hoover Wizard; (b) the initial page with the suggested program of Adaptive Hoover Wizrad. 163
Fig. 99: The two App compared: on the left the program setting page of the Hoover Wizard, on the right the program setting page of the Adaptive Hoover Wizard. 164
Fig. 100: The Adaptive Hoover Wizard App - Dynamic change of setting: (a) at the temperature of 30 ° a spin speed of 800 is associated; (b) at the temperature of 20 ° a spin speed of 700 is associated; (c) at the temperature of 0 ° a spin speed of 500 is associated. 164

Fig. 101: The Hoover Wizard App: the change in the program does not affect the temperature or the spin speed. (a) at the temperature of 30 ° a spin speed of 1000 is associated;(b) at the temperature of 40 ° a spin speed of 1000 is associated; 165
Fig. 102: The two App compared. (a) In the Hoover Wizard App the timer starts at the current time; (b) in the Adaptive version the most used time of the delayed start is suggested. 166
Fig. 103: The objective evaluation 170
Fig. 104: The pre test questionnaire 171
Fig. 105: The Post test questionaire 173
Fig. 106: Comparison of the user's experience with the Adaptive Hoover App and with Hoover App 174
Fig. 107: The New Application Improvement 175

List of Tables

Table 1: Adaptation Variables 59
Table 2: Goals of Adaptation (*Dieterich, et al., 1993)* 60
Table 3: User Features Profile 69
Table 4: User Use Profile 70
Table 5: Log Adaptation Actions 70
Table 6: Context Data 71
Table 7. System functions related to the kitchen household appliances 90
Table 8. Oven Use Case 91
Table 9: Log Adaptation Action (kitchen case study) 92
Table 10: User use profile (kitchen case study) 93
Table 11: Context Data (kitchen case study) 93
Table 12. The SUS and Mars Score 128
Table 13. Usability Evaluation with Cognitive Impairments Patients 129
Table 14. Usability Evaluation with Dexterity Problems Patients 130
Table 15. Usability Evaluation with Visual Deficit Patients 130
Table 16. Nielsen's Heuristic Evaluation 132
Table 17. Database Description 143
Table 18. Adaptation Goals and Action Definition 144
Table 19: Oven Cluster 154
Table 20: Dishwasher Cluster 154
Table 21: Washer Cluster 155
Table 22: Simulation Thresholds for the washer 159
Table 23: Simulation Thresholds for the oven 160
Table 24: Simulation Thresholds for the dishwasher 161
Table 25: The pre-test questionnaire 167
Table 26: The Post-test questionnaires 168

Chapter 1.

Introduction

1.1 Research Context

The introduction of information and communications technology in every area of human activities is making computer technology increasingly oriented toward supporting communication with users. In the last twenty years, a growing need to understand how to design the computer systems interaction with users in order to obtain easy-to-use systems has been. Systems designers need to know how to think in terms of future users' tasks and how to translate that knowledge into an executable system. This can be implemented by using a good interface design to let the user interact and deal with the system without any problem. The Human-Computer Interaction can be described as the point of communication between the human user and the computer itself.
In detail, the HCI is a cross-disciplinary area that studies methods and techniques for the design and development of interactive systems that are usable, reliable and that support and facilitate human activities. The most authoritative definition of human computer interaction is that provided by Hewett et al., which defines the HCI as:
"*Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them (Hewett, et al., 1992).*"
Another important definition is provided by Preece et al. that define the HCI as
"a discipline concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them (Preece, et al., 1994)."
The term HCI relates to several stages in the development process focusing on designing, evaluating and implementing computer systems and being concerned with two issues: studying the relationship between human and computer, and discovering the methods to improve human-machine interaction (***Fig. 1***).

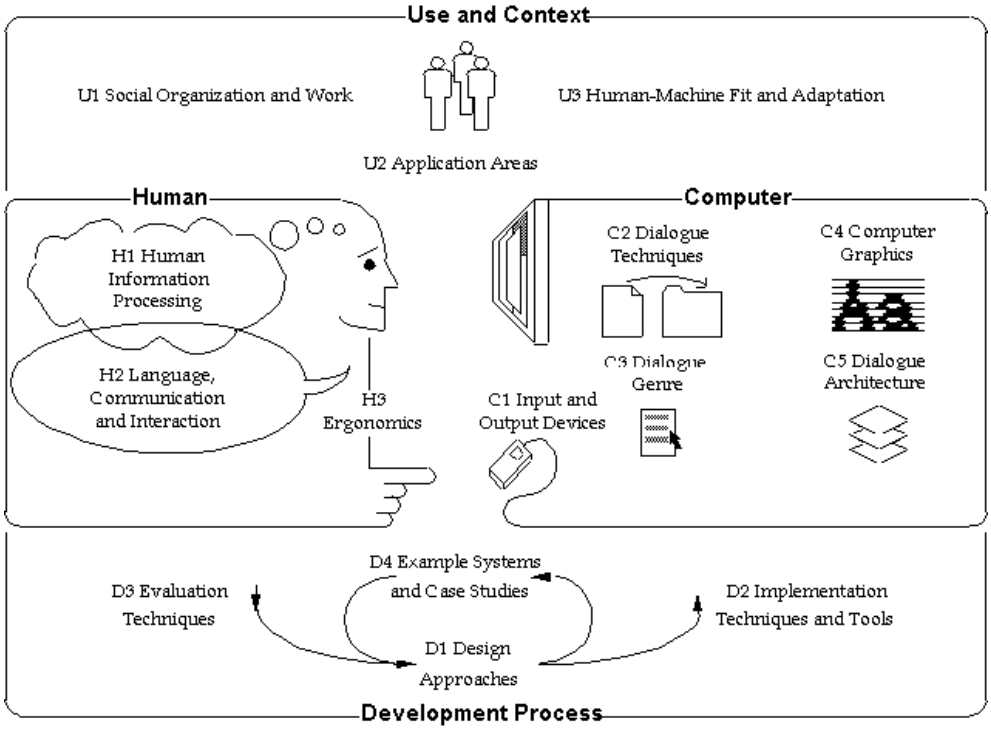


Fig. 1: A Taxonomy of Human Computer Interaction (adapted from Section 2 of the ACM SIGCHI Curricula for HCI, ACM Press, 1992)

In recent years the HCI has developed rapidly with the increase of information technology devices into every human activity and has been influenced by different disciplines such as engineering, psychology, ergonomics, design, etc (Lata, et al., 2011). The basic goal of HCI is to improve the interactions between users and computers by making computers more usable and receptive to the user's needs. This goal focus on the services that the system provides, how quickly the tasks can be achieved, and ensuring that users like the system. For this reason, involving the user, integrating different classes of knowledge and expertise and making the design process iterative through users testing, is the approach needed to develop a system, which provides the user with productivity and efficiency (Preece, et al., 1994).

In general, usability is an essential concept in HCI and is concerned with making systems easy to learn, easy to use, and with limiting error frequency and severity (Issa & Isaias, 2015). There are different general definitions of usability in the literature, however the most used is that of the ISO 9241 standard, in which the usability is defined as ***(Fig. 2)***:

"The extent to which a product can be used by users to achieve goals with effectiveness, efficiency and satisfaction in a specific context of use" (ISO/IEC 9241-14: 1998)

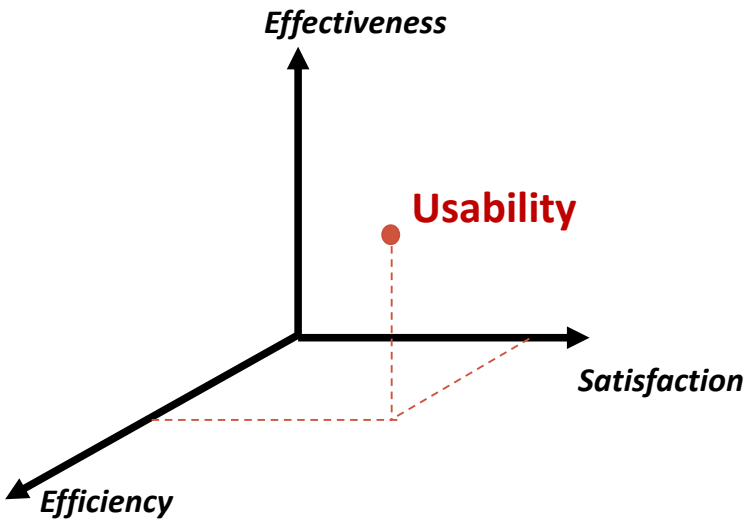


Fig. 2: The three dimensions of usability according to ISO 9241 (ISO/IEC 9241-14: 1998)

Where effectiveness means the accuracy and completeness with which users can achieve their goals; efficiency, instead, is the set of resources spent in relation to the accuracy and completeness of the objectives achieved (for example, the time required); finally, satisfaction is the comfort and acceptability of the system for users and other people influenced by its use. In general, usability refers to the "quality of the interaction in terms of parameters such as time taken to perform tasks, the number of errors made, and the time to become a competent user" (Benyon, et al., 2005). Through usability it is possible evaluate the ease of use of an object or user interface during the design phase (Nielsen, 2003). The usability evaluation stage is an effective method by which an expert team can establish the positive and negative aspects of its prototype releases and make the required changes before the system is delivered to the target users. There are different methods for usability evaluation such as heuristic evaluation that verifies compliance with the usability requirements by experts (Nielsen & Molich, 1990); the cognitive walkthrough that is an inspection method characterized by the evaluation of the effects of the interface on the end user by a group of experts, designers or technicians (Wharton, 1994); the think aloud

protocol that consists in observing the "thinking aloud" of the user in carrying out the task through the use of the system (Jääskeläinen, 2010); the usability test consisting in the observation in a controlled environment of the interaction between user and system in the execution of a pre-established task and subsequent analysis of the behaviour and finally the use of questionnaires to be administered to the users (Sauro, 2011; Tullis, et al., 2004).

From the user's perspective, usability is considered a very important aspect in the development process as it can mean the difference between performing and completing a task in a successful way without any frustration.

To understand what is meant by human-machine interaction and how this discipline is linked with usability, it is appropriate to refer to the Norman model (Norman, 2013). The purpose of this model is to identify the main steps in user interaction by providing a useful and structured indication of the main aspects to consider in the systems and interfaces designing.

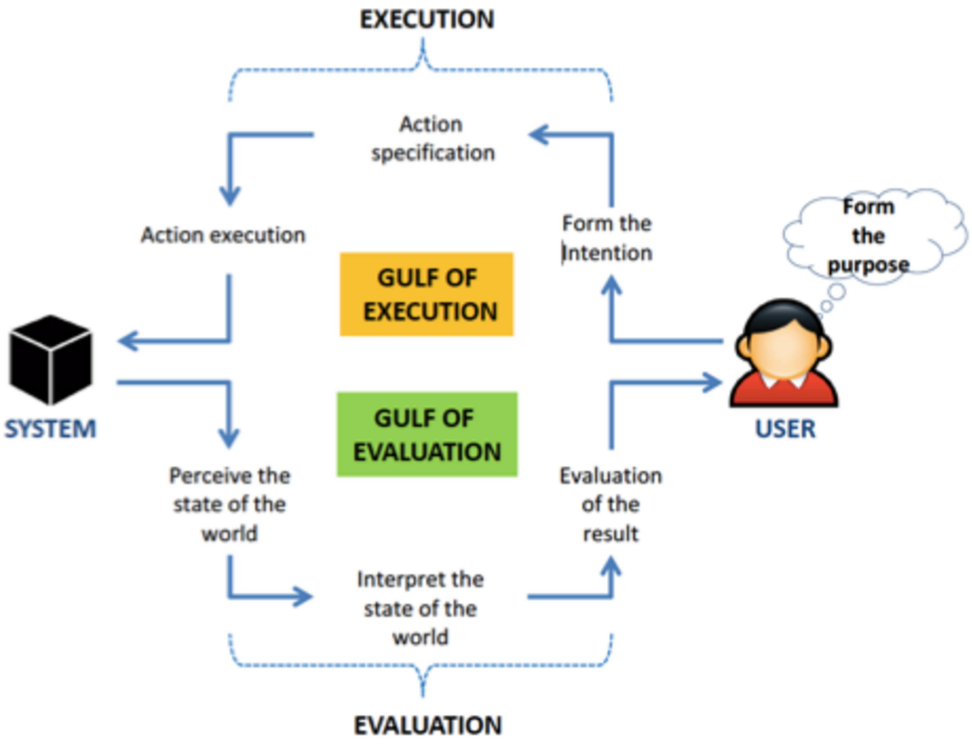


Fig. 3: The Norman Model (Norman, 2013)

The Norman model identifies seven possible steps to describe human-computer interaction: formulate the goal; formulate the intention; identify the action; perform the action; perceive the status of the system; interpret the system status and finally evaluate the result with respect to the final objective (***Fig. 3***). Norman places the seven phases in the context of the cycle of an interaction, in which he identifies the "gulf of execution" (the difference between the user's intentions in terms of the actions to be performed and the actions actually allowed by the system) and the "Gulf of evaluation" (the difference between the representations provided by the system and those expected by the user). In interfaces design the two gulfs can be useful to identify the discrepancies between what the user would like to do and what he can do (gulf of execution) and between what the system presents and what the user expects (gulf of evaluation). The user interface must allow these activities to be carried out in the most immediate and intuitive manner possible. To this end, an important phase in planning is the task analysis, which aims to identify the most relevant tasks and their characteristics. To understand this, it is important to involve the end user in the planning process and to keep in mind how the activities currently take place.

In this regard, various techniques may be useful: interviews, workshops, questionnaires, observing users in the usual context, considering the existing documentation and the learning methods used. The analysis of the tasks can be done at different levels of granularity, also according to the objectives. The result is the identification of the most important tasks for the application considered, the related problems, attributes and preferences of the user. This is a central point in the design. The User-Centered design calls for involving users throughout the design process via a variety of research and design techniques so as to create highly usable and accessible products for them. The UCD is an iterative design process in which designers focus on the users and their needs in each phase of the design process (Abras, et al., 2004). The process has been defined and described by several authors and even by some ISO standards, such as the 13407, Human-Centered design process (Jokela, et al., 2003). This ISO standard establishes four main activities for the UCD process (***Fig. 4***):

1. Understand context of use, in which users can use a system;
2. Specify user requirements;
3. Design solutions, in which designers develops possible approaches;
4. Evaluate against requirements and understand how the project is close to user's specific context and needs.

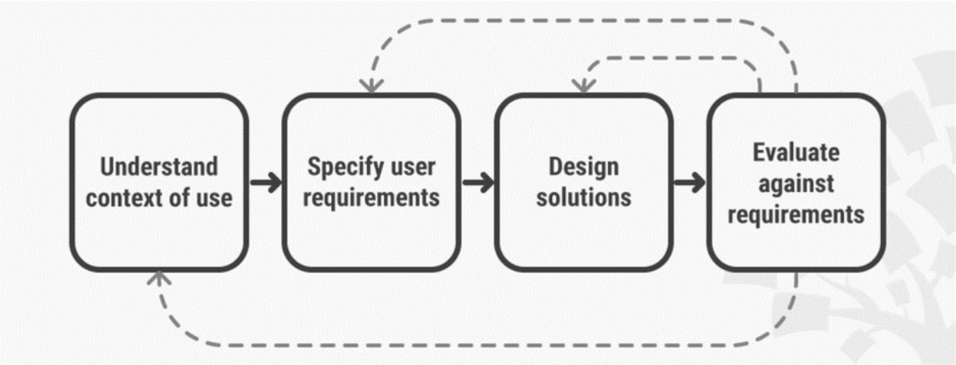


Fig. 4:User-Centred Design process (Source: The Interaction Design Foundation)

In order to have feedback from users of the interface or system to improve it in itinere, the UCD process implies the involvement of users in the various design and verification phases.

During the process, mockups and/or prototypes are submitted to users in order to test their reactions and obtain suggestions and useful indications to obtain a better interface. This process, which uses active feedback from users, produces a significant number of advantages: final applications are easier to understand and use; in this way a positive feedback is obtained from the users, the satisfaction of the user is improved and the request for explanations and support is reduced.

User interface (UI) design is the process of making interfaces in software or computerized devices with a focus on looks or style. An interface is defined by Hackos and Redish **"***as a bridge between the world of the product or system and the world of the user. It is the means by which the users interact with the product to achieve their goals. It is the means by which the system reveals itself to the users and behaves in relation to the users' needs (Hackos & Redish, 1998)".*

The aim is to design the elements carefully, to define clear, economical, convincing solutions that can be more easily assimilated, understood and remembered, and which attract the user to the important aspects allowing to achieve the objectives quickly and without errors. Another important aspect of designing user interfaces is how to structure and organize their presentation (Brown, 1999). For this purpose, there are a number of communication techniques aimed at helping the user to scroll through the interface, interpret the elements and find what they are looking for. What the designer must try to do is to group elements, create hierarchies, represent

relationships, indicate sorts between elements and, in the end, find an overall balance (Mayhew, 1992).
Grouping elements is useful to indicate those that are semantically more connected to each other, to create hierarchies to highlight the most important elements (for example, those that are accessed more frequently), the relationships are used to understand how one or more elements can influence other elements and, finally, there may be various types of logical or temporal arrangements between groups of elements. Getting the most immediate understanding of these semantic relationships between elements of the interface is the task of the designer.
The wealth of modern computer technology allows many uses of interactive systems. It often becomes necessary for the user interfaces to be adapted to the context of use, which can be considered from three points of view: those relating to the user, the device and the surrounding environment.
As for the user, important aspects are the objectives and the relative tasks, the preferences and the level of knowledge of the application domain and of the modalities of interaction. In the device used for interaction, it is important to consider the supported modes, the width and resolution of the screen, the capacities and speed of connection with other devices. Finally, the environment has various aspects that can influence the modes of interaction such as noise level and current light, the objects that are available. User interfaces must adapt to these aspects for greater usability. The problem of adapting user interfaces is not trivial. Creating a mechanism that can ultimately produce a unique UI for every single user that exists out there seems like a difficult. Each element of the interface must adapt to the user in terms of knowledge, habits and needs.
Considering the complexity of the problem, the need to know and distinct it down of terms adaptivity and adaptability was introduced. Adaptivity and adaptability are terms used to describe the “how” an intelligent mechanism can achieve the goal of tailoring a UI to a specific user.
In detail, adaptability is the ability to modify aspects on the user's explicit request according to predefined options and adaptivity, consists in the ability of the system to modify aspects dynamically without explicit user request. While adaptability allows the user to choose how to interact with an application between a predefined set, adaptivity implies systems that dynamically change it from the context of use (Findlater & McGrenere, 2004).
On the one hand, this implies more flexibility, but on the other, it implies new usability problems. There are different elements that can be adapted in a user interface such as a layout, dynamic behaviour (in the choice of navigation modes,

the activation and disabling of interaction techniques, etc.) and finally the content of the information provided.
Adaptation and adaptability are on their way to become common features of user interfaces that claim to provide access to users with different abilities (user interfaces for all). Facing this fact, one should expect a sound conceptual and technical background for adaptable or adaptive system design and evaluation. Unfortunately, neither a common terminological framework nor a comprehensive framework for design and evaluation exists. Furthermore, the design of adaptive and adaptable user interfaces represents a difficult challenge for the designer because it depends on the complexity of human intentions in a particular time and place. A first criticality is represented by the complexity of the users, in terms of physical, cognitive, socio-cultural and attitudinal diversity. The second criticality is represented by the environment complexity; it includes the environment/device typology, context definition and the definition of functional requirements of the environment. Finally, the third criticality is represented by the complexity of environment interaction.
With the aim of solving this issue adaptive interfaces have been designed and employed in specific contexts: for example, in domestic environments (Kushwah, et al., 2004; Gullà, et al., 2016), in workplaces application (Letsu-Dake & Ntuen 2010; Evans, et al., 2013) and healthcare domain (Vasilyeva et al., 2005; Shakshuki, et al., 2015). However, all these applications propose specific approaches for the development of an adaptive interface in specific contexts, without considering the opportunity to design a unique methodology for the design of adaptive interfaces.
For this reason, we propose a new approach that aims to support the design of adaptive interfaces by improving the user-environment interaction. This can be achieved by developing an appropriate system that works on two levels: The first level is the technological adaptation that consists of a custom system design into the initial setup which can continuously monitor the user's behaviour and mode. The second level is the utilities resulting from the correspondence between the activities performed by the user. This level consists of a study of the proposed service in terms of satisfaction of needs and user's expectations, as well as the technical implementation of specific functions.

1.2 Research Scope, Objectives, Methodology and Limitations

In order to create a standardized approach to Adaptive interface design, the present research work aims to propose a new method and tool to support the design process of adaptive and adaptable user interfaces.

The result documentation might be used as an extension to existing guidelines as a reference for application developers and UI designers for adaptive interfaces creation in the different context of use.
In order to achieve the main objective, the following objectives have been achieved:

- **to design a new method and architecture** to support the adaptive and adaptable user interface design; The first objective includes the creation of a new methodology able to support the designer in adapting a new or already existing system.
- **to model application** with respect to the formulated adaptive user interface guidelines and patterns; The second objective of this Thesis is to exploit the proposed approach to designing a new ex novo adaptive application able to support different users with several typologies of impairment (i.e., visual, motor, cognitive) in performing cooking activities.
- **to re-design an existing application** for making an adaptive system. The third objective is re-design the Wizard Hoover Application in order to improve user experience increase UI accessibility and readability and optimize the layout through removing features that are not required by the user and adapting others to his preferences.
- **to validate the results** of the created and improved applications**;** The fourth objective is to illustrate the benefits of the designed approach.

Fig. 5 shows the research methodology followed during the three years of activity: first, the state of the art in the HCI context, adaptive and adaptable user interface, was explored and a first analysis was carried out on the main existing adaptations mechanisms. Subsequently, on the basis of the acquired knowledge, the methodology and architecture for the design of AUI were developed. A decision algorithm has also been defined to support the architecture.

Finally, the methodology and architecture were tested in two different case studies:

- **1st Case Study** – **D4ALL**: Prototyping of a Simulated Adaptive System in VR (ex novo).
- **2nd Case Study** – **Candy-Hoover**: Adaptive Component Addition into Existing System (Re-Design).

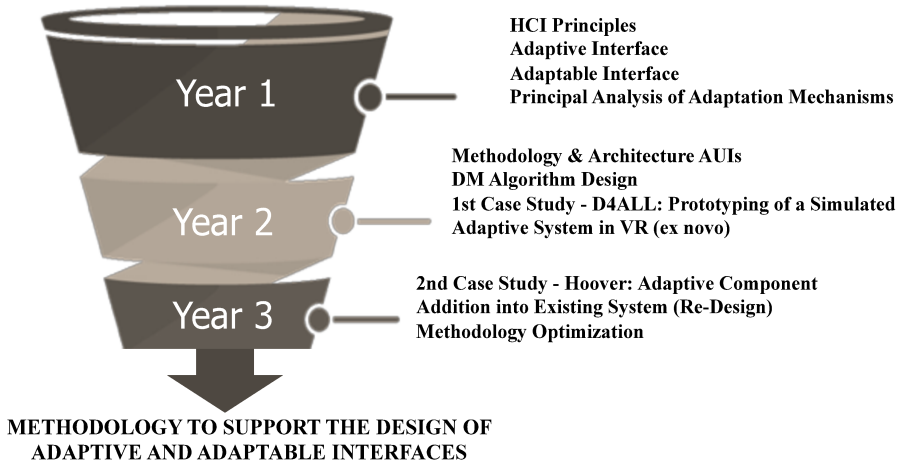


Fig. 5: Research Methodology

Although this research has reached its aims, there was some unavoidable limitation. First, because of the time limit, this research was conducted only on small size user groups. Therefore, to generalize the results for larger groups, the study should have involved more participants at different levels. Furthermore, the proposed methodology should be tested in various applications and contexts of use to validate it and highlight further advantages and disadvantages.

1.3 Structure of the Thesis

This thesis is organized as follows. In **Chapter two** an overview of the Adaptive and Adaptable system is provided.

Chapter three focuses on the Adaptive interface methodology. A new design approach to adaptive interface has been applied that consists in three different models: the User Model (UM) which provides the description of the user's profile pattern useful to understand the end user's capabilities and needs; the Environment Model (EM) which identifies the main system functionality and the information necessary to describe the human machine interaction and the Interaction Model (IM) that consists in the definition of the adaptation goals, rule, variable and the adaptation mechanism.

In **Chapter four** a global architecture based on three functional modules is proposed, which is based on the Database, the Core and the User interface. The mind of the

application is represented by the core layer that provides to manage the application logic. Besides managing the application routine activities, in the application core, the adaptive engine is developed, aimed to make adaptive the proposed system.

Chapter five describes the first case study that aims to develop a new adaptive smart system able to support user (with and without disabilities) in performing a daily task by recognizing their preferences and actions and adapting the system feedback consequently. With the aim to develop an easy, efficient and usable adaptive smart system, the final users have been involved in the whole design and development process. The system was validated through a Virtual Reality (VR) system allowing the user interaction evaluation and helping the usability improvement.

In **Chapter six**, the second case study is presented. In detail, the described methodology is applied to an existing Wi-Fi system for the control and management of household appliances developed by a large Italian Company Hoover – Candy Group. In order to simplify the control and the setting of household appliances, the mobile application UI has been redesigned. Based on the user profile, the previously interaction and the monitored actions, the adaptive module introduction allows finding the optimal features combination for the interface.

Finally, in **Chapter seven** the main conclusions and recommendations of this work are presented.

Chapter 2.
Research Background

2.1. Adaptive and Adaptable Interfaces

Adaptive systems are entities that adapt their displays and available actions to the user's current goals and abilities, by monitoring user status, the system state and the current situation" (Rothrock et al., 2002). Benyon and Murray (1993) define Adaptive Systems as "*systems which can automatically alter aspects of their functionalities and/or interface in order to accommodate the differing needs of individuals or groups of users and the changing user needs over time*". To achieve this, such systems have to learn the context in which they evolve and to adapt their interfaces and interaction modalities to communicate with users. The research literature describes many approaches that can be used to design flexible user interfaces, which can be classified into two broad categories: adaptable and adaptive. The two terms– 'adaptive' and 'adaptable' are often used interchangeably to classify interfaces that have the capability to change certain aspects of their behaviours due to external factors or user input.

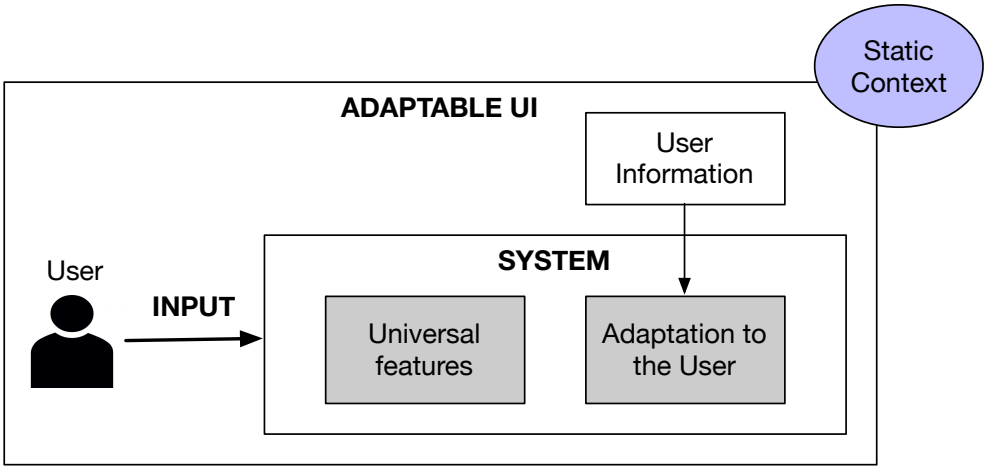


Fig. 6: Adaptable User Interface Approach

Adaptable User Interfaces (AdUIs) can be defined as systems in which the activation of user-computer interaction is performed by the final user through the selection of a specific user profile from a predefined list (***Fig. 6***).
In an adaptable interface the features can be modified by the user to fit the context of use (Letsu-Dake and Ntuen, 2010). Thus, an adaptable interface leaves the interface in a modifiable form and places the onus of modification on the operator giving the user full control of the adaptable interface. The Adaptability is based on the users' known characteristics and preferences; these are defined prior to their interaction session and, in any case, are assumed to remain static during the session (Grundy and Hosking, 2014).
Instead, *Adaptive User Interfaces* (AUIs) are systems that change their structure, contents and elements according to the need and context of the user (Schneider-Hufschmidt et al., 1993; Shneiderman and Maes, 1997) (***Fig. 7***). An adaptive interface autonomously adapts its information processing mechanisms and their actions to meet current goals and abilities of the user (Rothrock et al., 2002). Martins further notes that adaptive user interfaces differ from adaptable user interfaces because they may change automatically (without user intervention) at real interaction time (Martins F., 1996).

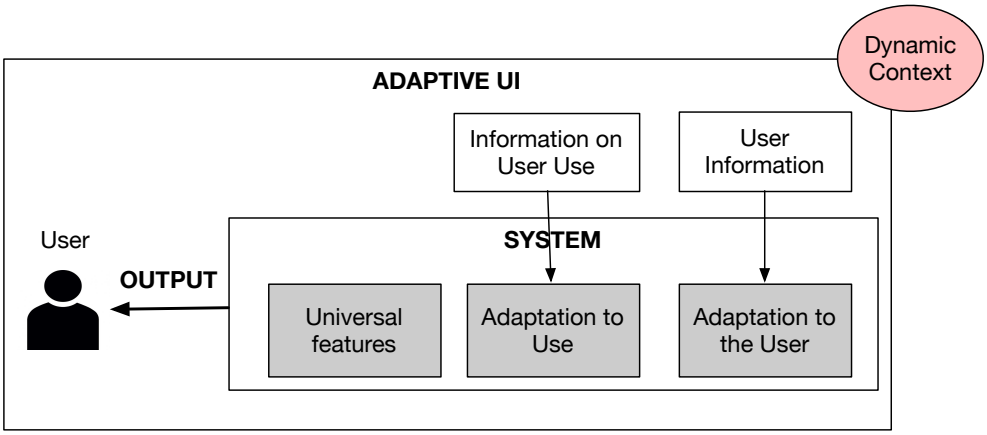


Fig. 7: Adaptive User Interface Approach

AUIs are situated and aim to enable the user to become smarter and more empowered. Adaptive user interfaces may be considered as a special class of learning systems that are developed to aid humans, in contrast with traditional ones, which was introduced with the aims to replace domain experts (Langley, User modeling in adaptive interfaces., 1999).

2.2. Adaptive VS Adaptable User Interfaces

The separation between the adaptively and adaptability is very thin. Both approaches have their pros and cons. In literature, there are few studies that highlight the benefits of the adaptively to the detriment of the negative effects of such systems. However, there are many studies that present the successes of an adaptable approach without examining the causes of their success or failure (Gajos, et al., 2006).
In 1985, Greenberg and Witten compared the performance of an adaptive menu and a static menu. The experiment consisted of searching for names in a telephone directory in both menus. The results showed that the adaptive structure that provided a shorter and more immediate search path was preferred by users compared to the static one. Another important study in which a prototype adaptable interface for Microsoft Word 2000 (MS Word 2000) was compared to the native adaptive interface is the one presented in 2002 by McGrenere. In this case, results favour the adaptable design, but the adaptive interface definitely had support.
In 2004 Findlater and McGrenere conducted an experiment that compared three different types of interfaces: static, adaptive and adaptable, and they found somewhat different results under different circumstances (***Fig. 8***).

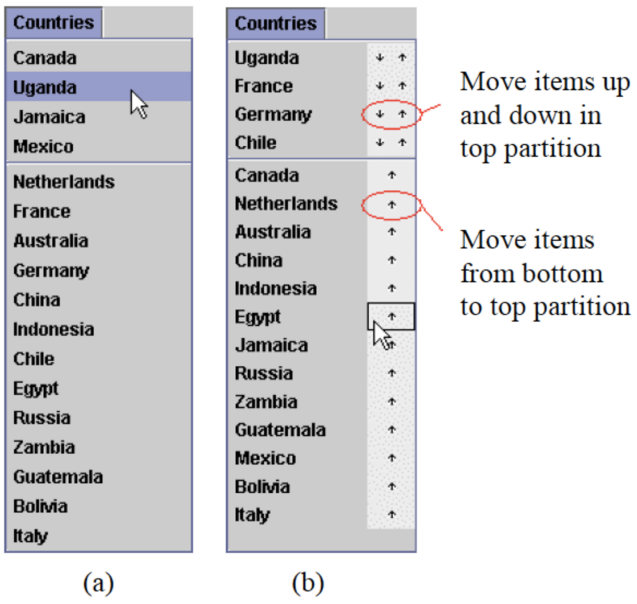


Fig. 8: (a) Static split menu;(b) Adaptable split menu (Findlater & Mc Grenere, 2004).

Results show that the majority of users (55%) preferred the adaptable interface to the other two. Despite the result, the few users who preferred the adaptive interface expressed strong support for it. This suggests that combining the two in a mixed-initiative design may be the best way to satisfy a wide range of the user.
Gajos et al. (2006) examine which aspects of adaptive interfaces affect their success or failure. In particular, they design and implement three adaptive graphics interfaces and evaluate them with respect to a non-adaptive one (***Fig. 9***). The results show that the predictive accuracy and the frequency of adaptation influence largely impact the user satisfaction. In fact, the slow pace adaptation of Sears and Shneidman in 1994 provided benefits compared to a non-adaptive interface, while the fast adaptation, presented by Findlater and Mcgrenere in 2004, led to negative results.

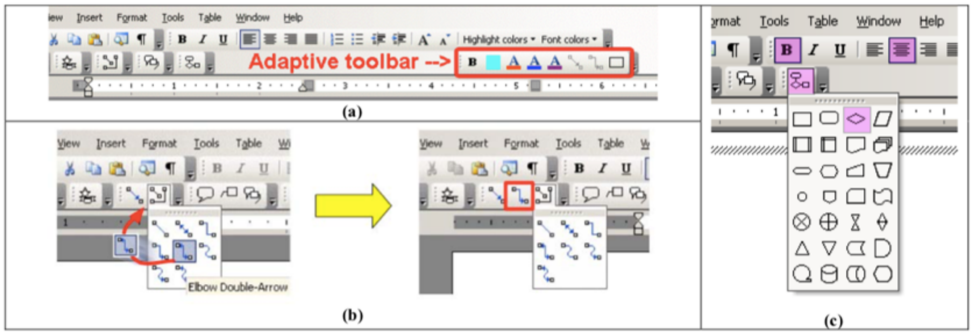


Fig. 9: (a) The Split Interface; (b) The Moving Interface; (c) The Visual Popout Interface (Gajos, et al., 2006)

In the 2007 Park, et al. examined the effectiveness of adaptable and adaptive menus in desktop applications. In detail, the study investigates an adaptable menu and two different adaptive menus. The two adaptive menus consist of an adaptive split menu that moves frequently used menu items to the top, and an adaptive highlight menu that automatically boldfaces frequently used menu items (***Fig. 10***). The experimentation conducted to evaluate the usability of these menus shows how the adaptable menu was the best in terms of performance and user satisfaction.
In a more recent study, Findlater and Mcgrenere (2008) have shown that AUIs are more beneficial when screen real estate is constrained and that the adaptive accuracy conditions were better in the small screen displays compared to the desktop sized displays. Moreover, Gajos et al., (2008) showed how the increase in predictability and accuracy in terms of performance of adaptive interfaces has led to greater satisfaction of end users and consequently to the greater use of these systems.

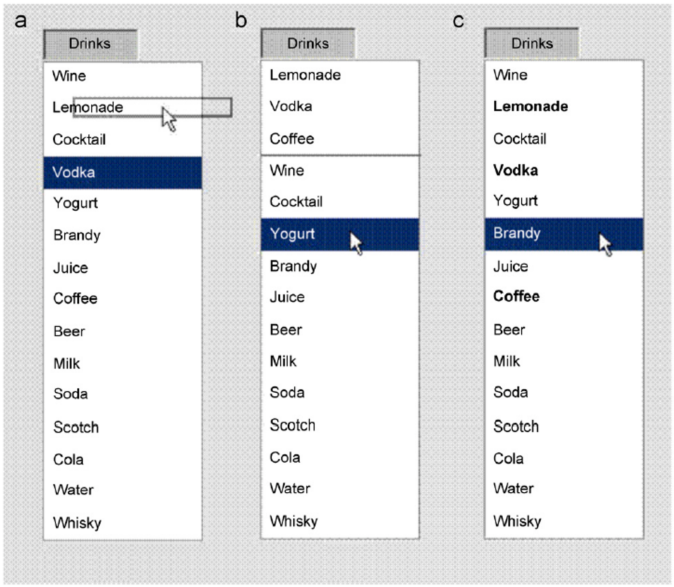


Fig. 10. Menu types: (a) adaptable menu (changing positions); (b) adaptive split menu; (c) adaptive highlight menu (Park, et al., 2007).

In 2010 Lavie & Meyer present an adaptive user interface for an onboard telematics system with four different levels of adaptivity (from manual to fully adaptive). The conducted experiment is aimed to examine the positive and negative effects of the interfaces by involving groups of users of different ages and subjecting them to different tasks in a routine and non-routine situation. The results show how the frequency of the tasks, the age of the user, the difficulty of the task and the user's involvement affect the adaptivity. In particular, in familiar/routine situations, the fully adaptive system has achieved a level of satisfaction for all the participants involved, while in non-family situations, the cognitive load substantially increased negatively affecting the performance. Intermediate levels of adaptivity keep users involved in the task and help them become more proficient when performing both routine and non-routine tasks. However, intermediate levels of adaptivity should also be implemented with care, because they may also have adverse effects when users encounter non-routine situations.

Furthermore, Letsu-Sake & Ntuen (2010) provide a study for the experimental evaluation of an adaptive interface and an adaptive one for the control of a complex process. The results show how the adaptive interface has achieved significantly better results than the non-adaptive one, in terms of time to reach the operational

target in fault conditions, error detection time, a number of system failures and detection rate of failures, respectively.
Another important study was conducted in 2011 by Park & Han. In this study, a method of coupling adaptable and adaptive approaches to the design of menus has been proposed. In particular, four different types of adaptation have been defined: adaptable with/without system support and adaptive with/without user control (***Fig. 11***). A controlled lab experiment was conducted to compare the menu types and investigate the effects of the system support and the user control in the adaptable menus. The results showed that adaptable and adaptive menus were superior to the traditional one both in terms of performance and user satisfaction.

Menu type	Advantages	Disadvantages
Traditional	• Easy to learn the hierarchy • No additional learning required • Easy to select infrequent items • No need to adapt	• Difficult to select frequent items • Requires more time • Requires more button clicks • Need to recall the category • Difficult to select items in the bottom part of the menu
Adaptable	• Can be organized by the user • Easy to select frequent items • Not confusing due to the consistency • Takes less time • Easy to select items in the bottom part of the menu • Easy to remember the hotlist items	• Requires effort to adapt • Difficult to determine what to adapt • Sensitive to variation in selection frequency • Difficult to determine the fixed hotlist order • Difficult to recall the category for infrequent items • Requires learning
Adaptable with system support	• Easy to adapt the hotlist • Easy to locate frequent items in the menu • Easy to notice change in frequency • No need to remember the frequency	• Need more detailed frequency information • Just did not help • Difficult to attend infrequent items • Can harm privacy
Adaptive	• No need to adapt • Enabled efficient selection • Automatically ordered by frequency • Easy to select frequent items • Not disturbing	• Difficult to remember the hotlist items • Confused by changed order • Not adaptable by the user • Difficult to select infrequent items • Can harm privacy
Adaptive with user control	• Partly adaptable • Helped to remember the hotlist items • Feels like under user control • Requires less effort for adaptation	• Distracting • Just did not help • Confused by delete/replacement notices • Requires additional effort

Fig. 11: "The participants' comments on the advantages and disadvantages of the five menu types (Park & Han 2011)."

The main challenge is to effectively combine adaptive and adaptable elements in the most natural way possible. For this reason, the present work would try to fill this gap through the introduction of a new adaptive and adaptable system.

2.3. Adaptive Interface Method

In spite of AUI research progress, there are many articles and books that focus on restrict domains since researchers come from diverse scientific disciplines. Furthermore, no technique has been shown to be applicable at a general level.
Dieterich et al., (1993) define the adaptation strategy as a decision-making process, which answers to the following questions: Where? Why? What? To what? How?

The first attribute identifies the application areas in which the adaptive system can be helpful; the second attribute identify the reason to make an adaptive system: depending on the context and on the system type, the “adaptation goals” concern a search goal, a problem-solving or learning goal (Stephanidis et al., 1997); the third attribute refers to what aspects are subject to adaptation in a particular system; in detail, what features undergo a change depending on which users: “adaptation constituents” (Dieterich et al., 1993); the fourth attribute refers to what aspects of the user consider to make adaptation: in general, the adaptation depends on the user's characteristics and preferences, on the user's goal, on the user's background and on the utilization context (Brusilovsky P., 1996); the fifth attribute identifies how to make adaptation through the definition of a set of rules "adaptation rules" (Stephanidis et al., 1997); finally, an additional dimension has been added to those highlighted by Benyon and Murray (1993): when to adapt? The adaptation can be carried out in run-time (during the use of the system) or statically (at the first access of the system). Stephanids et al., (1997) define these attributes as “adaptation determinants”.

A standard framework for developing Intelligent User Interfaces is the one proposed by Benyon and Murray (1993).

The framework is based on three different models: the user model that represents the user characteristics or more generally the agents, with which the system can interact; the domain model that defines the aspects of the application which can be adapted and forms the basis of all the inferences and predictions which can be made from the user-system interaction; the interaction model that defines the relationships between user and domain model. Through this model, the system can make inferences about the user’s beliefs or goals and may tailor its behavior to the needs of a particular interaction (***Fig. 12)***.

Subsequently, Brusilovsky (1996) defines an adaptive interface through a decomposition in two stages: the first one models the user while the second relies upon the generated user model to provide the basis of adaptation (***Fig. 13***).

In the (2001) Jameson defines a general information processing diagram for a user-adaptive system.

Rothrock et al., (2002) classifies existing approach to adaptive interface design along three main points of views: human-factors, human-computer interaction and hybrid. Furthermore, they provide a general framework of an adaptive interface through the definition of a set of elements that are common to all AI: inputs, user variables, inference mechanism, interaction model, decision inference mechanism, system or environment variables, selection mechanism, evaluation mechanism.

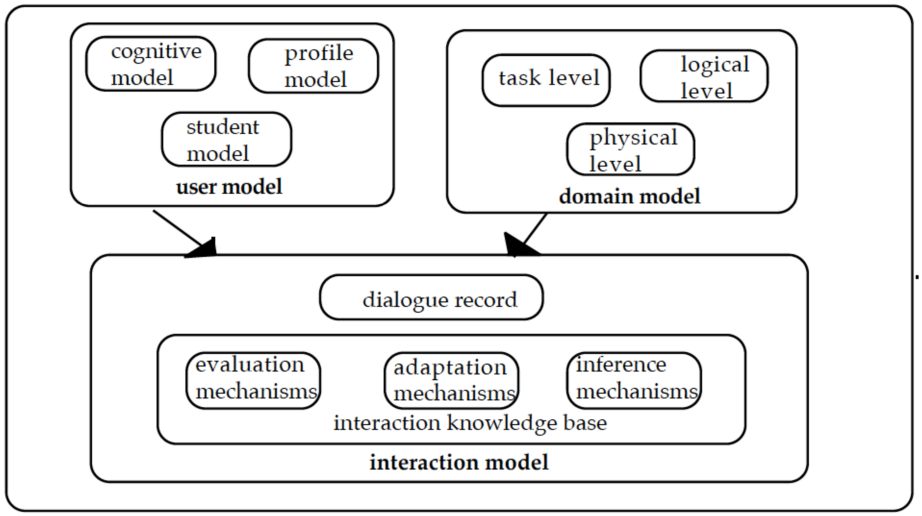


Fig. 12: "Architecture for intelligent interfaces" (Benyon & Murray 1993)

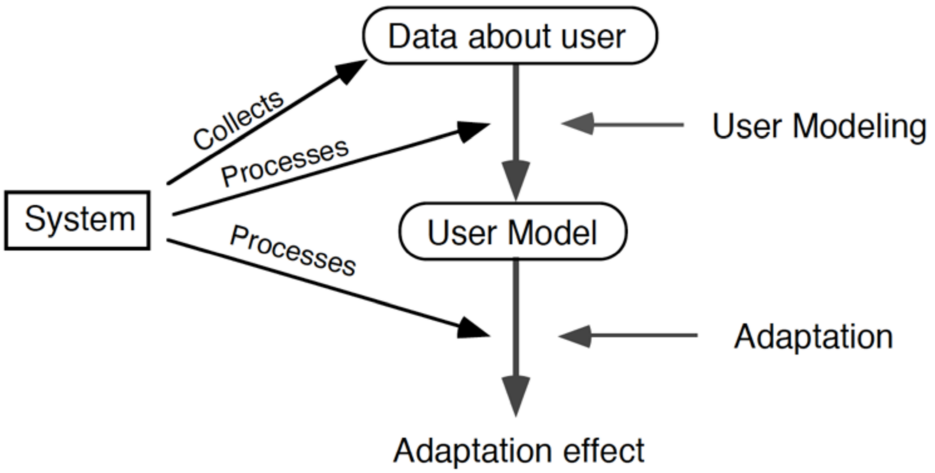


Fig. 13:" Classic loop "user modeling - adaptation" in adaptive systems" (Brusilovsky, 1996)

Combining all these elements they obtain the generalized adaptive interface model shown in ***Fig. 14***. Ghédira et al., (2002) provide a method for the development of adaptive hypermedia navigation interfaces, with an emphasis on adaptability and the ability to personalize user interfaces for decision making.

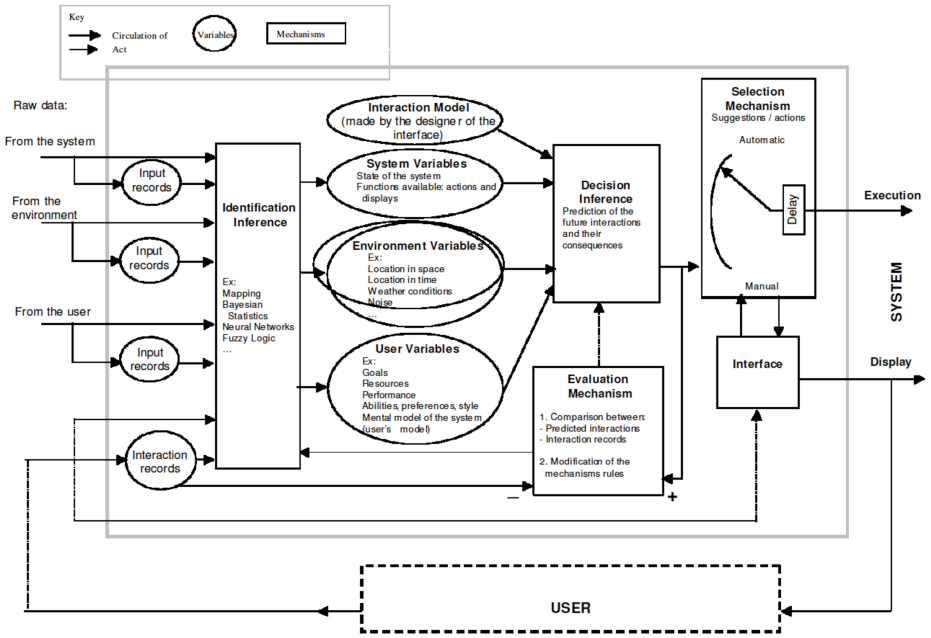


Fig. 14: Generalized Adaptive Interface Model (Rothrock et al., 2002)

In the (2003) Liu et al., design a dynamic, seamlessly personalized adaptive user interface through the use of an approach based on episodes identification and association (EIA) (***Fig. 15***). In particular, through their approach, they identify a set of rules able to describe and predict the user's behaviour through the identification of recurrent events in user actions sequences.

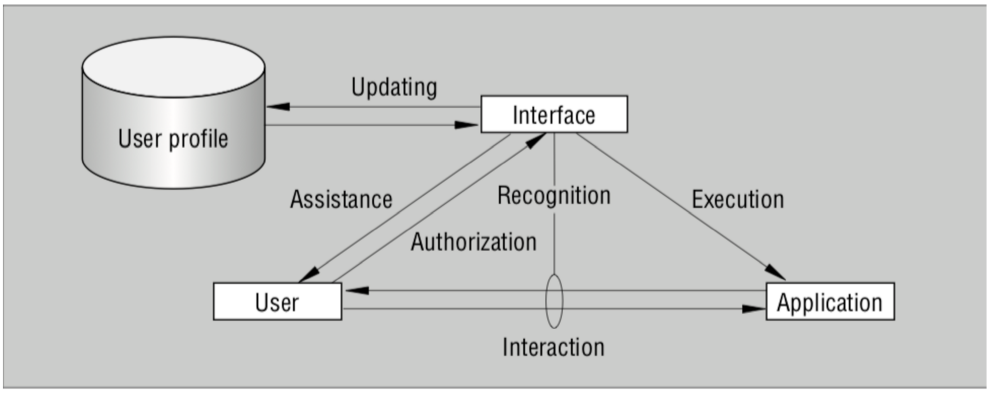


Fig. 15: An adaptive user interface scheme (Liu et al., 2003)

Savidis and Stephanidis (2004), sustain that a method that leads to the single instance construction in the design of interfaces is inappropriate as it cannot accommodate the differences resulting from the needs of individual users. Therefore, they provide a design approach that supports the definition of a polymorphic design solution, for each system functionalities, so that allows the adaptation for each user in different contexts of use. This approach is called the unified interface design process method. However, the afterward produced scientific literature has developed additional algorithms and approaches applicable to individual cases and not generalizable.

In (2006) Namgoong et al. argue that in a ubiquitous environment, there are various types of devices that provide different types of services to the users, but these devices have interfaces that do not fit the end customer. In order to make easy the devices use, it is necessary to create integrated and heterogeneous interfaces that adapt to the user. For this reason, they propose an adaptive user controller exploiting the representation of devices through semantic description. In this way, the controller dynamically builds the interfaces using the knowledge of the devices present in the environment and the adaptation of user preferences.

Yen and Acay (2009) introduce a generalized framework for the adaptation of the user interface in complex environments (***Fig. 16***). In detail, their approach is based on the identification of mapping between two spaces: the psychological space, understood as a level of preference and/or comfort of the user interface; the parametric space intended as a combination of interface parameters to increase the interface preference level in the corresponding psychological space.

Psychological distance between this desired interface and the current interface should be minimized in order to increase the efficiency.

The proposed method can effectively improve human-computer interaction and can be a valid approach for the interface adaptation in complex environments.

Subsequently, an adaptive graphical user interface generation framework for Web Services has been proposed by He et al., (2008). The framework is designed using a service-based approach. A GUI display and layout model that used preferences and constraints is proposed in the framework to facilitate the display layout decision during the GUI generation.

In (2010) El-Bakery et al., sustain that user interface optimization is a key part of simplifying the management of user activity in software applications.

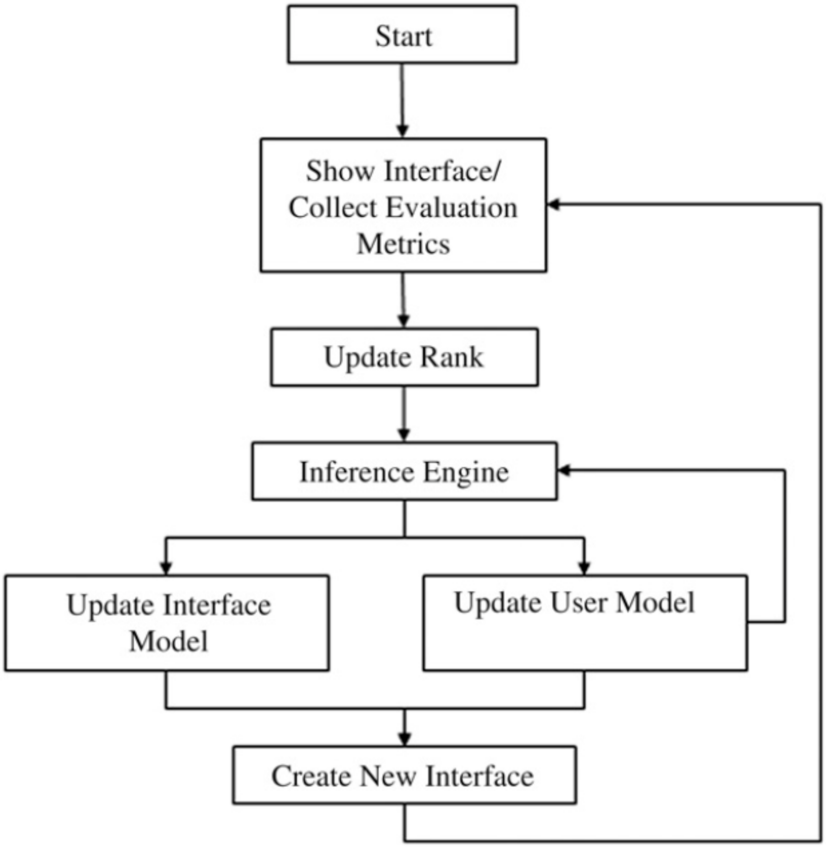


Fig. 16: Interface Adaptation flow chart (Yen & Acay 2009)

For this reason, they provide a new design for the adaptive user interface, analysing the main design problems for web interfaces, and discussing guidelines for web application design.

Another approach to supporting the adaptability of information for users in the Web is provided by Hervás & Bravo, (2010). Bongartz et al., (2012) define a new architecture for the AUI design in the work environment context; the architecture is based on the design concept of the model-based interface (***Fig. 17***). Adaptation rules are specified in terms of events, conditions and actions and are triggered by contextual events such as user preferences or environmental changes, etc. The rules impact the partial or complete modification of the user interface (interaction mode, font size, color, etc.)

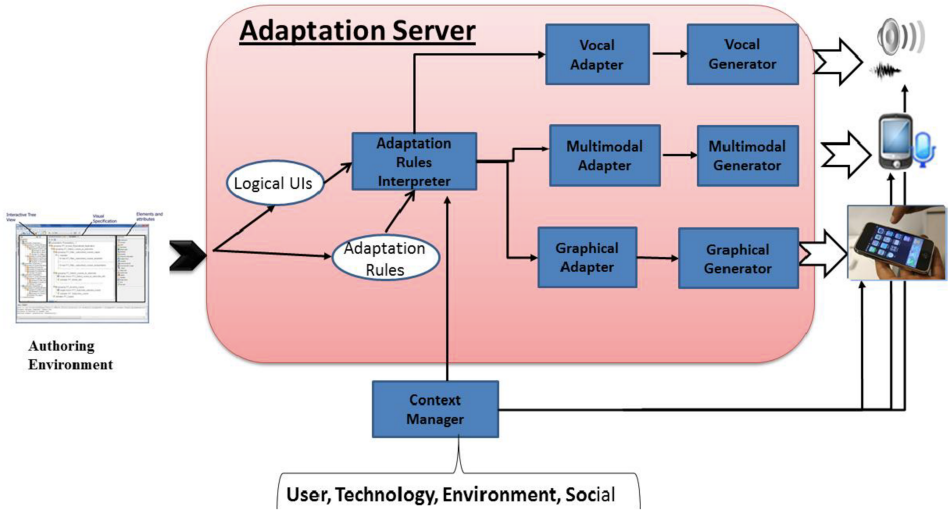


Fig. 17: The new architecture for the AUI design (Bongartz et al., 2012)

Song and Cho (2013) provide a new context-adaptive user interface for the control of devices in a ubiquitous home environment through the use of Bayesian Network (***Fig. 18***).

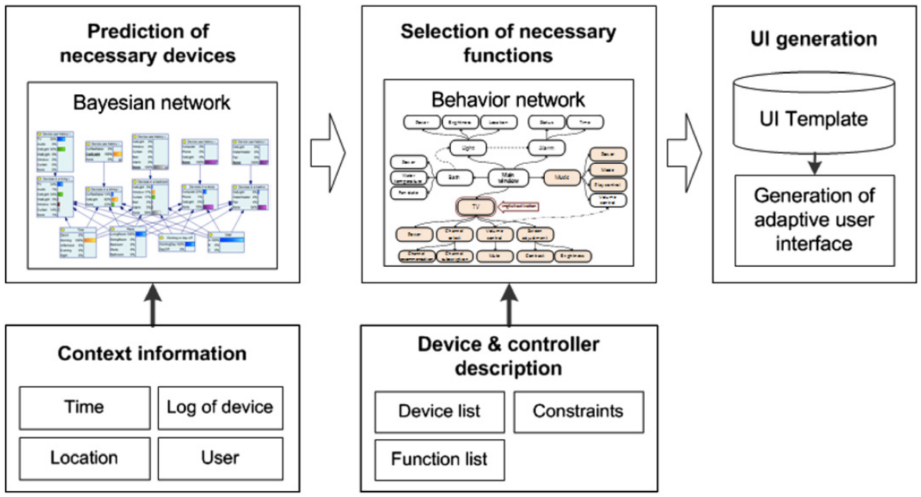


Fig. 18: The proposed adaptive user interface generation process (Song & Cho, 2013)

Through the network the necessary devices depending on the situation are inferred; then the useful functions for each device are selected. Finally, the adaptive interface for a given controller is generated using the appropriate UI template.
The adaptive user interface with the fixed user interface by involving fourteen subjects has been compared. The results show that the generated interface was more usable than the fixed one. Wang et al., (2014) provide an adaptive user interface model based on the perceptual control theory; they use the perceptual control theory to extrapolate the user's intent, analyse the context of use and redesign the interface to satisfy the end user. The system is developed for mobile devices. In (2015) Feng and Liu propose an adaptive interface model for services based on a mobile location that contains three main sections: purpose, adaptation and adaptation. In the (2016) Rebai et al., Use learning algorithms and Bayesian network to provide a new AUIs for web applications. In detail, in this paper, the authors show how the Bayesian networks are a valid inference algorithm used to make predictions and how the adaptive user interface is preferable to the fixed one in terms of a 31% reduction in the time required to complete an activity. In order to improve the efficiency of human-computer interaction and reduce operating load, a new design method of adaptive GUI was proposed by Li et al., (2018).
Despite the existence of different methods for the adaptive user interfaces design, the literature on the search context of adaptive interfaces is very heterogeneous and closely linked to each application domain. In fact, most of the existing case study design ad hoc frameworks for the design of AUI, without considering the existing approaches.
This outlines the lack of a single effective methodology for the design of Adaptive interfaces, able to support the designer in the UI design phase, to identify the critical issues and finally to identify the best redesign strategy. For this reason, the present work would try to fill this gap.

2.4. Adaptive Interface Case Studies

The literature on the adaptive interfaces research context is very heterogeneous and closely linked with each domain of application. There are several contexts in which the adaptive systems have been used: health, smart environments, air traffic control, web applications, electrical network management and thermal plant supervision, warehouse picking system, web services, automotive environments, mobile device context etc., (***Fig. 19***).

Year \ Context	Smart Environments	Automotive Environments	Learning	Health	Augmented Reality	Mobile Applications	Complex Environments	Electrical Management	Industrial Machine-Rob	Web Applications	Geo-Visualization System	Work Smart Environments
2000		x		x			x	x		x		
2001										x		
2002			x	x								
2003			x		x							
2004			x								x	
2005				x								
2006	x		x			x						
2007				x						x		
2008										x		
2009							x					
2010		x				x				x		
2011		x	x	x		x				x		
2012	x											x
2013	x					x	x					
2014						x						
2015				x								
2016			x			x				x		
2017			x	x		x			x	x		
2018				x		x						

Fig. 19:Adaptive User Interfaces Application (since 2000)

In the Health Context, for example, Stephanidis et al., (1998) design and develop an adaptive interface for Web applications aimed at people with disabilities, in the context of the AVANTI project. In the (2000) Bental et al., provide a framework for the provision of computer-based information for cancer patients through the interactive system developing, which provide at the right time health information to the patient. The presented system selects and presents information that is adapted at different levels to the situational and process'-based aspects of the patient's illness and treatment. Furthermore, the adaptive user interface has been used in a hypermedia data browsing to develop a clinical workstation prototype for the cardiology domain: "HEMA" (***Fig. 20***). In detail, it has been designed for reviewing patients record (Ghédira et al., 2002). They presented two different instances of visualization of the prototype: on the basis of the user's type the system automatically displays the interface model associated with the group's profile. Any change will

trigger the modification of the stored user's profile. Through the use of this system, the authors recorded a satisfactory operation of the user profiles customization through dynamic changes of the group and user profiles.
Adaptive interfaces have also been developed for home health applications. Jimison et al., (2003) propose a method to measure the cognitive performance of users by monitoring their interactions with the computer. Information from interactions is used to dynamically adapt the user interface to various applications to improve the user experience. In the (2005) the adaptive user interfaces have been used to develop a framework for eHealth (Vasilyeva et al., 2005). In (2007) Nestler and Klinker develop mobile user-interfaces for the healthcare domain. In particular, they focus on the development of interfaces for the management of unpredictable and unstable situations. Minón and Abascal (2011) present the extension of an architecture developed for an AAL environment (AmbienNet) by inserting the adaptive component both on the instructions and on the interfaces generated to support the elderly at home. Finally, Shakshuki et al., (2015) provide a multi-agent system for tracking and monitoring health data for patients (such as heartbeat or blood oxygen levels) through a system architecture developing. In detail, the actions and behaviour of users are monitored and used to modify their respective user interface over time. Two scenarios have been used to test the system: patient and doctor.

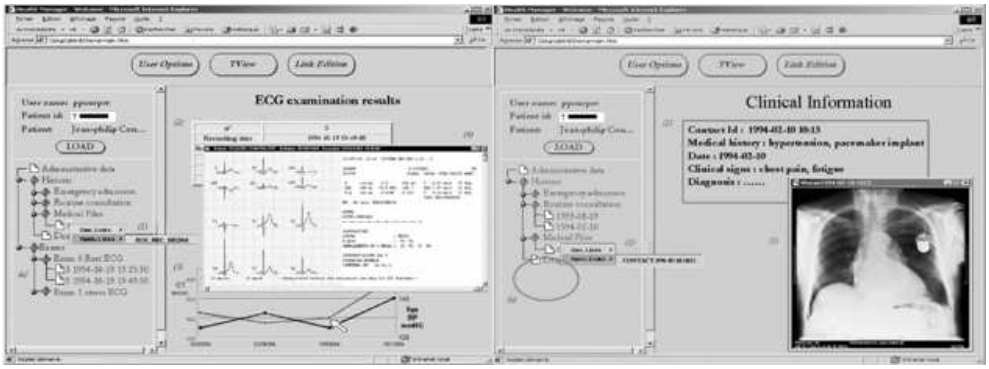


Fig. 20: HEMA workstation display a) a cardiologist health record consultation session; b) a GP health record consultation session (Ghédira et al., 2002).

In the Smart Environments Context, the adaptive user interface has been used to manage and control various devices simultaneously in a modern home theatre system including smart TV, DVD etc. In order to generate an adaptive user interface in the smart environment Namgoong et al. (2006) propose an adaptive user controller exploiting the representation of devices through semantic descriptions. The controller recognizes devices and dynamically constructs interfaces adapting to user

preferences. Furthermore, Song and Cho (2013) model the ubiquitous home environment through the use of a Bayesian network able to predict the custom devices in each situation (***Fig. 21)***.
In addition, the adaptive user interfaces have been used in a Work smart environments context, for a warehouse picking system (Bongartz et al., 2012). In particular, they exploit the model-based approach in supporting adaptation and develop a set of rules to provide the adaptation action. In a supermarket chain in the industries domain the system has been tested, through the GUI Application (***Fig. 22)***. A user study has been conducted to evaluate the rules on the end user point of view. The result of the user test showed in general that adaptive rules are received well if they trigger the delivery of additional information. Omitting some information without notification, however leads to some usability problem.

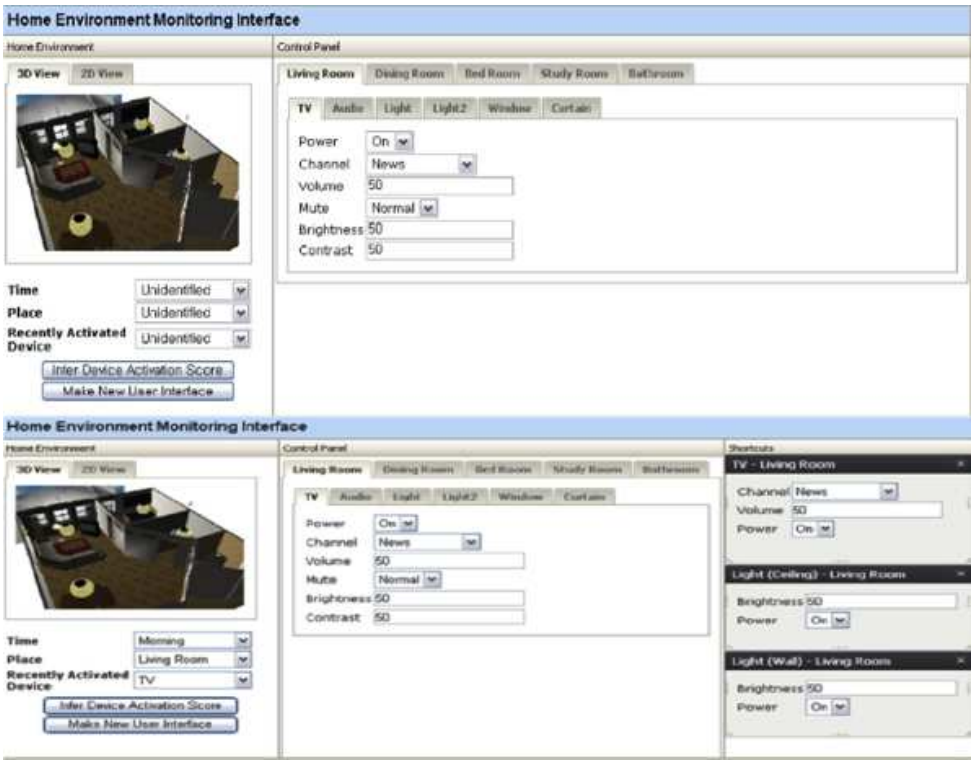


Fig. 21: A general user interface in a modern home network environment (above); the proposed adaptive user interface (Song & Cho, 2013)

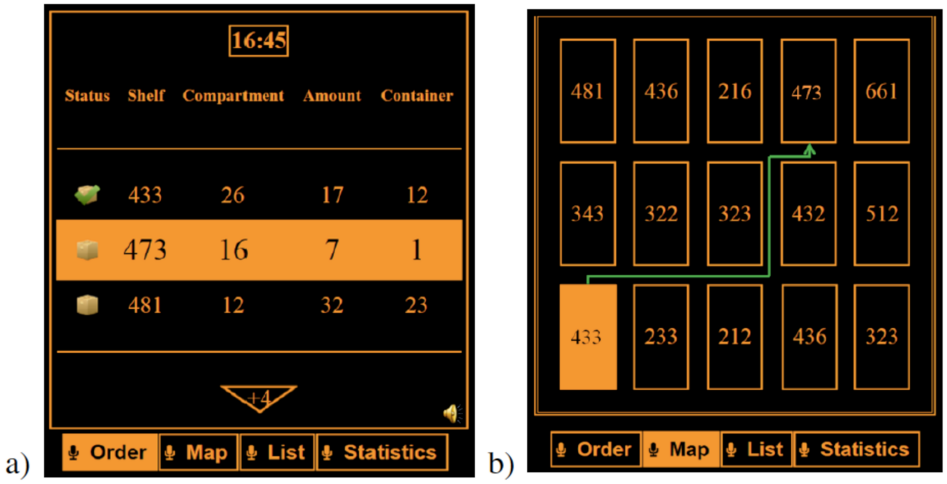


Fig. 22: A Graphical user interface. A) Order view; b) Map view (Bongartz et al., 2012).

Yen et al. (2009) provide a generalized framework for the UI adaptation in Complex Environments. In particular, the framework has been tested in a complex supervisory task: Air Traffic Control. In order to test the interface, a data set (time, number of mouse clicks etc.,) for each interface has been collected. In 2013, Evans et al., devised an adaptive interface for supervising control of multiple remote aircraft, introducing a new perspective for integrating adaptive user interface development.
In (2000) Ardissono et al., present the personalization features offered by SETA, a prototype toolkit for the creation of adaptive Web application stores, that suggests the items best fitting the customers' needs and adapt the layout and the description of the store catalog to their preferences and expertise. In (2007, 2008) He and Yen propose an adaptive user interface generation framework for Web Service Context.
El-Bakry et al., (2010) present a new adaptive interface design for the web application. In particular, they describe the guidelines, the architecture, the languages and framework for the web application interface design.
Moreover, Rebai (2016) uses Bayesian Networks to design an adaptive Social Network interface for web applications. To test the new approach the web interface that allows exchanging documents and courses between students and teachers has been tested. Ciolacu and Beer (2016) present an Adaptive User Interface (AUI) for online courses in higher education as a method for solving the challenges posed by the different knowledge levels in a heterogeneous group of students. Finally, a conceptual framework for developing an Adaptive User Interface for a web-based

application has been presented by Machado et al., 2018. In particular, the system aims to help seniors who have difficulty interacting with technology.
In the Electrical Network Management field and in the Thermal Power Plant supervision an innovative auto-adaptive multimedia interface (AAMI) architecture has been developed and implemented. The benefits of the AAMI approach has been evaluated (Viano et al., 2000).
In the Automotive Environments the adaptive user interfaces have been used to provide quick access to car information. For example, Rogers et al., (2000), used UIs to provide content and information to the driver and to record user driving preference. Nasoz et al., (2010) built an intelligent and adaptive car interface, able to recognize the driver's emotions and to communicate naturally. In particular, they tested the system in a virtual reality environment in which the main vital parameters of the user were collected (heart rate, temperature, HRS, etc.) during different driving experiences. Bayesian networks were used to draw conclusions and adapt to the context. Furthermore, in (2011) Tchankue et al., designed and integrated an adaptive interface into a conventional car dialog system to prevent the driver from receiving calls and sending messages, avoiding distractions.
In the Mobile Device Context, the adaptive user interface has been used to adapt a graphical interface to a various type of user (Mitrovic and Mena, 2002; Mitrovic et al., 2008). Hussain et al., (2014) provide a personalized user interface thanks to the continuous involvement of the user through web monitoring, feedback etc. The interface is adapted with the help of the user's user experience (***Fig. 23***).
Nivethika et al., 2013, propose a new architecture for a framework that provides an adaptive user interface for mobile applications. This framework includes data pre-processing step, learning step and execution and rendering step to deliver a suitable user interface. Furthermore, the framework is flexible and support user requirements through the configuration of a number of UI component. Wang et al., 2014 build an adaptive user interface model for mobile devices based on perceived control theory. In (2015) Feng and Liu propose an adaptive interface for services based on mobile localization. The process of functioning of the interface adaptation through user interaction is described. The interface applies the user's requests and suggests the route (***Fig. 24***).

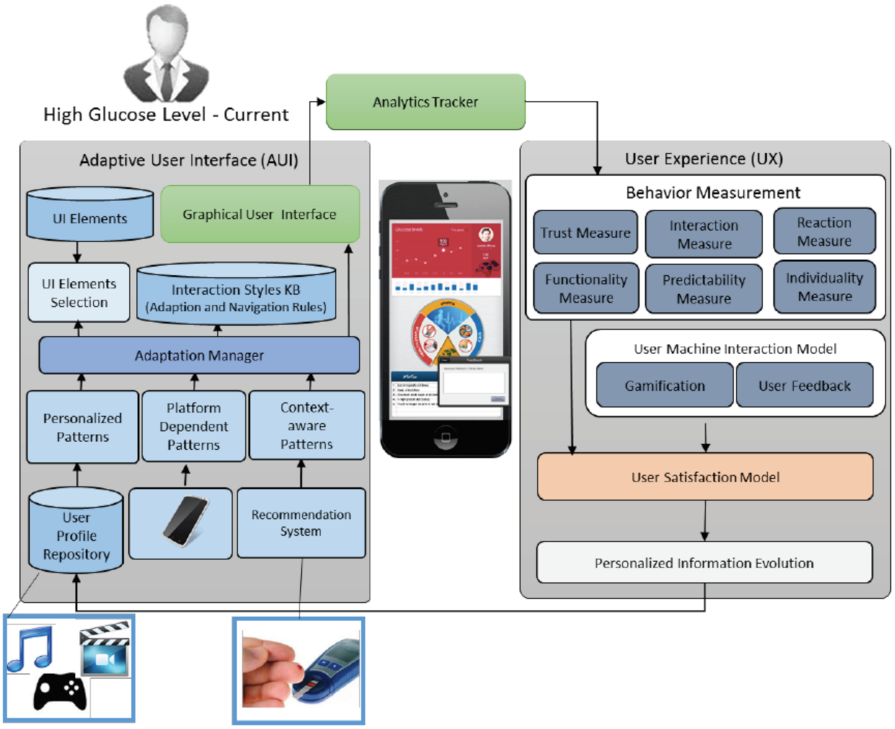


Fig. 23: The AUI workflow (Hussain et al., 2014)

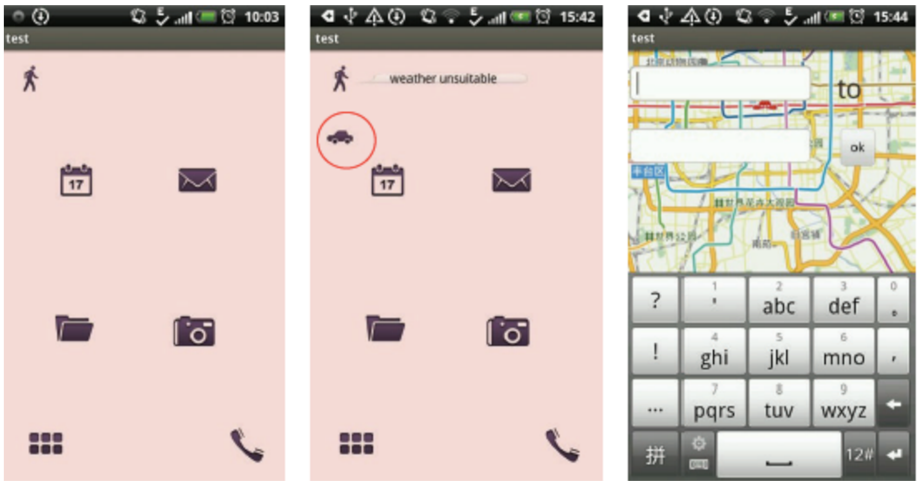


Fig. 24: The adaptive user interface for route searching (Feng & Liu, 2015)

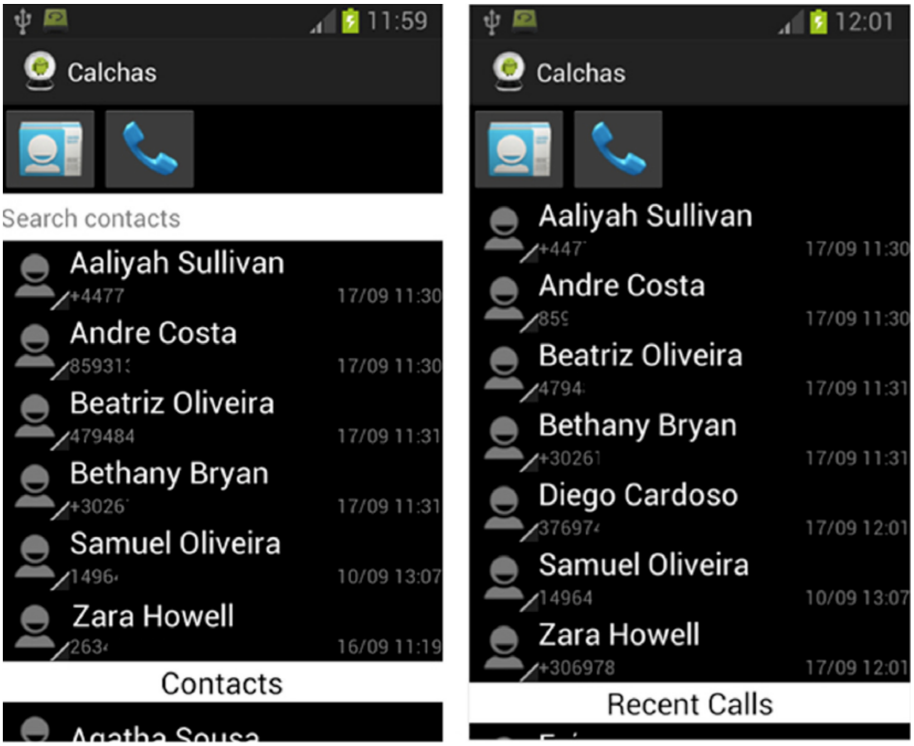


Fig. 25: The adaptive user interface presented by Plessas et al., (2017). "the list with suggested contacts (A–Z) followed by the embedded contact list (left) and the list with suggested contacts (A–Z) followed by the embedded list of recent calls (right)."

Furthermore, Plessas et al., (2017) implement an adaptive interface for a mobile contact recovery application. They develop a prediction algorithm and design interface that provide a modest benefit in users' ability to find contact, in cases of non-successful predictions (***Fig. 25***).

Finally, another important domain of application of adaptive interfaces is the field of Learning. Several applications have been designed to increase student learning (Jokinen et al., 2002; Muda et al., 2006). In 2003 developed a custom AUI to enhance the functionality of the originally encoded interface of Microsoft Word (Liu et al., 2003). In (2017) Hariyanto and Kohler sustain that a specific strategy is needed to provide a learning environment suited to different student preferences. For this reason, they design an adaptive e-learning system based on multiple learning criteria. In this context, this paper aims to enrich the state of the art by proposing new applications of adaptive systems particularly in the smart environment.

2.5. Adaptive Interface Evaluation

Although there exist today a number of established approaches and frameworks for the design and implementation of AUIs, their evaluation is yet to be addressed in a comprehensive way. In fact, there are no unique valuation methods in this field that provide adequate feedback in the AUI design process. Keppel et al., (1998) identify dependent and independent variables to be used during the evaluation of an adaptive system. In particular, they identify the independent variable as the range of treatment conditions under the control of the experimenter; the dependent variable, on the other hand, consists in observing the user's behaviour. Examples of dependent variables are:

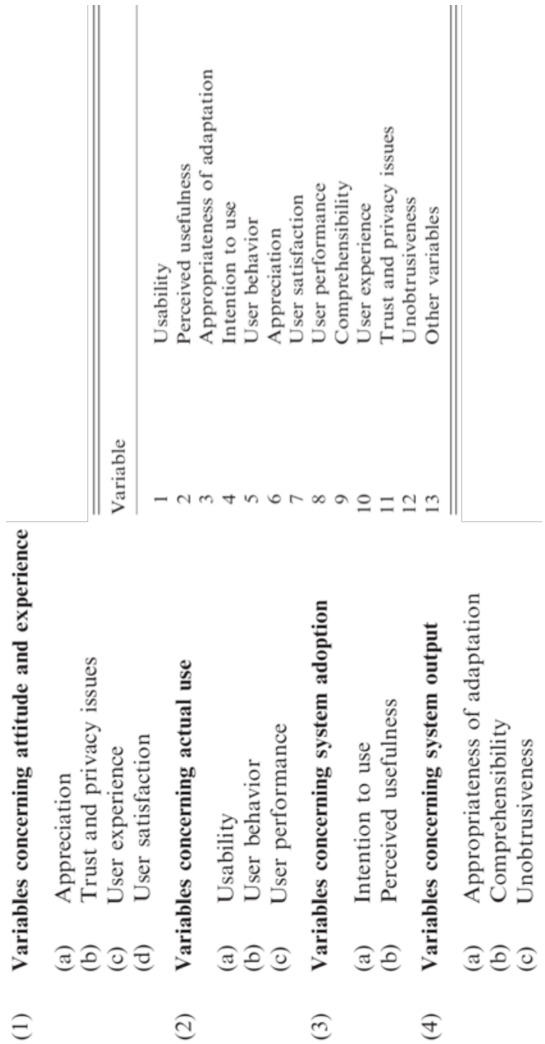
1. the task completion time;
2. the number of errors;
3. proportion/qualities of tasks achieved;
4. interaction patterns;
5. learning time/rate;
6. user satisfaction, etc.

In the same year Whiteside et al., provide a list of parameters used in usability metrics and useful as a dependent variable also in HCI experiments.

1. time to complete a task;
2. percentage of task completed;
3. percentage of task completed per unit time;
4. ratio of successes to failures;
5. time spent in errors;
6. percentage or number of errors;
7. percentage or number of competitors better than it;
8. number of commands used;
9. frequency of help and documentation use;
10. percentage of favorable/unfavorable user comments;
11. number of repetitions of failed commands;
12. number of runs of successes and of failures;
13. number of times interface misleads the user;
14. number of good and bad features recalled by users;
15. number of available commands not invoked;
16. number of regressive behaviours;
17. number of users preferring a specific system;

18. number of times users need to work around a problem;
19. number of times users are disrupted from a work task;
20. number of times users lose control of the system;
21. number of times user's express frustration or satisfaction.

Another used technique is the comparison between adaptive and non-adaptive system in terms of user satisfaction (Park et al., 2007; Park and Han, 2011; Plessas et al., 2017); adaptive support effect (Brusilovsky and Eklund, 1998; Brusilovsky and Pesin, 1998), user performances (Greenberg and Witten, 1985), usability (Liu et al., 2003) and efficiency (Findlater and McGrenere, 2004).
In the (2013) Van Velsen et al., provide a literature review focuses on the main evaluation methods of adaptive and adaptable systems. They highlight a total of 44 variables attributable to 4 different concepts (***Fig. 26***).
Another approach in the AUI evaluation has been put forward by Paramythis et al., (2001) that provide a method to separately evaluate each adaptive system constituents through the decomposition of adaptation "from the perspective of HCI-oriented evaluation": adaptation is no longer considered as a singular process, but a set of different components: interaction and monitoring; inferences; explicitly provided knowledge; modeling; applying adaptations and automatic adaptation assessment.



			Variable
(1)	**Variables concerning attitude and experience**		
	(a)	Appreciation	1
	(b)	Trust and privacy issues	2
	(c)	User experience	3
	(d)	User satisfaction	4
(2)	**Variables concerning actual use**		
	(a)	Usability	5
	(b)	User behavior	6
	(c)	User performance	7
(3)	**Variables concerning system adoption**		
	(a)	Intention to use	8
	(b)	Perceived usefulness	9
(4)	**Variables concerning system output**		
	(a)	Appropriateness of adaptation	10
	(b)	Comprehensibility	11
	(c)	Unobtrusiveness	12
		Other variables	13

Fig. 26: Main variables used for the evaluation of interfaces (Van Velsen et al., 2013).

Then, they identified seven evaluation modules (composed of one or more of the adaptation components listed above), which can be evaluated individually and in combination on the basis of the evaluation goals. Similarly, Weibelzahl (Weibelzahl, 2001), proposed an evaluation framework made up of different steps, as shown in the figure below (***Fig. 27***).

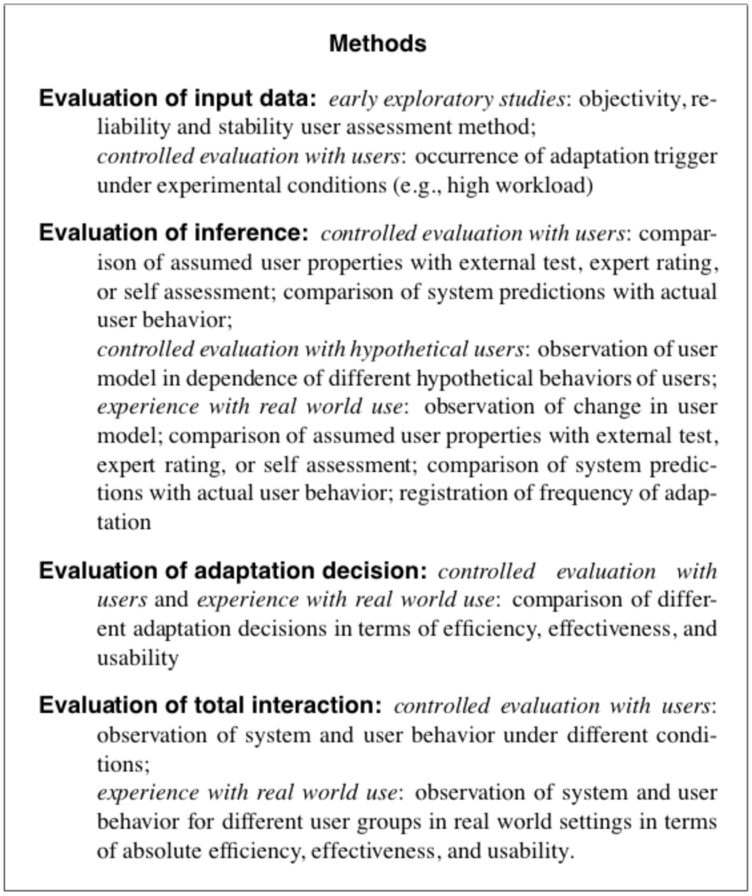


Fig. 27: An Evaluation Framework for Adaptive Interfaces

Other approaches concern the adaptive interface evaluation in terms of Cost and benefit (Gajos et al., 2006), user's error (Yen and Acay, 2009; Zudilova-Seinstra, 2007), cognitive workload predictability and accuracy (Gajos et al., 2008), and accessibility (Lin et al., 2017).

Chapter 3.

Adaptive Interface Methodology

3.1. Introduction

Designing an adaptive user interface means to define an interactive system able to manage its knowledge about the user (i.e., who is using the system) and the environment (i.e., the context in which the user-system interaction takes place), in order to provide information content, functions, and interaction modalities, in the most adequate way, according to different users and context of use. To this end, a method that leads to the single instance construction in the user interfaces design is inappropriate as it cannot accommodate the differences resulting from the needs of individual users. Therefore, the requirement of a systematic process (flow) which may include alternative decisions making procedures able to accommodate the resulting diversity of individual users, is necessary (Savidis, 2004). To achieve this objective, it is necessary to adopt a design approach which supports the definition of the polymorphic design solution, for each system functionalities, so that it allows the definition of how the system should support different users in different contexts of use.

For this purpose, in order to develop an adaptive smart system, a new design methodology to adaptive interface has been developed.

According to this, the proposed Adaptive Interface Methodology is illustrated and explained in detail in the following paragraphs. Three fundamentals model characterize it:

1. The ***User Model*** **(UM)** provides a description of the user's profile pattern, according to its cognitive and physical structure, status and preferences.
2. The ***Environment Model*** **(EM)** supplies the information pattern necessary to describe the human machine interaction environment.
3. The ***Interaction Model*** **(IM)** consists in the definition of the adaptation goals and rule, the definition of the interaction and adaptation variable and in the adaptation mechanism choice. It is the core of the whole adaptive

process: this model oversees the user and environmental model data management.

Fig. 28 shows schematically the information flows from one to another module to generate an Adaptive User Interface.

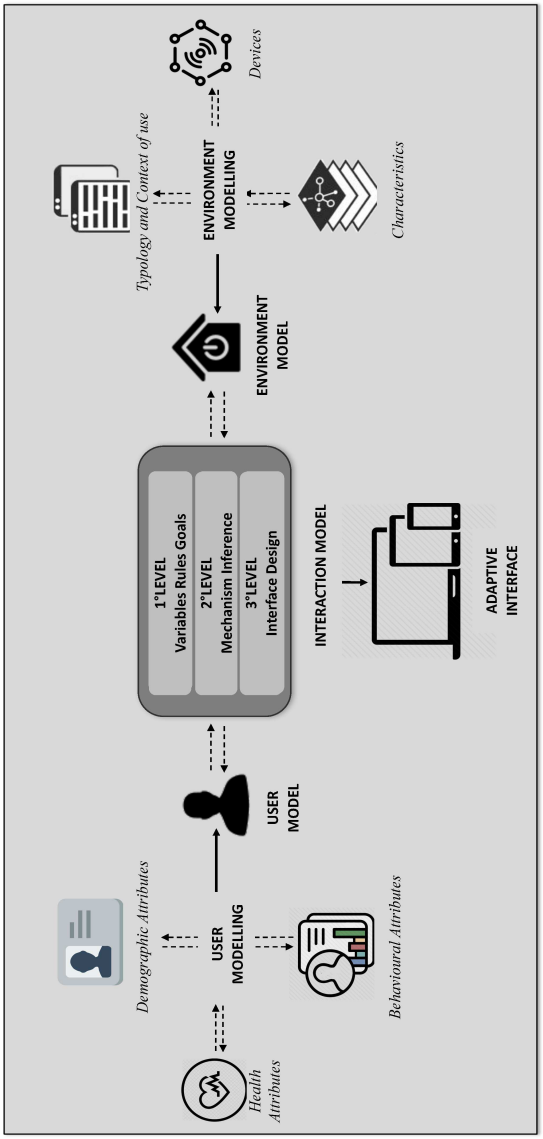


Fig. 28: The Adaptive Interface Methodology

3.2. User Model

The evolution of computer technology has led to new challenges and new problems for user modelling (Razmerita & Lytras, 2008). In the last few years, user modelling has been widely applied to interactive and adaptive systems, or into systems that suggest resources based on the user profile. In general, these systems need to change their behaviour according to user characteristics (Kobsa & Pohl, 1994). One of the most important aspects needed in building such systems is the user's information: their goals, their needs, moods, preferences, intentions, etc., (Murray, 1990). This information can be acquired by an explicit (Pannu, et al., 2011) or implicit (Shen, et al., 2005) process called user modelling.

User modelling means the interpretation of the user's actions, for a software system (Sosnovsky & Dicheva, 2010). The collection of user information used for service customization is the profiling process. In general, two are the main problems in the user modelling: firstly, understand what are the most significant user's characteristics that need to be considered in a software system; secondly investigate how to acquire the user information through a system (Rich, 1979). The user model represents user's features useful to customize interaction with the system. The user modelling process can be described as a sequence of three steps taken during user interaction with the system (Brusilovsky P. , 1996): collecting data, in this phase in which the system collects all the information concerning the user; inference, at this stage, the system processes the previously collected data and performs the user's profiling by classifying his interest features , preferences and objectives; adaptation, this step is the actual procedure of the user model aimed to provide personalized experiences. As shown in (Scime, 2005) there are two types of user modelling approaches: behavioral-based and knowledge-based. The knowledge-based approach for modelling considers the level of user's knowledge identified through questionnaires presented to users (Kobsa, 1990). The behavioral-based approach instead considers the user's behaviour observed during the interaction with the system (Scime,2005). There are various methods of representing information in a user model; the main approaches used are the Stereotype-based and Feature-based (Brusilovski & Millán, 2007). Summarize the user model is a representation of the characteristics of the users and is a very important feature in the user modelling in an adaptive interface (Langley, 1999).

In this work, the User model (UM) is constructed by using a hybrid perspective. It starts with the user modelling that consists of the representation of the user

characteristics through incomplete descriptions of interests and preferences, an approach based on stereotypes. For each stereotype, characteristics and objectives are defined. Subsequently following a feature-based approach the dynamic aspects linked to preference changes and interest are modelled (Brusilovski & Millán, 2007). By using a top-down approach, the user model is defined through the definition of user characteristics.

User characteristics include three user information domains: demographic, behavioral and health-related (***Fig. 29***). Such information can be acquired by the system by making inferences about users from their interaction (Rich, 1983) (Rich, 1989), by carrying out some from test, or from assigning users to generic user categories usually called 'stereotypes', by making explicit acquisition through asking relevant question, or by making previously stored information (Murray, 1990).

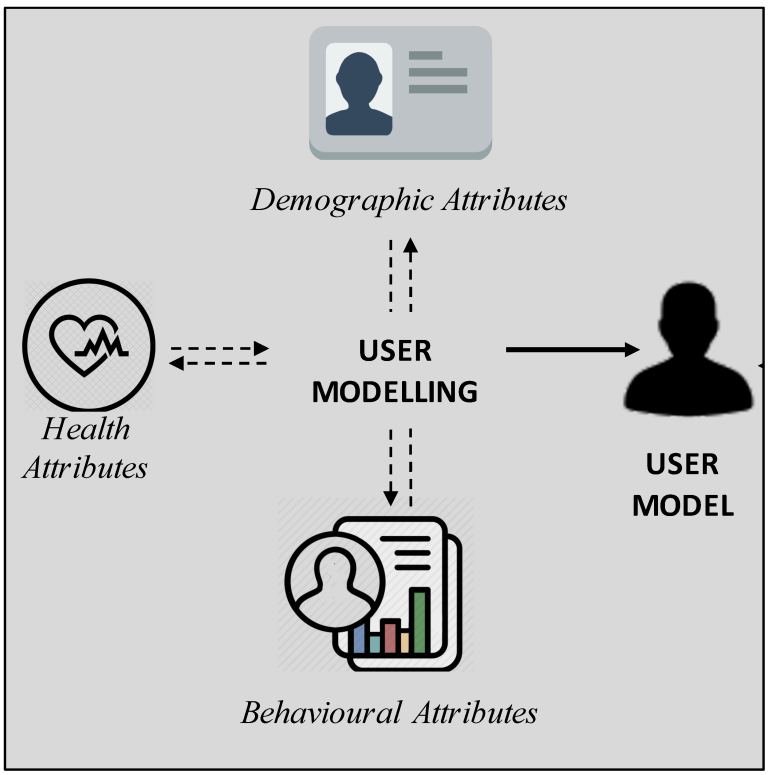


Fig. 29: The User Modelling

3.2.1 Demographic attributes

Demographic attributes concern the user's personal and socio-economic characteristics. The user's personal information includes age, gender, name, address, etc. The socio-economic characteristics include five classes and it is important to determine the level of expertise possessed by the user in performing the activities/ tasks that have to be considered as a prerequisite for the use of the product itself (e.g., swimming, using the smartphone, etc.). The classes are: (1) education (e.g. master, degree, etc.), (2) occupation (e.g. student, engineer, office worker, etc.), (3) family role (e.g. father, mother, etc.), (4) skill (e.g. cognitive, management, relational, personal effectiveness, etc.) and (5) conjugal status (***Fig. 30***).

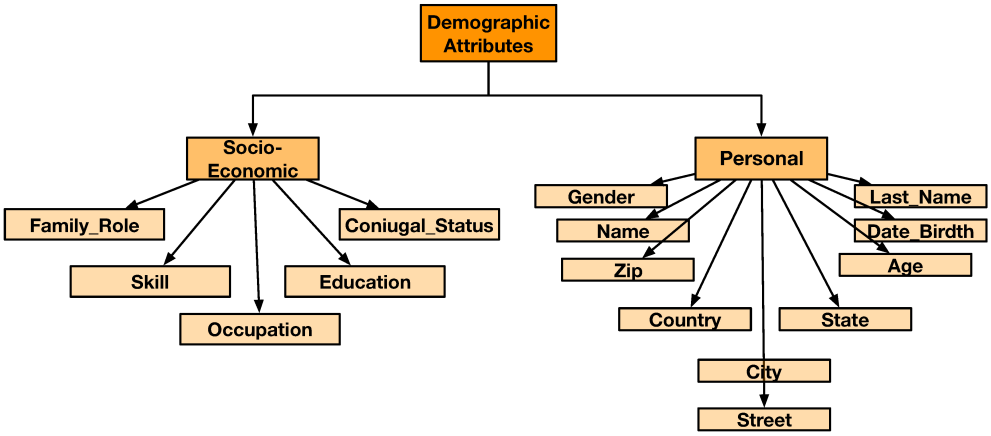


Fig. 30: Demographic Attributes

3.2.2 Behavioral attributes

The behavioral attributes refer to user's preferences and interest, which are important to determine the user favourites (e.g., "loves cats", "likes blue colour" or "dislikes classical music") and user hobby or work-related interests (e.g., "interested in sports", "interested in cooking. In addition, behavioral attributes include context information regarding the user's information needs, the history of user interaction with the system, information about the type of context in which the user is located (the kitchen, the living room, an external environment) etc., (***Fig. 31***).

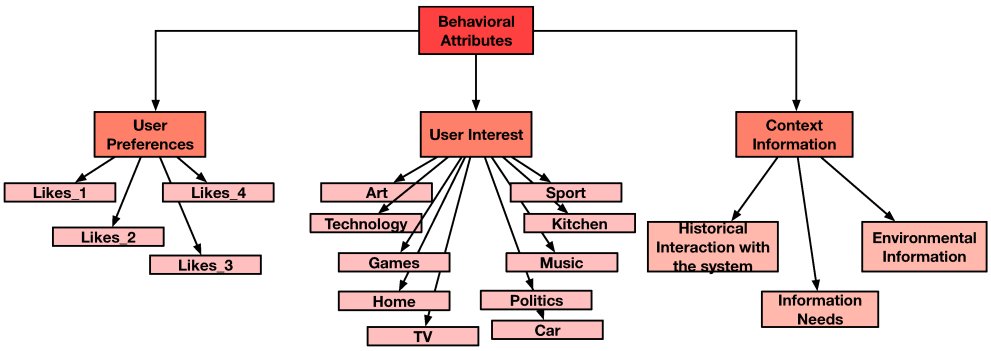


Fig. 31: Behavioral Attributes

3.2.3 Health attributes

Health-related attributes aim to define the spectrum of abilities of the individual, according to his/her health status (e.g., motor abilities, sensorial abilities, etc.). In particular, the health information includes two different classes Physical Factor (e.g. weight, height, etc.,) and Physical Abilities & Disabilities which is divided into Cognitive, Perceptual and Motor abilities. In particular Cognitive Abilities includes: perception, recognition and interpretation of sensory stimuli (smell, touch, hearing, etc.); attention, ability to sustain concentration on a particular object, action, or thought, and ability to manage competing demands in our environment; memory, short-term/ working memory (limited storage), and long-term memory (unlimited storage); language, skills allowing us to translate sounds into words and generate verbal output, visual and spatial processing, ability to process incoming visual stimuli, to understand spatial relationship between objects, and to visualize images and scenarios, executive functions, abilities that enable goal-oriented behaviour, such as the ability to plan, and execute a goal. Perceptual Abilities include seeing, hearing etc. Finally, Motor Abilities according to Fleishman's Taxonomy (Fleishman et al., 1991) include control precision, rate control, aiming, response orientation, reaction time, manual and finger dexterity, arm-hand steadiness, wrist and finger speed (***Fig. 32***).

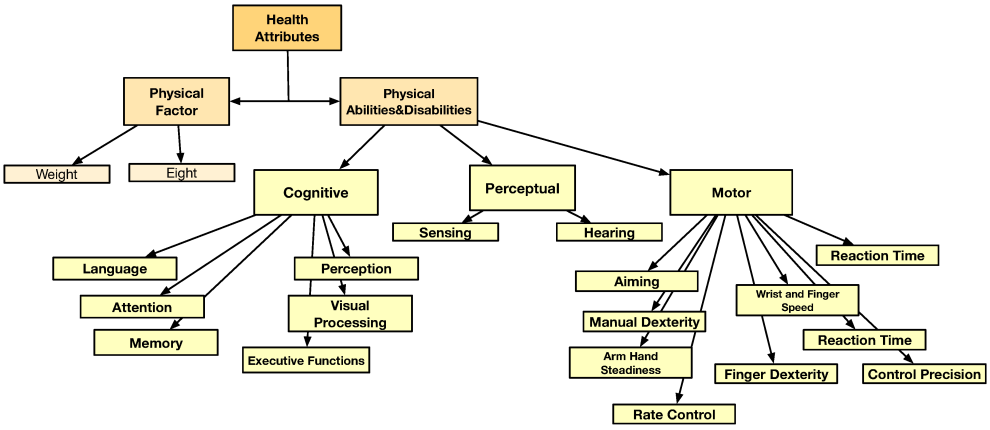


Fig. 32: Health Attributes

User profile models sourced from bibliography were also considered and derived concepts were appropriately adapted and included in the user model.

3.3. Environment Model

The environment model (EM), also called domain model (Benyon & Murray, 1993) (Puerta & Eisenstein, 1998), supplies the information pattern necessary to describe the human-machine interaction environment and defines the application aspects which are required for the system operation. In particular, it is a description of the system that contains information about the domain, i.e. the objects, their attributes and the relationships between objects (Benyon & Murray, 1993). In general, the EM consists of a system abstraction, which serves to define all the aspects that make the adaptive system (Cockton, 1987). Indeed, the environment modelling is a primary process in defining inferences and forecasts of user-system interaction for the designer development (Benyon & Murray, 1993). Puerta et al., define this model such as "*a model that defines the objects that a user can view, access, and manipulate through a user interface.*" (Puerta & Eisenstein, 1998).
In this work, the Environment model describes the human-machine interaction environment and is constructed by using a bottom-up approach. It starts from the environment modelling that consists of the definition and characterization of its typology, physical characteristics and devices (***Fig. 33***).

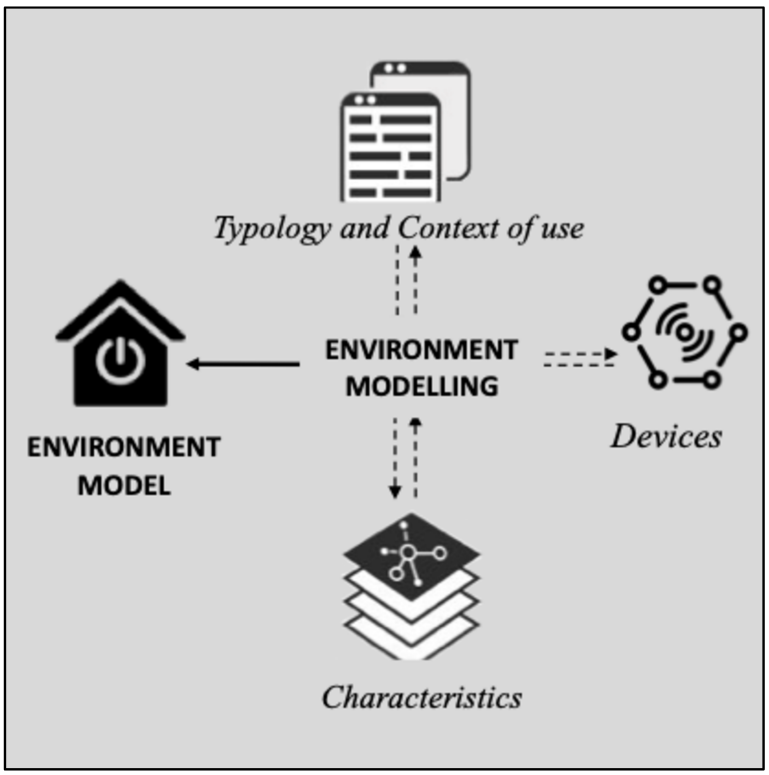


Fig. 33: The Environment Modelling

3.3.1 Typology and Context of use

This level describes how an environment is structured in terms of typology and context of use. In detail environment analysis covers the description of the environments within which the system is to operate. This level is important because the user needs to be aware of the system purpose and context of use.

In terms of type, environments can be classified into laboratories, offices, support environments (used environments), service environments (warehouses), domestic environments (kitchen, living, etc.) and smart environments. The definition of the environment typology is fundamental for the definition of the tasks and for the definition of technical and environmental constraints.

3.3.2 Devices

The device term means the device type that populates a determined environment. These differ from each other by type, function, controlled parameters etc. For this reason, they were first classified according to their membership category.

Through the use of clustering algorithms, devices have been classified for categories and sub-categories (i.e., health, sport, home, etc.) to define their domain of action. Subsequently, a depth analysis to identify the specifics of the devices has been carried out. In order to accomplish the task for which it was designed, the device performs one or more functions. They have been classified in the following four groups: actuating (i.e., ability to transmit data to actuators to control a system remotely); event (i.e., ability to manage a scenario as, for example, by notifying to another device that a certain threshold value has been exceeded); meter (i.e., ability to monitor the value of a specific attribute by getting data from a meter) and sensing (i.e., ability to detect a specific event). In relation to these functions the parameters that can be controlled (i.e., status parameters such as on, off, open, close, etc.), measured (e.g., activity parameters such as steps, speed, sleep time, etc.; ambient parameters such as temperature, humidity, air quality, etc.; health parameters such as heart rate, blood pressure, SpO2, etc.) and detected (e.g., fall, gas leakage, intrusion, etc.) has been identified and classified.

According to its nature, each device is able to communicate with other devices, enabling a set of specific functions. For this aim, the communication protocols (both wired and wireless), the compatible devices (smartphone, tablet, laptop, desktop), the compatible Operating System (e.g., android, iOS, windows phone, etc.) and the applications have been mapped (***Fig. 34***).

Another important aspect concerns the technical specifications, which distinguish each device: they include operating conditions (e.g., temperature range), power requirements (e.g., source, autonomy, charge time, etc.), physical characteristics (e.g., dimensions, colour, material, etc.), the data management formats and warnings. Moreover, all the characteristics of the main components are considered such as the display resolution, the memory capacity, CPU frequency, sensors (e.g., accelerometer, gyroscope, altimeter, etc.), etc. Also, the specifications related to the measured parameters are considered: accuracy, range, resolution, unit, etc.

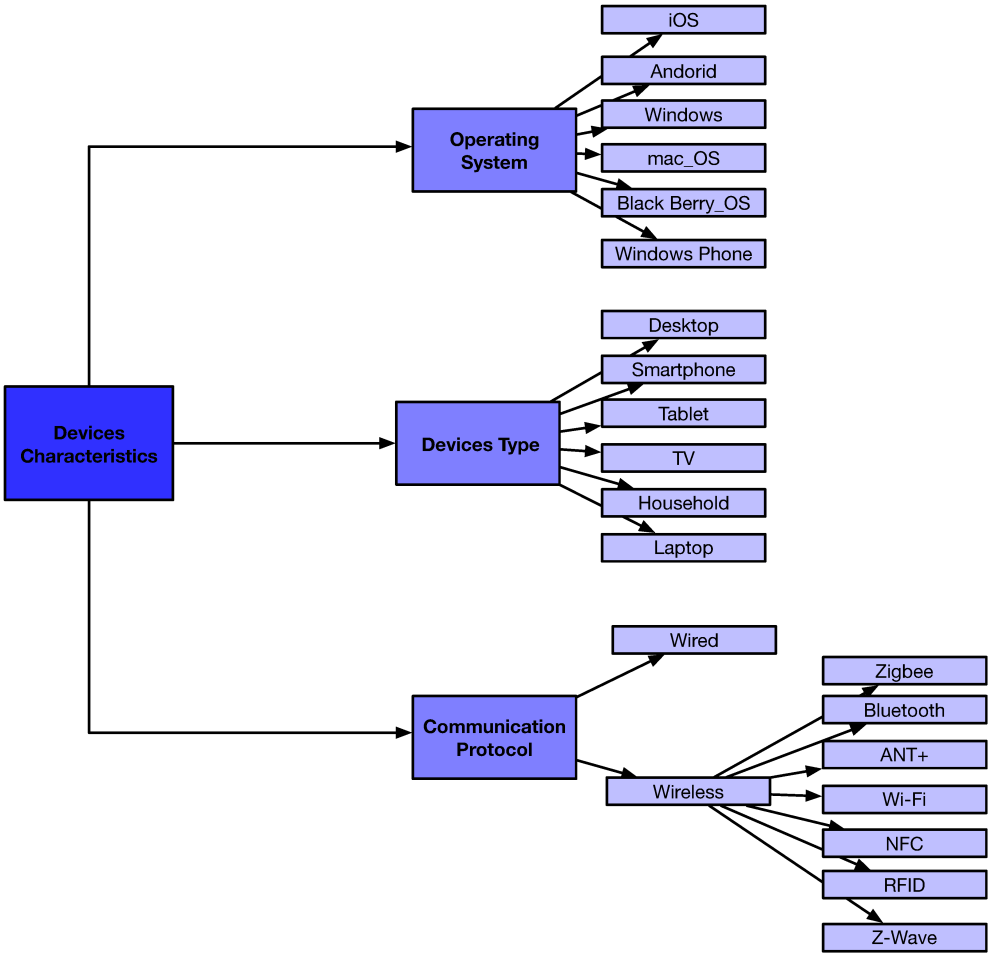


Fig. 34: Devices Characteristics

The class of unit of measure collects the standard for measurement of a quantity (***Fig. 35***).

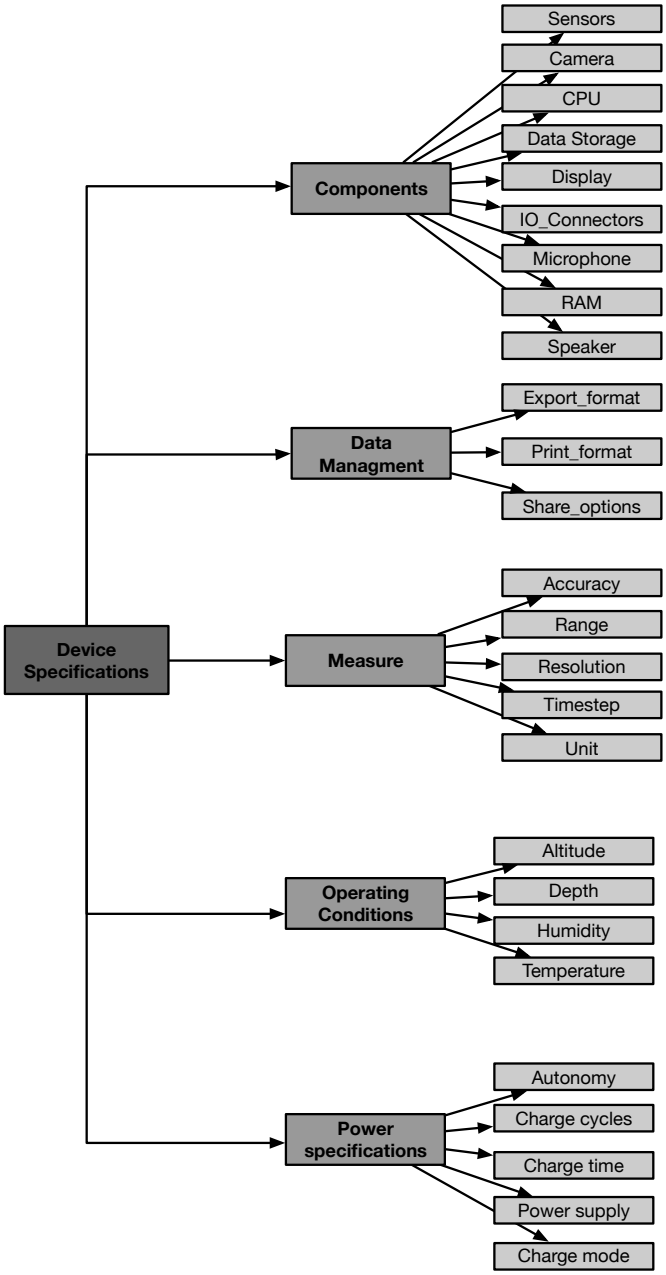


Fig. 35: Device Specifications

In addition to the main function, the Device can offer a set of more general **functionalities** grouped in ten classes (***Fig. 36***). They refer to the geolocation (e.g., GPS); the user management (e.g., multi-user, automatic user recognition, etc.); time and date (e.g., time of day, calendar, etc.); settings (e.g., wizard, customized thresholds, personalized scenarios, etc.); data management (e.g., export, share, etc.); the interaction with compatible devices (e.g., answer to phone call, read a message, play music, etc.); the interaction mode with user (e.g., gesture, touch, voice over, etc.); the power management (e.g., battery status indicator, power saving mode, etc.); the user training in relation to the main function of the object (e.g., goals to reach, rewards, personal assistant, etc.), notifications (e.g., alerts, alarm, message received, incoming call; etc.).

Different channels (e.g., audio, video, haptic) can be exploited by the SO to communicate information or give feedback to the user. For this aim, the class related to the interaction mode has been created.

Finally, a set of services has been identified according to the main function of existing devices in order to simplify the matching with the users' needs. The main goals for which these products are designed refer to the possibility to control, improve, monitor, save or manage some aspects of people daily life in several contexts (e.g., office, home, outdoor, etc.).

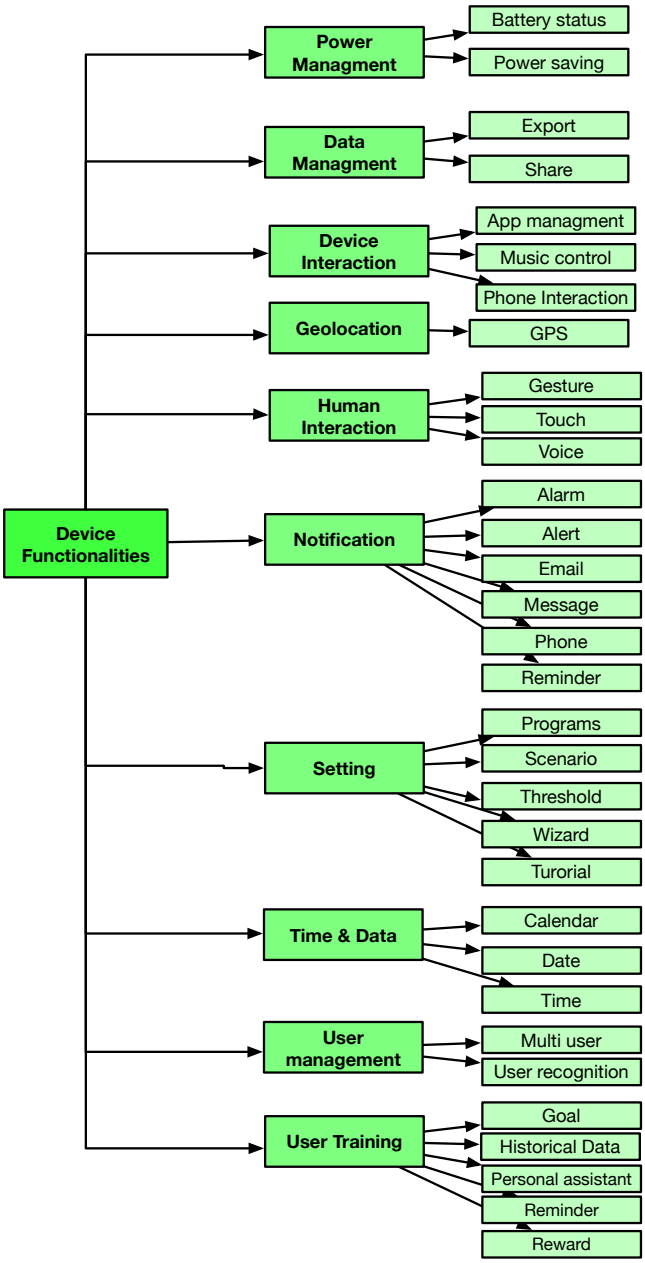


Fig. 36: Device Functionalities

3.3.3 Physical Characteristics

An environment is an orderly grouping of interdependent components linked together according to a plan to achieve a specific objective. This last level involves the study of the main environment features such as the physical features as well as the relationships between elements.

In particular, when defining the characteristics of an environment it is important to pay attention to the practical and functional aspects of the environments such as living the house, for example the furniture must have the right size, the spaces must be respected, the furniture must be arranged in a comfortable and functional way, the materials and technologies used must be of good quality, there must be no potential health hazards, must be a good sound insulation, a good relationship between energy consumption and comfort, and finally the whole environment must be in harmony between the clutter of full spaces and the use of empty spaces. In general, the main characteristics of an environment can be summarized in six points (***Fig. 37***).

- Organization: An environment should have structure and order. It is the arrangement of components that help to achieve objectives.

- Design: The main goal of environmental planning is the well-being of the human being in the living environment.

- Interaction: In an environment, the term interaction refers to the manner in which each component functions with other components of the environment.

- Integration. In an environment, the term interdependence means that parts of the organization or environment depend on one another. They are coordinated and linked together according to a plan.

- Objective: An environment should have a central objective. Objectives may be real or stated. The important point is that users must know the central objective of an environment early in the analysis for a successful design.

- Technical and environmental constraints: An environment is also defined on the basis of the existing elements or features (technical and environment) that cannot be modified (i.e., constrained components) and existing requirements that can be changed (i.e., editable components).

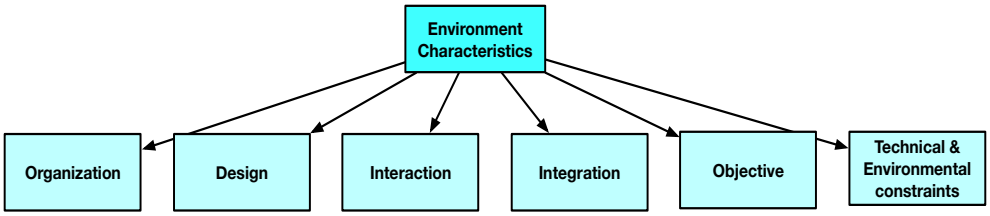


Fig. 37: Environment Characteristics

3.4. Interaction Model

The third component of an adaptive system is its representation of the actual and designed interaction between User and Environment: The Interaction Model (IM). The Interaction Model is the core of the whole adaptive process; as a matter of fact, this model is in charge of the user and environmental model data management. ***Fig. 38*** shows the mode of operation of the Interaction Model. Benyon et al. define an interaction as a user making use of the system at a level which can be monitored (Benyon & Murray, 1993). The IM contains everything which is concerned with the relationships existing between the user model and the environment model; from the data gathered, the system can make inferences about the user's beliefs, plans and/or goals, long-term characteristics, such as cognitive traits, or profile data, such as previous experience. The system may tailor its behavior to the needs of a particular interaction or, given suitably 'reflective' mechanisms, the system may evaluate its inferences and adaptations and adjust aspects of its own organization or behavior. For this reason, the IM have to recognize the user, store its needs and preferences.
In addition, it must be able to extract human-computer interaction information, provide the correct logical and task interpretation, allow a much more suitable environmental usability and define the event activation schedule.

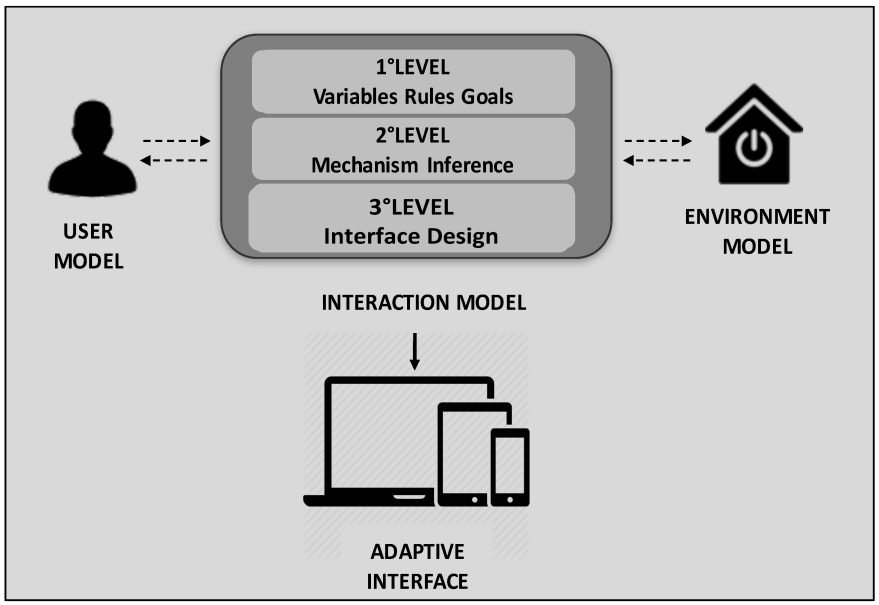


Fig. 38: The Interaction Model

Such complex adaptive systems require inference and evaluation mechanisms, which need to learn and interact with the local environment.

In the formal logic-based systems, it is assumed that available information and conclusions resulting from inferential rules are simply true or false. In an adaptive mechanism, there is no direct access to the whole domain reality; the system to be developed must act within a range of uncertain data: such as unreliable, missing and inaccurate data. In addition, incorrect environmental data may raise inaccuracy. Probabilistic theories provide the methods to correctly deal with inaccurate data systems, resulting from a lack of domain knowledge. The Bayesian approach, for example, provides a robust theory that merges together different technologies. Its approach to probability may be extended to the learning aspects.

Merging the results is a very powerful theory that provides a general solution for learning and optimal forecasting. Automatic learning is based on the idea that experience may improve the" agent" capability behaviour and future events, providing the ability to automatically update users' profiles and predict user's behaviour.

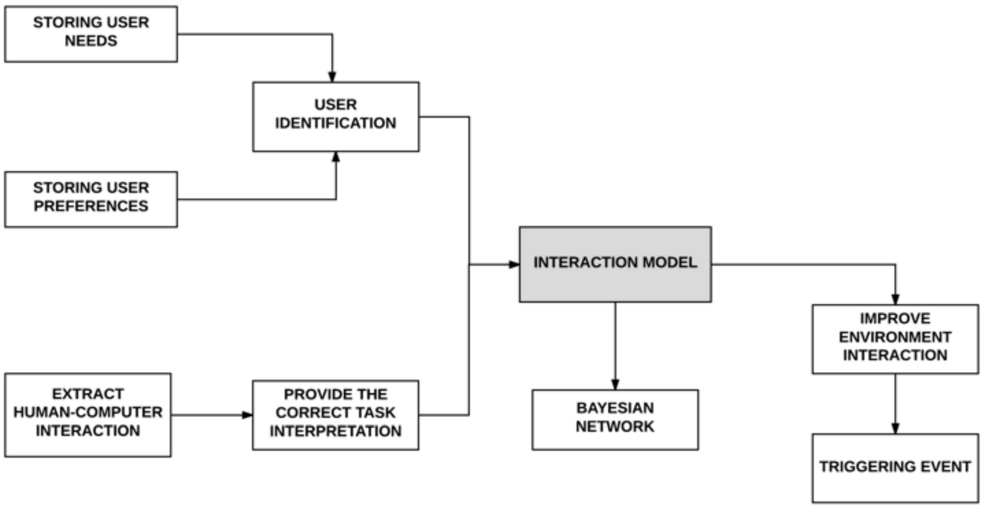


Fig. 39: Interaction Model function

The interaction model is based on three different levels:

1. First level: Definition of Variables, Rules and Goals;
2. Second level: Definition of Mechanism Inference;
3. Third level: Design Interface

3.4.1 The first level

The first level consists in the definition of a set of attributes dealt with in literature and focused on the adaptation strategy as a decision-making process, which answer to the following questions: Where? Why? What? To what? How? (Dieterich, et al., 1993) (***Fig. 40***).

- The first attribute identifies the application areas in which the adaptive system can be helpful (Where?). There are several contexts in which the adaptive systems have been used. In the health context, for example, the adaptive interface has been used to develop a clinical workstation prototype for the cardiology domain (Ghédira, et al., 2002) to develop an interactive system which provide at the right time health information to the patients with cancer (Bental, et al., 2000), to track and monitor health data (Shakshuki, et al., 2015), to develop framework for eHealth (Vasilyeva, et al., 2005). In the smart environments' context, the adaptive user interface has been used to manage and control various devices simultaneously including smart TV, DVD etc. (Song & Cho 2013; Namgoong et al., 2006;

Bongartz, et al., 2012). Moreover, in the mobile device context, the adaptive user interface has been used to adapt a graphical interface to different user type (Mitrovic & Mena, 2002; Nivethika, et al., 2013; Wang,et al., 2014). In addition, other adaptive interface applications relate to contexts such as: air traffic control (Yen & Acay, 2009), Semantic Web (Hervás & Bravo, 2010), Web Applications (El-Bakry, et al.,2010), Warehouse picking system (Bongartz, et al., 2012), etc.

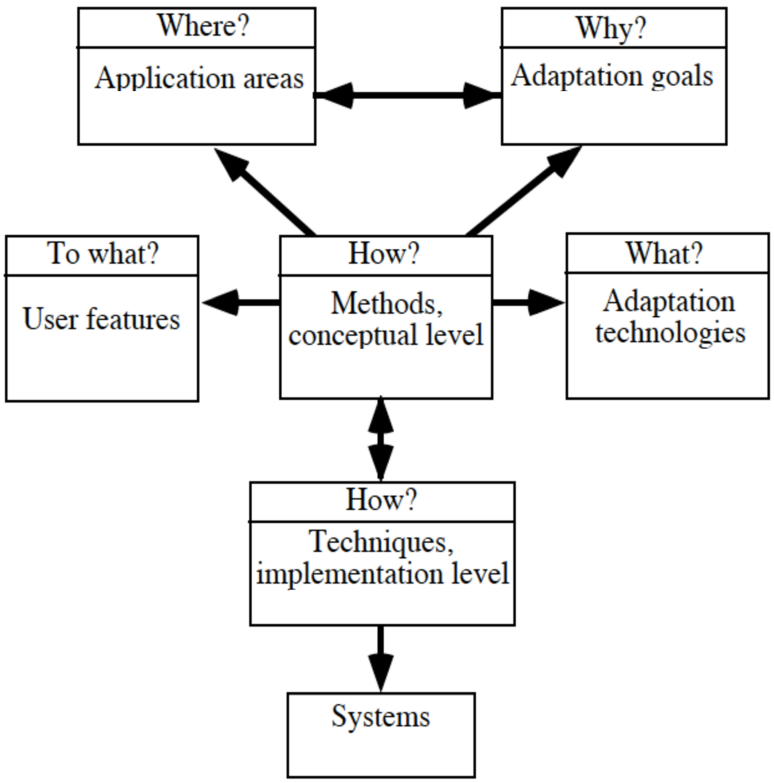


Fig. 40: Classifications for Adaptive System methods and techniques (*Brusilovsky P., 1996)*

- The second attribute identifies the reason to make an adaptive system (Why?). Depending on the context and on the system type, the "**adaptation goals**" concern a search goal, a problem-solving or learning goal. In general, the final adaptation goal depends on the user and have to be decided by the

designer. Possible examples of adaptation goals are easy, efficient, effective use etc.

- The third attribute refers to what aspects are subject to adaptation in a particular system; in detail, what features undergo a change depending on which users (What?). Usually, adaptation may refer to semantics, syntactic (i.e. sequencing of the interaction) or lexical aspects of interaction (i.e. interface objects, interaction techniques, media and modalities) (Stephanidis, et al., 1997). Dietrich et al., define these characteristics as "**adaptation constituents**" and provides a list of those usually employed in existing systems; this list includes generic functions (error correction, help, etc.,) and interaction level access to capabilities, task simplification, system-user information, etc.,) (Dieterich, et al., 1993).

- The fourth attribute refers to what aspects of the user consider making adaptation: to what? In general, the adaptation depends on the user's characteristics and preferences, on the user's goals, on the user's background and on the utilization context (Brusilovsky, 1996).

- The fifth attribute identifies how to make the adaptation. Defining a set of rules that guide the adaptive process by assigning specific adaptations to specific behaviours is a key point in the adaptive process. Stephanidis et al., define these attributes with the term "**adaptation rules**" (Stephanidis, et al., 1997).
- Finally, an additional dimension has been added to those highlighted by Benyon: when to adapt? Determining the right time for adaptation is a crucial point in the adaptive process. The adaptation can be carried out in run-time (during the use of the system) or statically (at the first access of the system). Stephanids et al., define these attributes as "**adaptation determinants**" (Stephanidis, et al., 1997).

3.4.1.1 Variables

Defining which aspects of the System require adaptation is the first step in the adaptive process. These attributes are defined "adaptation constituents" (Karagiannidis, et al., 1995) (Stephanidis, et al., 1997) and may address the semantic (i.e. information content), syntactic (interaction sequencing) or lexical level of interaction (interface objects, interaction techniques or media and modalities).

Variables requiring adaptation can be classified according to their physical origin: from the user, from the current situation and from the task. Variables can be defined as dynamic, which refers to the change during task execution, or static (Rothrock, et al., 2002). The majority of calling variables deal with the user. On the other hand, the main adaptation goal is to improve user performance. For this reason, the user is the most variability subject and the user performance is one of the most used criteria to trigger adaptation. Hettinger et al., sustain that some inputs are related to the behavior of the user, and some reflect the user's internal physiological state (Hettinger, et al., 1996). To support the real-time adaptivity, the whole interaction of the user (keys presses, gesture, mouse clicks or movements) with the system must be recorded.
In particular, according to Rothrock's methodology, the main factors that trigger adaptation are the following (***Table1):***

- **User Performance** represents the user error percentage in performing a task, as well as the time required to perform the task (Rouse, et al., 1987). Existing examples include computer/tablet/smartphone data entry speed, errors number, reaction time and tracking deviation. (Yen & Acay 2009), (Rothrock, et al., 2002). The lack of deterministic methods to measure user performance in terms of cognitive tasks, like decision-making, and user reactions, represents the most important difficulty in this filed (Scallen, et al., 1995).
- **User Goals** represent the structure levels to accomplish a task. Brusilovsky et al., define the user goals as the features that *always changes from session to session and often can change several times within a session* (Brusilovsky & Eklund 1988)
- **User Cognitive workload** represents the variables input class associated with the workload. This feature is very important because provides a direct link to user performance. Bi,s et al., shown as both high and low cognitive negative affect user task performance (Bi & Salvendy 1994). Different techniques are used to measure cognitive workload. Some techniques are based on a psychophysiological method such as the index of cognitive activity that represents an innovative technique to provides an objective measurement of workload (Marshall, 2002). Other techniques are based on subjective measures, such as NASA Task Load Index (TLX) (Hart & Staveland, 1988) (Hart 2006), or workload assessment technique (SWAT)

(Reid & Nygren 1988). These techniques rely on the user self-assessing the cognitive load through the use of questionnaires.

- **User situation awareness** defined by Endsley as " the perception of the elements in the environment within a volume of time, and the comprehension of their meaning, and the projection of their status in the near future" (Endsley & Strauch 1997). There are different techniques to calibrate situation awareness measures. For example, Endsley defines a method able to compare the user's declared perceptions with the real stimuli: Situation Awareness Global Assessment Technique (SAGAT) (Endsley 1987). Taylor designed a method to understand the workload through the use of a questionnaire: SART (Taylor 2017).
- **User knowledge** represents the most important feature of the user for existing adaptive systems. Brusilovsky shows how one-third of adaptation techniques are based on user knowledge (Brusilovsky 1996). Furthermore, adaptive systems have to recognize the changes in the user's knowledge and to update the user model. There are several ways to model the user (3.2. User Model).
- **Group of users** represents a standard group model using as a starting point for user interaction (Brusilovsky 1996). In detail, any interaction made by a new user with the system implies the new profile construction based on known features that belong to existing profiles of user groups (Rothrock, et al., 2002).
- **Situation variables** represent possible situations in which the process passes from a normal state to a disturbed state (Yen & Acay 2009). This means that the user's goals and resources are influenced by the situation. Three different situations under time pressure have been identified by Francisco-Revilla: emergency, consultation and educational (Francisco-Revilla & Shipman 2000). The adaptive system role should be to help the end user in managing awareness situations.
- **System variables,** like the situation variables, represents the same changes of the task. These variables are interdepend with the user and situation variables (Hettinger, et al., 1996).

Table 1: *Adaptation Variables*

Adaptation Variables	
User Performance	The user error percentage in performing a task
User Goals	The levels structures to accomplish a task
User Cognitive workload	The variables input class associated with the workload
User sitation awareness	The perception of the elements in the environment
User knowledge	The most important feature of the user for existing adaptive systems
Group of users	A standard group model using as a starting point for user interaction
Situation variables	Possible situations in which the process passes from a normal state to a disturbed state
System variables	Changes of the task.

3.4.1.2 Rules

The second step in the adaptation process is the definition of rules. The rules that guide the adaptivity vary in currently implemented systems. Moreover, in most adaptive systems, the adaptation process is guided by fixed case adaptation rules. Practical experience developing adaptive systems has shown that the majority of the adaptive systems use specific rules.

Specific rules are easy to apply and are reliable, but only apply to a narrow range of adaptation problems; abstract rules span a broad range of potential adaptations but are often hard and expensive to apply because they do not provide task- and domain-specific guidance. In those adaptive systems that do perform adaptation, specific rules are often used, requiring that the developer performs difficult analysis of the task and domain to determine which rules will be needed (Karagiannidis & Sampson, 2004).

For example, Arens et al., shown some typical rules: “If information is urgent, then choose a medium with low default detectability and a channel with no temporal variance” (Arens, et al., 1992). Petoud and Pigner (1990) for example, use a set of selection rules of the form: “a window is selected for each entity, a simple edit box is selected for each attribute of an entity, a table is selected for each repetitive aggregate of attributes, and a push button is designated for each function”. Benyon uses specific adaptation rules to exploit the data in the user model in order to effect the changes (Benyon & Murray, 1993). Baresi and Pasqale (2011) define three different type of rules: a triggering rule, that activates the evaluation of the trigger associated with the goal, a condition rule that evaluates the conditions linked to the goal and finally the activation rule that executes the adaptation actions when the previous rules provide positive feedback. In this context, a common problem is that

often these rules are usually hard-coded in the user interface and cannot be easily modified.

3.4.1.3 Goals

Finally, the last step of the first level is the definition of adaptation goals (***Table 2***). An adaptation goal defines the right actions sequence to achieve the final goal of the system. Each adaptation goal is associated with a system condition set which refers to rules, properties, variables, etc. The rules constitute the system trigger and indicate how and when the adaptive target have to be activated (Baresi & Pasqale 2011). Each adaptation goal is implemented through adaptation actions. Depending on the application requirements and the final user, the main adaptation goals vary substantially in the adaptive system. For instance, in the control system of multiple remotely piloted aircraft, to improve the operator performance and user productivity and to reducing error occurrence, the adaptive user interface has been used (Evans, et al., 2013) (Yen & Acay 2009).

Table 2: *Goals of Adaptation* (*Dieterich, et al., 1993)*

Adaptation Goals	
	Present what the user wants to see
	Speed up use
	Simplify use
	A UI that fits heterogeneous user groups
	A UI that considers increasing experience
	Easy, Efficient, effective use
	Make complex systems usable

Moreover, in the ubiquitous environment, the use of adaptive interface allows the final user to manage various kinds of devices such as automated appliances and sensing devices providing the user with convenient services (Namgoong, et al., 2006) (Song & Cho 2013). Furthermore, in the automotive environments, user interfaces have been used to efficiently attempt to provide content the driver needs and wants (Rogers, et al., 2000). More examples of this main goals can be found in the literature.

Dieterich et al., define a list of adaptation goals. In general, they identify as an overriding goal an easy, efficient and easy to use user interface (Dieterich, et al., 1993)

3.4.2 The second level

The second level consists in the definition of a set of inference mechanism dealt with in literature and used to draw conclusion.

3.4.2.1 Inference Mechanism

In the field of Adaptive Systems, the inference engine is a component of the system that applies logical rules to the knowledge base to deduce new information. The first inference engines were components of expert systems. The typical expert system consisted of a knowledge base and an inference engine. The knowledge base stored facts about the world. The inference engine applies logical rules to the knowledge base and deduced new knowledge. This process would iterate as each new fact in the knowledge base could trigger additional rules in the inference engine (Hayes-Roth et al., 1983). Inference can be regarded as a classifier algorithm. The most used algorithms are Decision tree (Janikow 1998), Artificial Neural Network (Yen & Acay 2009), Fuzzy Logic (Janikow 1998) and Bayesian Networks (Yen & Acay 2009).

Decision Tree. Decision tree learning is a method used in data mining and machine learning. The resulting knowledge, in the form of decision trees and inference procedures, has been praised for comprehensibility. This appeals to a wide range of users who are interested in domain understanding, classification capabilities, or the symbolic rules that may be extracted from the tree (Quinlan 1993) and subsequently used in a rule-based decision system. It is one of the predictive modelling approaches (Freund & Mason 1999; Tso & Yau 2007; Moore, et al., 1991; Darnell, et al., 2007). Decision trees assign symbolic decisions to new samples. This makes them inapplicable in cases where a numerical decision is needed, or when the numerical decision improves subsequent processing.

Artificial Neural Networks (ANNs). Artificial neural networks are non-linear mapping structures based on the function of the human brain. They are powerful tools for modelling, especially when the data relationship is unknown. RNAs can identify and learn related models between sets of input data and corresponding target values (Mitchell 1997). After training, RNAs can be used to predict the outcome of

new independent input data. RNAs mimic the learning process of the human brain and can process problems that involve non-linear and complex data even if the data are inaccurate and noisy. RNAs have become the object of much attention, largely due to their wide range of applicability and the ease with which they can treat complex problems (He & Xu 2010). The ANN has used for the prediction (Kalogirou & Bojic 2000), for modelling complex systems (Goh 1995; Yeh 1998) and for machines learning (Haykin 2009; Pharm, et al., 2017; Kim 2018).
The main limitations of neural networks refer to long learning periods and to the outputs provided that may not represent the perfect solution.

Fuzzy Logic. A fuzzy inference system employs fuzzy if-then rules to model the qualitative aspects of human knowledge and reasoning processes without employing precise quantitative analyses (Jang 1993). The fuzzy modelling has been used in different context such as control (Pedrycz 1993; Ben-Ar 6 Mondada; Mostafa, et al., 2018), prediction (Sugeno 1985; Eyoh, et al., 2018; Caesarendra, et al., 2018)] and inference (Schneider, et al., 1988; Jang 1993; Chin & Lin 2018; Karaboga & Kaya 2018). However, there is a lack of standard methods for transforming human knowledge or experience into the rule base and database of a fuzzy inference system and there is a need for effective methods to minimize the output error measure and maximize performance.

Bayesian Belief Networks. Bayesian networks (BBNs) or probabilistic expert system, are a technology for automating human-life reasoning under uncertainty in a specific context (Jensen 1996; Nielsen & Jensen 2009). In particular are a type of Probabilistic Graphical Model that can be used to build models from data and/or expert opinion. They are built from probability distributions and can be used for a wide range of tasks including prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction and decision making under uncertainty. The BBNs are used in many different applications such as medical diagnosis (Duda, et al., 2018; Lakho et al., 2017; Nikovski 2000; Kononenko 1993), software development, industrial process control (Weidl, et al., 2005; Friedman 2000; Alaeddini & Dogan 2011; Heger & Aradhye 2002), aircraft control (Li, et al., 2017; Nannapaneni, et al., 2018), smart environments (Song & Cho 2013; Anitha & Priya 2018) etc. The main advantage of probabilistic reasoning lies in the possibility of reaching rational descriptions even when there is not enough deterministic information about the functioning of the system.

3.4.3 The third level

Once defined the user target and the context in which the interface will be developed, the adaptation variables, the rules and the objectives to be satisfied, the suitable inference mechanism chosen to develop the adaptation, the last one step consists in the user interface design process and in detail how it is carried out and what steps are taken to complete it (Redmond-Pyle & Moore, 1995).

The main thing to do when designing an application or a system with a specific product is to observe and understand the product itself, what it is and what its logic is, why people would need it (*observe*, ***Fig. 41***). Subsequently, it is essential to make explicit user-centered requirements and an indication of appropriate priorities. In this case is useful create use cases or list of actions or event steps typically defining the interactions between an actor and a system to achieve a goal, according to a specific scenario (*ideate*, ***Fig. 41)***. This will help in creating test information for the end users to test later after the design has started (Hix & Hartson, 1993).

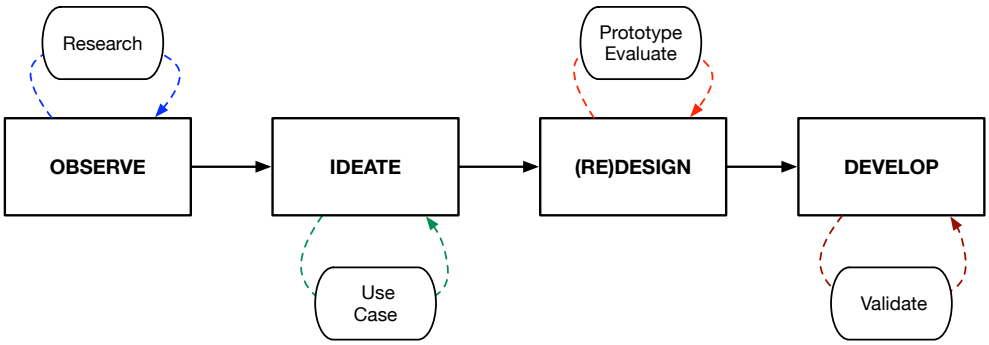


Fig. 41: The User Interface Design Process

Another important step in the interface design process is the (re)design phase, where design solutions are simulated using paper or computer-based mockups (prototype). A prototype is a semi-functional layout that can give a high-fidelity preview of the actual app or website user interface (front-end) functionality. Generally, it gives customers and/or end-users an ability to click around the elements of the interface and simulate the way the app will actually work; while it gives designers the opportunity to bring their ideas to life, test the practicability of the current design, investigate how a sample of users think and feel about a product, clarify some requirements and compare the alternative. Finally, this phase helps to test the ideas and make some changes at early stages of the project ((*re)design*, ***Fig. 41)***.

The last phase is the interface development where the UI is validated. The main purpose of the evaluation stage is to assess the quality of a design, not in the abstract, but how it fits the users, meets their needs and displays all the product's features. In this phase, the best solution for the system in light of the requirements is selected. Furthermore, the results and recommendation for refining the design at this stage are reported. This evaluation process is iterated at each stage until design/usability goals or requirements are met (*develop*, ***Fig. 41***).

Chapter 4.

Architecture

4.1. Introduction

In order to study and develop an adaptive and adaptable user interface, a new design method able to support the modelling of information related to the user and to the context of the interaction has been presented [*Chapter 3, Adaptive Interface Methodology]*.
On the basis of the previous studies results and with the aim to develop an adaptive user interface, a new general and replicable architecture has been developed. This structure is based on three modules, constantly communicating with each other:

1. The Database (DBMS)
2. The Core
3. The User Interface (UI)

The architecture is showed in ***Fig. 42***. The user-system interaction takes place by the appropriate device (tablet, smartphone, smart Tv, etc.), allowing the system to implement a dual recognition:

- ***Action Recognition***; this module is responsible for the user actions recognition on the interface. In particular, the user action recognition occurs through a constant communication between the *User Action/Adaptive Engine* core module and the *User Use Profile* database module. Each user action is stored at the *User Use Profile* database module and sent to the *Bayesian Network*, located in the *Adaptive Engine* module. This allows a continuous training and population of the network. Furthermore, the *Adaptive Engine,* consisting of the *Bayesian Network* and the *Decision-Making Algorithm*, receives information from the *Context Data* database module. Based on the network knowledge gained, the *Decision-making*

Algorithm manages the adaptation. In particular, there are two adaptation types:

- *Static Adaptive Behavior,* that consists of an initial adaptation limited to default user recognition or the last setting used.
- *Dynamic Adaptive Behavior*, that consists of a run-time adaptation based on the system usage.

In order to track all the system adaptation, the static and adaptive behaviors, in the Log Adaptation Action database module, are saved. Based on the network-triggered adaptation, graphical and content interface information will be subject to change.

- ***User Recognition***; this module recognizes the user type and profile. The user information, such as demographic or preferences, are saved in the *User Profile* core module. Depending on the user profile, the *Adaptable Engine* changes the interface setting: layout, font and zoom.

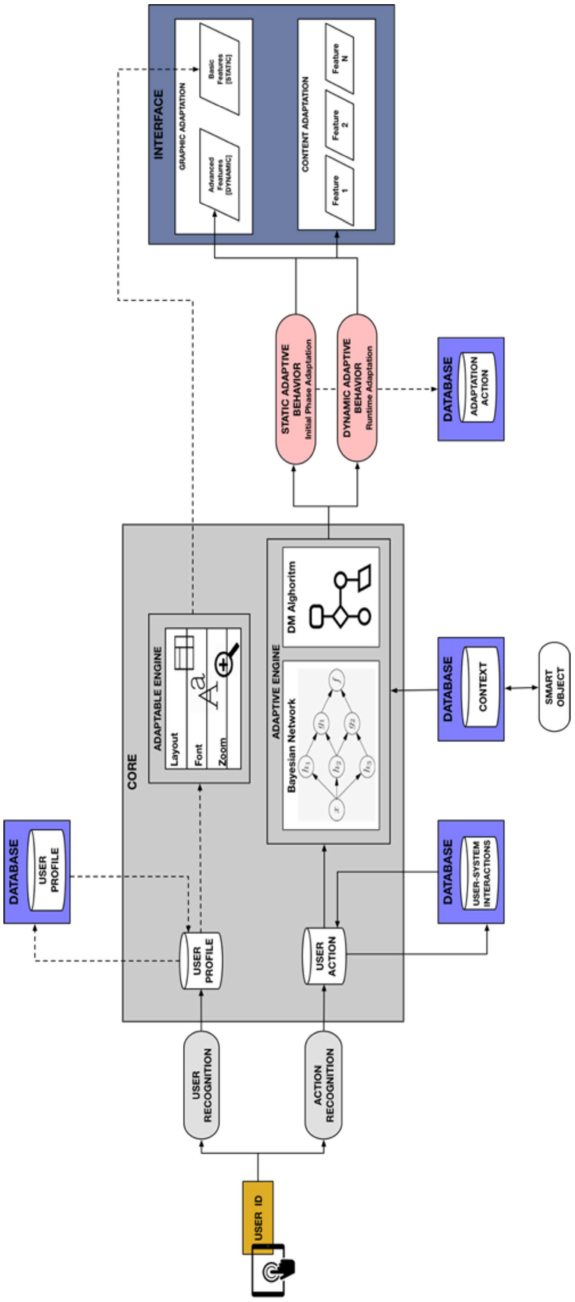


Fig. 42: The System Adaptive Architecture

4.2. Database

The Database Management System (DBMS) is a software designed to achieve a large set of structured data inputs and processes the amount of data requested by numerous users, also called "database manager or engine", and hosted on a special hardware architecture. This module aims to collect all the data deriving from the different system inputs. In details, the information is structured in four semantic areas:

- *User Features Profile,* which contains the personal information described in the User Model *[3.2. User Model]* such as aptitude, personal skills, demographic information, etc. Such information can be acquired by the system by making inferences about users, by carrying out some from test, or from assigning users to generic user categories usually called 'stereotypes', by making explicit acquisition through asking a relevant question, or by making previously stored information (***Table 3).***

Table 3: *User Features Profile*

User Features Profile	**Demographic Attributes**	Socio-Economic
		Personal
	Health Attributes	Physical Factor
		Physical Abilities&Disabilities
	Behavioural Attributes	User Preferences
		User Interest
		Context Information

- *User Use Profile*, which contains all the contextual information such as the history of interactions with the environment or with the service. In this sub-module, all the user interactions are stored, and his/her overall interaction history can be reconstructed (***Table 4).***

Table 4: *User Use Profile*

User Use Profile	**Historical Environment Interactions**	**Devices**	Start/Pause Command
			Setting Temperature
			Setting Programs
			Other
	Historical Service Interactions.	**Instructions/Information**	Help
			Interactive Guide
			Other

- *Log Adaptation Actions*, which contains the collection of all the adaptations provided by the Adaptive Engine during the application usage. Whenever the system performs an adaptation action, the information is stored. This information is needed to the system control adaptation and to improve the user's usability during the process (
- ***Table** 5*).

Table 5: *Log Adaptation Actions*

Log Adaptation Actions	**Default Setting Device**	Adaptation Event of Program Default
		Program adaptations number counter
		Adaptation Event of Temperature Default
		Temperature adaptations number counter
		Adaptation Event of Duration Default
		Duration adaptations number counter
	Instructions/Information detail	Event Adaptation detail interactive guide
		Instructions adaptations number counter

- Context Data, which contains all relevant data for the user context definition. It records the information from the environmental sensor, the device and the time information [*3.3. Environment Model] (**Table 6**).*

Table 6: *Context Data*

Context Data	**Devices**	On-Off
		Program Status (Start/pause)
		Program Set
		Temperature Set
		Temperature Set
		Others
	Sensors	Temperature
		Humidity
		Air Quality
		Lighting
		Others
	Time Info	Time
		Data

4.3. Core

The brain of the application is represented by the "core" layer that provides the application logic management *[3.4. Interaction Model]*. The Core is composed of two engine mechanisms (Adaptable and Adaptive engine) and a changes monitoring system.

4.3.1 Adaptable Engine

Adaptable Engine shall make the system more suitable to the user profile. Adaptability is based on features and preferences that are known as a first interaction, and they are assumed to remain static during a single session of interaction. This engine takes as input the collected information in the User Profile Features to adapt the graphical interface such as text, size and type of font.

4.3.2 Adaptive Engine

The Adaptive Engine represents the overall adaptive system pivot: it is composed of an adaptive mechanism and a Decision-making algorithm. The Adaptive Engine makes the adaptive system according to the user profile. In particular, adaptivity consists of change mechanisms which include all dynamic features, such as preferences based on the history of user's interaction, information contents, icons, layout, etc. On the basis of the user-system interaction, the adaptive engine monitors all the user interaction with the system and provides the best solutions in terms of

context or graphic elements. In particular, the chosen adaptive mechanism is based on the Bayesian Belief Network (BBN) *[3.4.2 The second level]*. As the ***Fig. 42*** shows the user actions are recognized through the *Action Recognition*, locally stored (*User Action* database module) and sent to the *Bayesian Network (Adaptive Engine*). The Adaptive Engine outputs are two:

- *Static Adaptive Behavior*, this output sets the type of interface and initial home appliance defaults. It takes care of the initial interface setup configuration and the initial default. It is handled by calculating the probabilities associated with finding the User Use Profile, and will set the defaults as the last adaptation made (the previous state, ended);

- *Dynamic Adaptive Behavior,* this output refers to user actions (which are saved and given to the network at runtime). It also manages household appliance defaults and updates the interface at runtime.

4.3.2.1 Inference Mechanism: Bayesian Networks

Nowadays, Bayesian networks are one of the most comprehensive tools and more consistent for the acquisition, representation and exploitation of knowledge in conditions of uncertainty. These networks are the result of the Bayesian Statistical Methodology convergence, which allows the exhange of information contained in experimental observations to the probability law. In particular, a BBN is a probabilistic graphical model that represents a set of stochastic variables with their conditional dependencies through the use of a directed graph; each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding random variable (Nielsen & Jensen 2009 (Jensen & Nielsen 2011).

In order to detect user actions during interaction with the smart system interface, a "using" profile model is constructed, which implement BBN to estimates the probabilities of the user expected behavior. ***Fig. 43*** shows an example of a general BN.

In order to manage BBNs, we propose to use the simulation software Netica, produced by Norsys Software Corporation (Application, 2004). This software actually is a very useful tool for working with BBNs and influence diagrams. It easily allows drawing of Bayesian networks or to create them by using powerful Netica APIs toolkit, which makes it interoperable with the major programming languages (i.e., Java, C, Visual Basic).

Moreover, it provides tools to:

- Detect optimal decisions for sequential decision problems and solve influence diagrams efficiently by using clique trees;
- Learn probabilistic relations from data, even with missing data;
- Allow the entry of probabilistic relations by the equation, with an extensive built-in library of probabilistic functions and many other mathematical functions.

The user model is then designed and implemented through the following steps:

1. User activity decomposition. A proper user activity decomposition analysis is carried out to identify: (a) actions performed by the user on the system, which is relevant to achieve specific goals; (b) system actions to detect user characteristics, which may affect user-system interaction (e.g., user's emotional, physiological and psychological states).
2. Context analysis. A context analysis is performed to understand how environment attributes can affect user-system interaction (e.g., year period, time of day, lighting condition, temperature, etc.).
3. Building of network structure in Netica. Definition of nodes representing context, user's actions, system actions and user's goals and define relations (i.e. flows) between each other.
4. Network modelling: for each node, a collection of properties and states are defined.
5. Compilation of the Network. Compilation process allows Netica to build the junction tree relative to the inferences.
6. Network Learning and Simulation. It consists of learning probabilistic relations between random variables. To this purpose, Netica Software offered two different possibilities: manually enter the probabilistic relations or to import data to learn from the existent case.

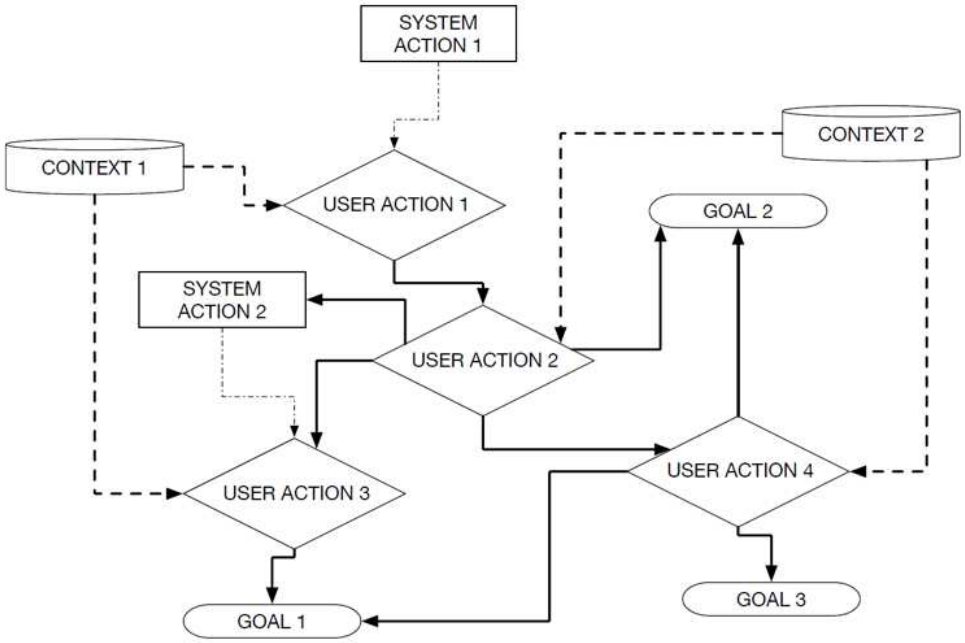


Fig. 43: Example of a general Bayesian Network (BN)

Netica automatically calculates the statistical probabilities of all events associated to each BN nodes. To allow the network to provide valuable support for the identification of proper system adaptation action, a decision-making process has been defined, as described in the next sub-section.
To handle the multi-user situation at the same time a user profile is created for each user and BBN is populated by his/her interaction events.

4.3.2.2 Decision Making Algorithm (DMA)

In order to detect the appropriate system adaptive behavior, a decision-making algorithm has been defined (DMA).
To be effective, DMA must support the identification of proper system adaptation actions, according to the collected user and context data. To this end, this algorithm determines if exist a predominant probability (PP) for a specific event (e.g., user's action or goal) in input node distribution and trigger system adaptive action only in this case.

A probability is defined predominant if its difference with the second greater probability (P_{2th}) is bigger than a chosen percentage (**thr_perc)** of the distribution mean.
Let be a target distribution **D**, the first step of the algorithm consists in calculating the higher probability, P_{max}.
Once the maximum probability has been calculated, the algorithm verifies the uniqueness of the maximum. If and only if the maximum is unique, the DMA proceeds with the calculation of the second greater probability, P_{2th}, otherwise there will be no adaptation.
Then, it is necessary to define the gap between the maximum probability and the second highest probability:

$$\boldsymbol{P_{max} - P_{2th} = d}$$

Once defined ***d***, the average of the target distribution probability is calculated: ***av***.
At this point, the **thr_perc** threshold is defined to allow the algorithm to provide a Boolean condition in order to inform the system about the right adaptation action. The threshold consists of the parameter to be set according to the level of restriction that is to be attributed to the algorithm.
Once the threshold has been chosen, the ***thr*** is calculated as follow:

$$\boldsymbol{thr = \frac{av}{100 * thr_perc}}$$

In particular, the behavior is suggested if and only if its probability exceeds the average of the other behaviors of a given percentage, i.e. ***thr_perc***. The adaptation event is triggered if ***d > thr*** is verified. Otherwise, no adaptation is required. The ***thr_perc*** threshold is chosen by the company or by the designer. The flowchart, shown in ***Fig. 44***, reports the described algorithm.

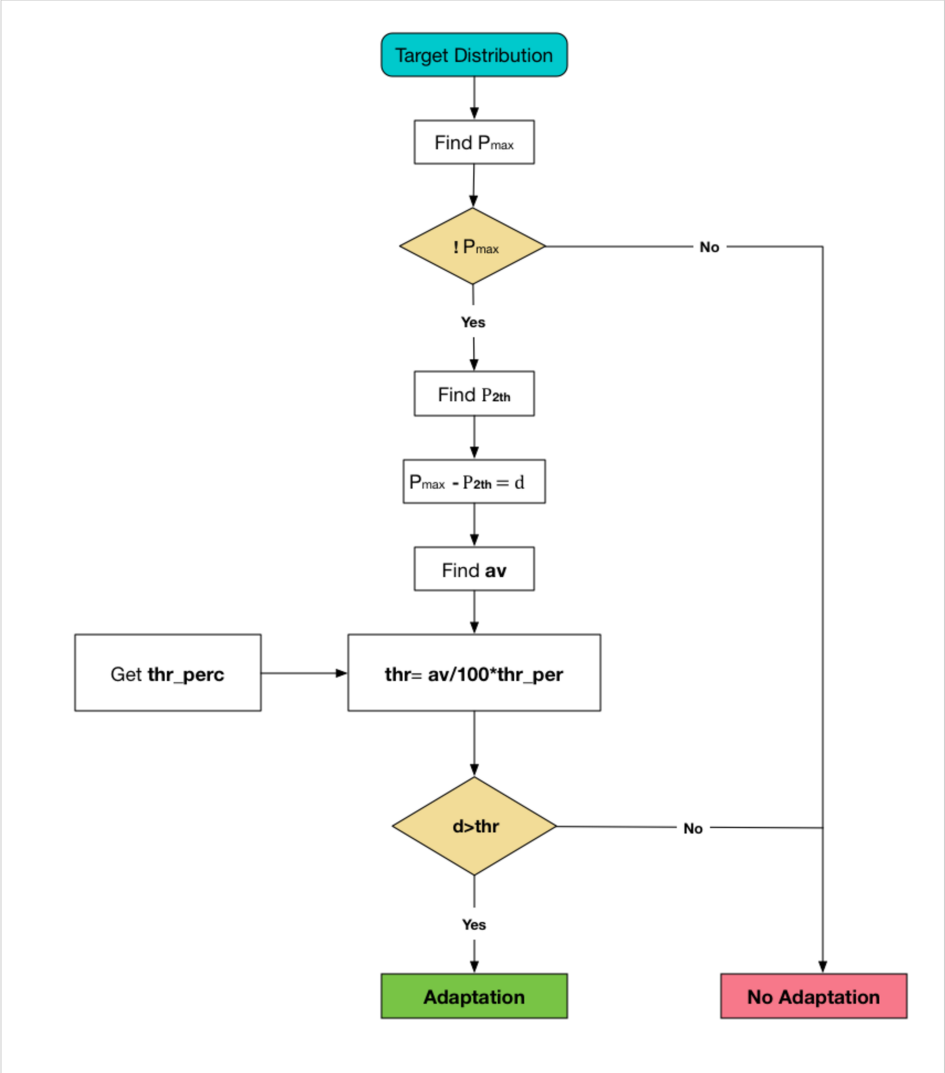


Fig. 44. The decision-making algorithm

4.3.2.3 Simulation BBN and DM

To test the effectiveness of BBN network and of the DM algorithm a BBN was built assuming the use of the system in a home environment characterized by three appliances: oven, dishwasher and refrigerator. The structure, shown in, consists of 14 nodes, defined as:

1. Year_Period: to specify the four year periods (Fall, Winter, Spring and Summer).

2. Week: the seven days of the week (Monday Tuesday Thursday Friday Saturday and Sunday).
3. Day_Range: schedules the day in four time periods (range 1 = 06:01 to 12:00, Range2 = 12:01 to 18:00, Range3 = 18:01 to 12:00, Range4 = 0:01 to 6:00).
4. Household_Devices: characterized by three appliances: oven, dishwasher and refrigerator
5. O_CycleTarget: specifies the oven program setting.
6. O_CyclePeriod: is the oven set cooking times.
7. O_TemperatureTarget: Consists of the oven temperature setting.
8. O_StartTime: Involves the oven time setting.
9. TemperatureFreezer: the freezer temperature setting.
10. TemperatureFridge: the fridge temperature setting.
11. D_CycleTarget: the dishwasher program setting.
12. D_CycleSetting: the options set washing in the dishwasher
13. D_CycleSoap: the dishwasher options related to the presence / absence of detergent
14. D_StartTime: the dishwasher time setting.

The network links show the direct dependence between the nodes; for example, the network root node "Household_Devices" is directly influenced by three context nodes: year, a day of week and time of day.
Depending on the appliances historical use, the node stochastic probability will be updated and return the more likely use devices in output. In turn this node will affect all the other nodes, which are related to oven, dishwasher and fridge data setting. Finally, this node will return the greater output use predictability. Once the network is built, it is necessary to set probabilistic relations between random variables.
To this purpose, existent case data from the external database are automatically imported through the Learning Case function. Once the network was built, a simulation to evaluate the actual behavior has been performed.
To create an external database necessary to implement and tests adaptation functions, several use case scenarios have been defined for each target user profile. An example of the scenarios is reported in the ***Fig. 45***. An extract of the respective external database is reported in ***Fig. 47.*** The database is designed to collect a set of structured data. In detail, the columns represent the network nodes, the lines indicate the node status and are updated at each simulation with the system.

EXAMPLE OF SCENARIOS

John used the dishwasher once a day, usually in the evening, with the breakfast, lunch and dinner dishes. Frequently uses the Daily program serving detergent tablets for convenience. The weekend habits change: Sunday john sets the Intensive program and start washing immediately after lunch, using detergent tablets. Furthermore, John uses the oven 4 times a week; Mondays and Fridays for cooking the seafood he always sets a delayed start and uses the Ventilated program. On Saturday evening he uses the oven to cook pizza with Thermo-ventilated program. Every Sunday evening, he also activates the oven cleaning

Fig. 45: Example of use case scenarios simulate to create learning data

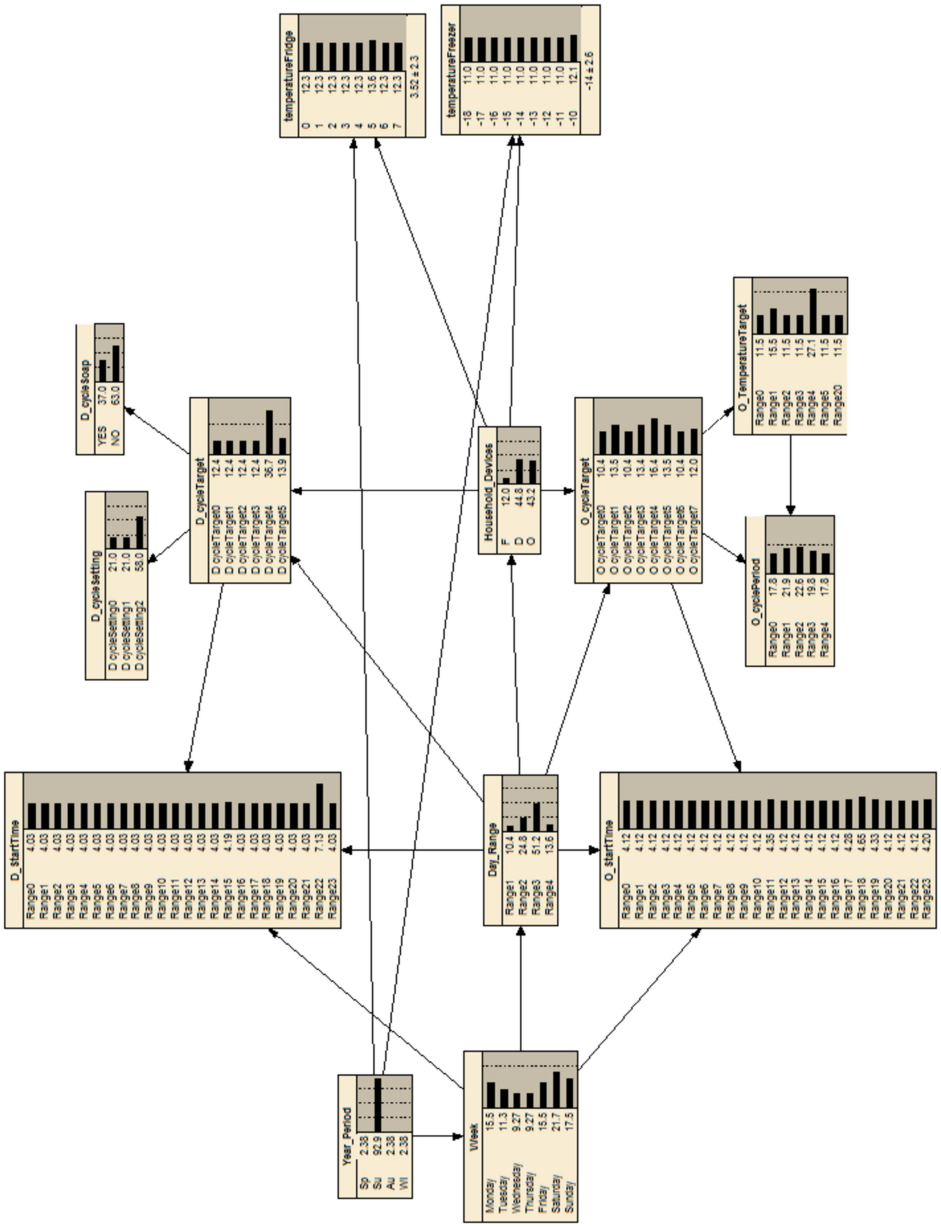


Fig. 46: The BBN Example

Day	Year_Period	Week	DayRange	Household_Devices	temperatureFridge	temperatureFreezer	O_cycleTarget	O_StartTime	O_TemperatureTarget	O_cyclePeriod	D_cycleTarget	D_StartTime	D_cycleSetting	D_cycleSoap
27/07/15	Su	Monday	Range3	O			O_cycleTarget2	Range20	Range1	Range0				
27/07/15	Su	Monday	Range2	F	5	-10								
28/07/15	Su	Tuesday	Range3	D							D_cycleTarget0	Range22	D_cycleSetting2	NO
29/07/15	Su	Wednesday	Range3	D							D_cycleTarget1	Range22	D_cycleSetting2	NO
30/07/15	Su	Thursday	Range3	D							D_cycleTarget2	Range22	D_cycleSetting2	NO
31/07/15	Su	Friday	Range3	O			O_cycleTarget4	Range18	Range4	Range2				
31/07/15	Su	Friday	Range3	D							D_cycleTarget3	Range22	D_cycleSetting2	NO
01/08/15	Su	Saturday	Range2	D							D_cycleTarget4	Range14	D_cycleSetting2	YES
02/08/15	Su	Sunday	Range4	D							D_cycleTarget5	Range22	D_cycleSetting2	YES
02/08/15	Su	Sunday	Range3	O			O_cycleTarget5	Range18	Range4	Range3				
03/08/15	Su	Monday	Range3	O			O_cycleTarget2	Range20	Range1	Range0				
03/08/15	Su	Monday	Range3	D							D_cycleTarget3	Range22	D_cycleSetting2	YES
04/08/15	Su	Tuesday	Range3	D							D_cycleTarget3	Range22	D_cycleSetting2	YES
05/08/15	Su	Wednesday	Range3	D							D_cycleTarget3	Range22	D_cycleSetting2	YES
06/08/15	Su	Thursday	Range3	D							D_cycleTarget3	Range22	D_cycleSetting2	YES
07/08/15	Su	Friday	Range3	O			O_cycleTarget3	Range20	Range4	Range3				
07/08/15	Su	Friday	Range3	D							D_cycleTarget3	Range22	D_cycleSetting2	YES
08/08/15	Su	Saturday	Range2	D							D_cycleTarget4	Range14	D_cycleSetting2	YES
09/08/15	Su	Sunday	Range3	O			O_cycleTarget5	Range18	Range4	Range3				
09/08/15	Su	Sunday	Range3	D							D_cycleTarget5	Range14	D_cycleSetting2	YES
10/08/15	Su	Monday	Range3	O			O_cycleTarget2	Range20	Range1	Range0				
12/08/15	Su	Tuesday	Range3	D							D_cycleTarget3	Range22	D_cycleSetting2	YES
13/08/15	Su	Wednesday	Range3	D							D_cycleTarget3	Range22	D_cycleSetting2	YES
14/08/15	Su	Thursday	Range3	D							D_cycleTarget3	Range22	D_cycleSetting2	YES
15/08/15	Su	Friday	Range3	O			O_cycleTarget4	Range18	Range4	Range2				
16/08/15	Su	Friday	Range3	D							D_cycleTarget3	Range22	D_cycleSetting2	YES
17/08/15	Su	Saturday	Range2	D							D_cycleTarget4	Range14	D_cycleSetting2	YES
18/08/15	Su	Sunday	Range4	D							D_cycleTarget5	Range22	D_cycleSetting2	YES

Fig. 47: External database related to Scenario

As previously described, the algorithm configuration needs one precise values of the threshold parameter (***thr_perc***).

To reach this goal, a testing procedure is developed: the data management and processing is carried out through the Numerical Computation Software Scilab 5.2.2, using the script coding approach.

The operational procedure is conducted as follows:

- The Beliefs of a Test Bayes Network achieved with Netica Software are exported in an XML file and loaded on Scilab Environment;
- Then, an empirical graphical evaluation allows selecting the distributions suitable to trigger an adaptation action.
- Then the value of the threshold parameter is chosen to define the simulation domain.

The assessment allows to demonstrate the goodness of this approach as a tool for achieving a decision-making algorithm based on the correlation between the probability value of the different level of a stochastic distribution.

In the present case the behaviors of the algorithm were simulated using two different threshold values:

***thr_perc* = 20**

***thr_perc* = 40**

The figure shows how increasing the threshold value increases accordingly the restriction level that is to be attributed to the algorithm.

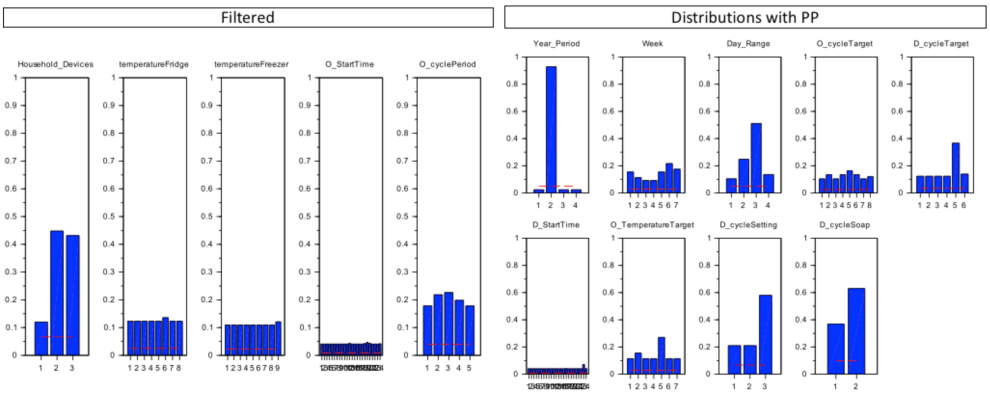


*Fig. 48. The DMA algorithm tested with a **thr_perc = 20**. On the left the plot of the filtered nodes is shown (not subject to adaptation). On the right the nodes that satisfy the condition and are adapted are shown.*

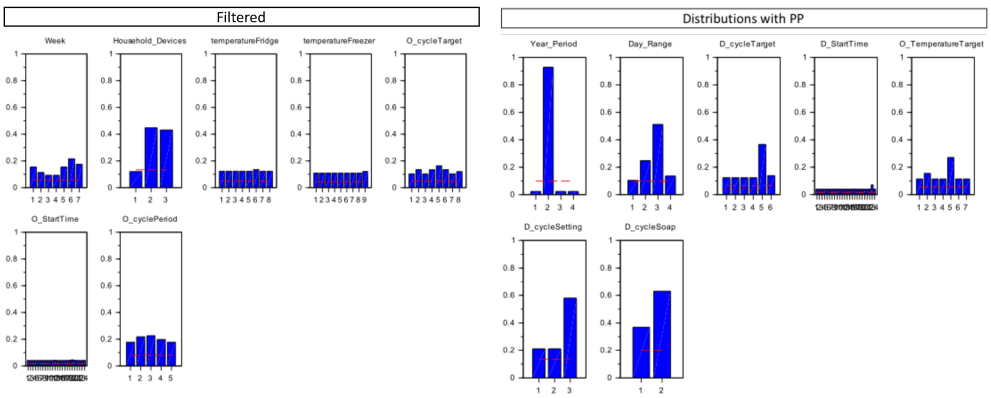


*Fig. 49: The DMA algorithm tested with a **thr_perc = 40.** On the left the plot of the filtered nodes is shown (not subject to adaptation). On the right the nodes that satisfy the condition and are adapted are shown.*

4.4. Interface

The User Interface is the module between the system and the end user and it enables the control of the environment *[3.4.3 The third level]*. The UI is managed by the Application Core that controls the adaptivity of the interface. The adaptation mechanism can operate both (a) on the graphics features and (b) on the contents of the interface. The graphic features (e.g. colour, text and buttons dimension, image size, etc.,) are uniquely related to a disorder (i.e. colour blindness, visual disturbances, motor problem) and they are being designed on the loss of body functionalities with the aid of existing guidelines.
The contents concern the quantity and type of information that the interface gives to the user and they are designed on the abilities of users.

4.4.1 Concept Interface

In order to manage the information needed to design the interface concept a Unified Scheme Functional (SFU) (Ceccacci et al., 2016) was chosen. It allows to represent in a single scheme, and in a simple and intuitive design, the entire of design space. Also, the SFU is used to represent the full set of design variations depending on the user's attributes and parameters that regulate adaptation mechanism.
To this end, the SFU scheme is represented by five important points:

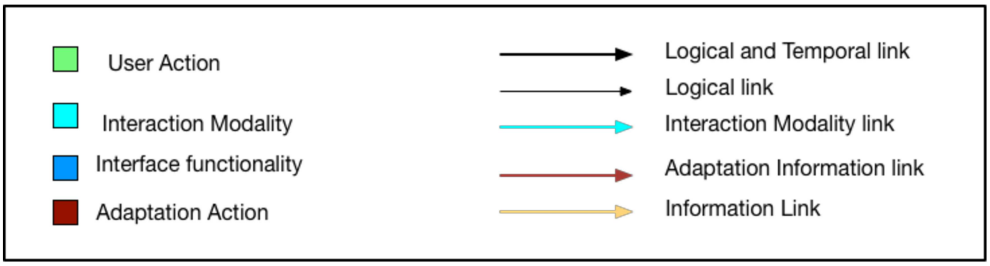
1. the user actions (user task),
2. the system actions performed by the system itself in order to provide the appropriate information to the user,
3. the interface variants (physical alternatives) and services (Interface functions) that are associated with each user profile,
4. the adaptation and data management.

The dynamic interface is represented by the following subsequent:

- Logical and temporal follow, to connect the user’s action;
- Adaptation flow, to manage the adaptation mechanism;
- Data flow, to regulate data management.

The SFU's construction activity can parallel the design and prototyping of the interface both in the case of an ex-novo design and in the case of the re-design of an already existing interface. In this way, the SFU scheme can provide a valid support to evaluate the problems and fix the adopted solutions. In addition, the SFU defines

all the design specifications of the interface and can also be used to manage complexity during development, so as to ensure compliance with the specifications of all project alternatives.
Below is a general outline of the SFU and some examples of solutions regarding the interaction adaptation and information of the programming function based on the information related to the User Feature Profile and the User Use Profile.



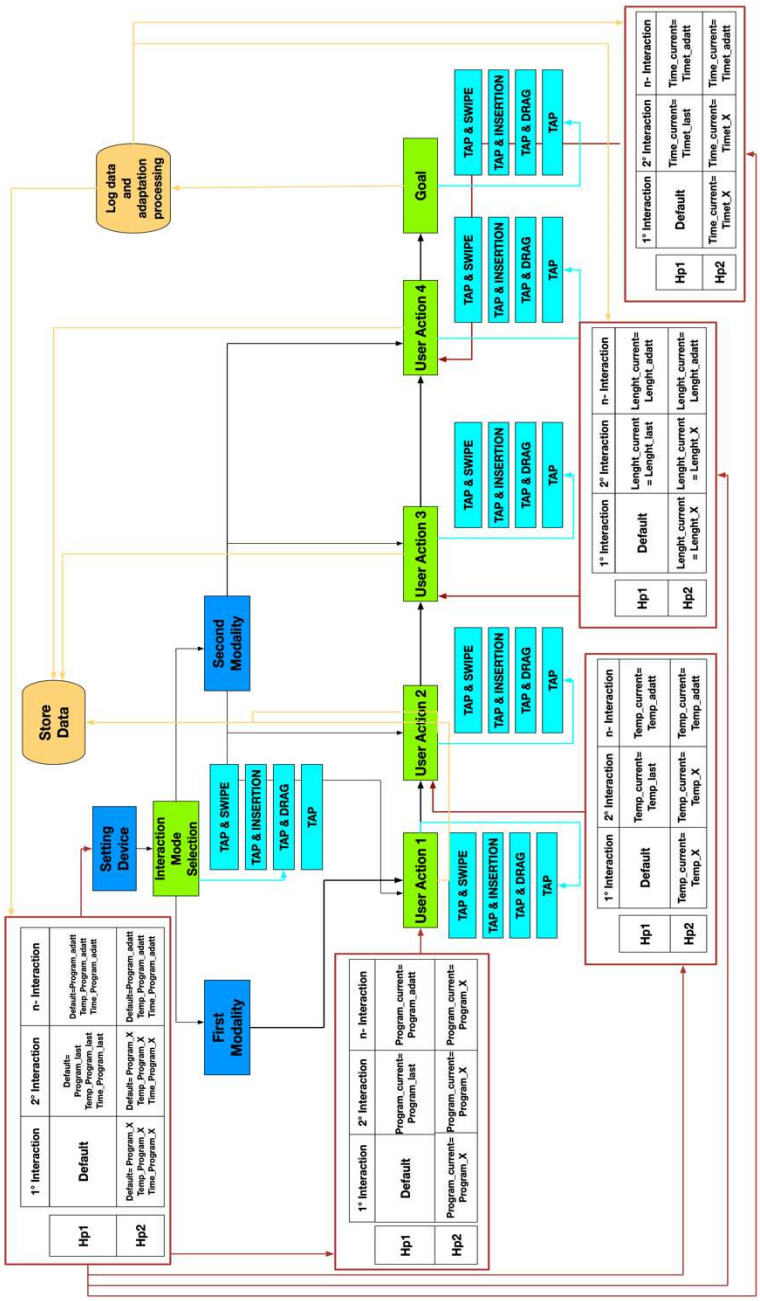


Fig. 50. The Unified Scheme Functional

Chapter 5.

The First Case Study: Prototyping Adaptive System in Smart Environment using Virtual Reality

5.1. Introduction

In recent decades, people's daily lives have been transformed by the faster developing of information and communication technologies (ICT). In particular, the spreading of smart products and the possibilities to connect IoT devices together have changed the living environments. Cook and Das defined these spaces as Smart Environments: “a small world where different kinds of smart device are continuously working to make inhabitants' lives more comfortable” (Cook & Das, 2004). Smart environment is a key challenge for current ICT research: it is one of the solutions that can improve people's quality of life and enable users with and without impairment to live independently. Furthermore, smart environments have the potential to support the independent lives of people, since they can improve comfort and energy efficiency, provide a continuous monitoring, to support them in performing daily life activities, provide a safe environment and help individual rehabilitation (Van den Broek, et al., 2010). They provide additional supportive functions by collecting environmental data and adapt improving the user experience in that environment.

Over the years, the scientific research has proposed several examples that help and improve the capabilities of its occupants, particularly in performing domestic tasks. However, most of these case studies often are focused on provided services and do not take full account of end-user preferences and applications. Experimentations are driven by technology improvement (technology push) and not from a careful user's needs analysis. Several solutions have been proposed e.g. to support users in performing daily life activities (Huch, et al., 2012), in preparing meals (Gullà, et al., 2016), controlling household appliances (Chaminda, et al., 2012; Sponselee, 2013), but they obtained rather low experienced benefits. The main reason was the lack of

adaptation, as the system did not adapt to the real needs of users with different profiles.
This first case study aims to develop a new adaptive system able to support the user in performing cooking activities and to help old or disabled people in carrying out domestic tasks that might otherwise be difficult or impossible. The system has been designed for normal people as well as users with several kinds of impairment (i.e. visual, motor, cognitive) in according to Design for All approach. It allows to manage all devices and appliances improving the usability for various utilization profiles/contexts and for different user requirements/skills. The system supports the user in the daily tasks execution by recognizing his/her preferences and actions and adapting the system feedback consequently.
To this end, the system supports the following functional areas:

- Meal preparation support: meal preparation will need information from refrigerator's food, user's profile and recipes.
- User interaction-household appliance support: the system provides the ability to access the appliance control enabling the latter to set up, launch and monitor a given program.

In order to support the greater number of end-user, two different UI modalities were designed: Normal for users without cognitive impairments and Wizard for users who have not familiarity with technology and/or have some cognitive dysfunction. With the aim to develop an easy (self-explanatory), efficient and pleasant (usable) adaptive smart system, the final users have been involved in the whole design and development process, according to ISO 9241-210:2010 (DIS, I., 2009). Finally, the system was validated through a Virtual Reality (VR) system allowing the user interaction evaluation and helping the usability improvement (Jiménez-Mixco, et al., 2009; Hall, et al., 2012).

The present research study was partially funded by the project “D4All: Design for all” (CTN01_00128_297089), by the Italian Minister of University and Research, under the National Cluster Ambient Assisting Living Technologies (TAV).

5.2. Analysis of Context of Use and Definition of Design Requirements

5.2.1 Definition of Personas

The main goal of this research work is the development of an adaptive system that enriches the user interaction with a daily life environment, the kitchen. In order to support the design phase, a reliable and realistic representation of target user groups for reference has been created ***[3.2. User Model].*** It is based on an ethnographic analysis. It allowed giving a clear picture of the users in terms of needs and expectations, from their backgrounds, limits, goals, and values to how they are likely to interact with the system. In particular, the personas (Cooper,1999; Gulliksen et al., 2003) related to the following three clusters of users have been defined: (U1) subjects with cognitive impairment, (U2) subjects with dexterity problems and (U3) visually impaired. The Personas method is a plain and effective tool to gather the strengths and objectives of the user profile for designer and developer. For instance, the identified profiles, are reported in the figure below:

	***BACKGROUND***	John is a former metalworker, now retired for 15 years. As a widower for three years now, he has been in charge of household affairs and homework alone. Very active in the social life of the local community, after his wife's death he suffers from a depressive state that partially compromises his cognitive state. For this reason, despite good physical health, it is unable to maintain a good level of control and environmental organization and is in a state of social isolation. The child lives far from his father's village and is unable to provide effective care for his or her needs.
Name: John	**Needs/Limits**	• Initial cognitive deterioration that prevents the use of common household appliances and the implementation of appropriate behaviours • Poor knowledge and attitude to technology • Lack of a formal caregiver at home
Age:72	**Behaviours**	A feeling of inadequacy that leads to a further state of environmental abandonment and neglect of one's physical condition (i. e. it always consumes the same cold meals because it does not know how to use household appliances)
Profession: Pensioner	**Goals**	• Increase environmental control capacity • Living independently • Timely monitoring of a possible worsening of cognitive status

	***BACKGROUND***	Teresa has always been a housewife. She lives with her husband Carlo, whom she takes care of as a result of her health problems, which have caused her little mobility. Lately, Teresa complains to her children of need for help in home activities, due to severe hand pain, caused by arthritis, which often does not allow her to carry out the necessary activities. Teresa has always been a housewife and, while knowing how to use common household appliances in the kitchen, she is unfamiliar with the latest technology, despite her children and her grandchildren always urged to rely on some help. Now, you also feel the need for a help, preferably technological and non-invasive, that allows you to personally carry out house management activities more quickly..
Name: Teresa	**Needs/Limits**	• Difficulty in manipulating small objects or manual precision activities, as a result of joint pain and stiffness • Poor knowledge and use of technology • Middle-low level of schooling • Sensation of loss of control over domestic activities with which Teresa has always been identified
Age:75	**Behaviours**	Attempts to do things alone often cause pain, fatigue and frustration
Profession: Pensioner	**Goals**	• Go back home independently, getting support as you interact with physical objects in your home.

	***BACKGROUND***	Sergio is visually impaired since birth. Because of this disorder, he can not be completely independent in managing his home, which mainly depresses him and causes his depressed mood. Thanks to IT, he gets support for his job as an employee, enabling him to develop a deep interest and good knowledge of technology and its potential. Despite this, it has not yet found the home appliance suitable for your needs. Indeed, in most cases, Sergio needs to enlarge what he observes in order to be able to perceive it visually and have difficulty in perceiving details of things and finding useful objects. In addition, vision worsens sharply in the evening and in dark environments. From the point of view of quality, it recently complains of a deterioration in the ability to distinguish between colours and image distortions. Cheats and personalization's in carrying out actions and practical activities are an indispensable resource. Searching and finding objects in the middle of many others and in ever different positions is for a more visually impaired person.
Name: Sergio	**Needs/Limits**	• Inability to live independently • Unable to see well if not under certain conditions
Age:46	**Behaviours**	• Bring objects closer to the face for a better view • Avoid social situations • Uses technologies for very specific and sector-specific activities (work)
Profession: Clerk	**Goals**	• Independence in domestic activities that require visual acuity for small details and sharpness; • Support for maintaining optimal environmental conditions (i. e. lighting, ability to find small and favourite objects) • Need for IT support to manage home control

Fig. 51. From above Personas related to the subjects with cognitive impairment (U1), with dexterity problems (U2) and visually impaired (U3)

5.2.2 Definition of System Functionalities

Personas method helped to uncover universal features and focus decisions surrounding system components by adding a layer of real-world consideration.
Subsequently, the environment model has been defined: the smart kitchen. This is principally composed of an oven, a fridge and a dishwasher.
For each household appliances the system functions have been defined. An extract of the system functionalities is given in ***Table 7*.** They refer to a kitchen environment that implements a home automation system able to detect and learn the user's behavior and help him/her accordingly, through an adaptive user interface that supports different functional areas: meal preparation, household appliances setting etc., *[3.3. Environment Model].*

Table 7. *System functions related to the kitchen household appliances*

Household appliance	*Function*	*User → System*	*System → User*
OVEN	On-Off	Turn the oven on/off	Feedback on machine status
	Cooking Program Setting	Select the cooking program from the list	Feedback on selected cooking program
		Set the cooking temperature	Feedback on set cooking temperature
		Set the cooking time	Cooking start time feedback set
	Start/Stop	Start/Stop the cooking	Feedback on cooking status
		Schedule the cooking start time	Feedback on scheduled start time
	Cooking Monitoring	Monitoring the cooking	Feedback on cooking progress
			Feedback on set temperature
			Feedback on cooking time (if it has been set)
	Automatic Setting	Select the Recipe	Set the cooking program according to the Recipe
			Feedback on cooking program
			Feedback on cooking temperature
			Feedback on cooking time (if it has been set)
	Instructions	Ask for an Information	Link to household appliance manual
		Ask for a support	Activate an interactive guide supports the user step by step
FRIDGE	On-Off	Turn the oven on/off	Feedback on machine status
	Temperature Setting	Set the fridgeTemperature	Feedback on temperature
		Set the freezer Temperature	Feedback on temperature
	Instructions	Ask for an Information	Link to household appliance manual
		Ask for a support	Activate an interactive guide supports the user step by step
DISHWASHER	On-Off	Turn the oven on/off	Feedback on machine status
	Wash Program Setting	Select the washing program from the list	Feedback on selected cooking program
		Set the washing options	Feedback on selected options
		Set the washing time	Feedback on washing time (if it has been set)
	Start/Stop	Start/Stop the washing	Feedback on washing status
		Schedule the washing start time	Feedback on scheduled start time
	Washing Monitoring	Monitoring the washing	Feedback on washing progress
			Feedback on washing time (if it has been set)
	Instructions	Ask for an Information	Link to household appliance manual
		Ask for a support	Activate an interactive guide supports the user step by step
RECIPE	Recipe consultation	Select the Recipe	CookBook
	Start Recipe in Wizard Mode	Select the Wizard mode	An interactive guide that provides step-by-step support for the user is provided

5.2.3 Definition of Use Case

Afterward, in order to define the adaptable and adaptive elements of the system, several use cases have been defined. In detail for each considered household appliance the Adaptable elements that depend on the user profile (zoom, contrast, etc.) and the Adaptive elements, which depend on the use of the system, have been identified. For each use case the involved functions, the rules and the goals have been defined *[3.4.1 The first level]*. For example, those concerning the cooking with the oven are shown in the table below:

Table 8. *Oven Use Case*

Description: Programming, starting and monitoring of the appliance (oven)	
The user turns on the oven. The user interacts with the interface to set, start and monitor the cooking program until it is completed.	
Involved Functions	
Switch on - switch off (ie, connection - disconnection from the home automation network)	
Cooking Program Setup	
Start /Stop cooking program	
Cooking Monitoring	
Adaptable elements	
ON/OFF Commands	Depending on user profile, different type of interaction are enabled (e.g. tap, swice, zoom, contrast etc.,)
Cooking Program Selection	
Cooking Temperature	
Cooking Duration Setting	
Cooking Start Time Setting	
Start/Stop Command	
Adaptive elements	
Program Setup Commands	According to the user's profile and his/her previous interactions, the system suggests the cooking program by highlighting it, setting it as default, or in any case facilitating its selection.
Temperature setting controls	According to the user's profile and his/her previous interactions, the system suggests the cooking temperature by highlighting it, setting it as default, or in any case facilitating its selection.
Cooking time setting commands	According to the user's profile and his/her previous interactions, the system suggests the cooking duration by highlighting it, setting it as default, or in any case facilitating its selection.
Setting commands for the delayed start of the cooking cycle	According to the user's profile and his/her previous interactions, the system suggests the delay start by highlighting it, setting it as default, or in any case facilitating its selection.

5.3. Architecture Application

Subsequently, the system architecture presented in [*Chapter 4, Architecture*] has been applied. It consists of three main modules, previously described: Database Management System (DBMS), Core, and Interface. The user-system interaction, implemented on a mobile device (e.g. tablet), is recognized by the system in terms of user profile and action. It occurs through a constant communication between Core

and DBMS, which also allows increasing the network knowledge and managing the adaptation. As a consequence, the Interface content and graphics elements are subject to change.

5.3.1 Database Application

This module is composed of 4 sub-modules, so structured in the case of real kitchen use *[4.2. Database]:*

- User Features Profile contains information on the user profile, in terms of personal data and health status. The profiles identified are three: U1 (visual disturbance), U2 (cognitive disorder), U3 (fine motor disorder).

- Log Adaptation Actions, which is the log of adaptation actions. In the specific case of the kitchen for each user-system interaction, it will collect the information described in the table below

Table 9: *Log Adaptation Action (kitchen case study)*

Log Adaptation Action	**Default setting smart object**	**OVEN**	Program adaptation event
			Counter number adaptations program
			Temperature adaptation event
			Counter number of adaptations temperature
			Duration adaptation event
			Counter number adaptations duration
		DISHWASHER	Program adaptation event
			Counter number adaptations program
			Program options adaptation event
			Counter number adaptations Options
		FRIDGE	Freezer Temperature adaptation event
			Counter number adaptations freezer temperature
			Fridge Temperature adaptation event
			Counter number adaptation fridge temperature
	Information detail	**RECIPE**	Recipe Adaptation event
			Counter number adaptation event
		Manual dishwasher	Help adaptation event
			Counter number adaptations manual
		Manual Fridge	Help adaptation event
			Counter number adaptations manual
		Manual oven	Help adaptation event
			Counter number adaptations manual

- User Use Profile, which is the user's usage profile. In the specific case of the kitchen, the following information will be collected in the database:

***Table 10**: User use profile (kitchen case study)*

User use profile	**Historical Interactions with the environment**	**OVEN**	Start/Stop Setting
			Program Setting
			Temperatur Setting
			Time Setting
		DISHWASHER	Start/Stop Setting
			Program Setting
			Options Setting
		FRIDGE	Freezer Temperature Setting
			Freezer Temperature Setting
	Historical Interactions with services	**RECIPE**	Recipe Selection
			Recipe Information
		Manual dishwasher	Help Selection
			Help Support
		Manual Fridge	Help Selection
			Help Support
		Manual oven	Help Selection
			Help Support

- Context Data represents the system on which all the data relevant for defining the user's context are collected.

***Table 11**: Context Data (kitchen case study)*

Context Data	**Smart Object**	**OVEN**	ON/OFF
			Program Status (Start/Stop)
			Program
			Temperature
			Duration Program
		DISHWASHER	ON/OFF
			Program Status (Start/Stop)
			Program
			Options
			Duration Program
		FRIDGE	ON/OFF
			Fridge Temperature
			Freezer Temperature
			Status Door (open/close)
	Sensor	**TEMPERATURE**	Temperature
		BRIGHTNESS	Level
	Time info	**TIME**	Time Event
		DATA	Data Event

5.3.2 Core Application

The adaptive module depends on the information stored in the database. The stored data will be used to perform adaptation actions on the user interface on a graphic (dynamic) and content level. The Core is composed of two engine mechanisms (Adaptable and Adaptive engine) and a DM Algorithm *[4.3. Core]*.

5.3.2.1 Adaptive Engine & DM Algorithm

The Adaptive Engine is composed of a Bayesian Network and a Decision-making algorithm. The Bayesian network has been designed by assuming the use of the system in a home environment, specifically in the kitchen environment, featuring three household appliances: oven, dishwasher and refrigerator. The network, consisting of a total of 25 knots (***Fig. 52***), is structured as follow:

1. Season: Consisting of four periods of the year (Autumn, Winter, Spring and Summer).
2. Week: Consisting of seven days of the week (Monday, Tuesday, Thursday, Friday, Saturday, Sunday)
3. Day_Range: Consisting of four-time bands of the day (Range1 = 06.01 - 12.00, Range2 = 12.01 - 18.00, Range3 = 18.01 - 0.00, Range4 = 0.01 - 6.00)
4. Household_Devices: Consisting of three appliances (Oven, Dishwasher, Fridge)
5. O_Information_Modality: Consisting of the GUI oven menu display modes ("Normal" or "Wizard")
6. O_CycleTarget: Consisting of the oven programs list (defrosting, leavened, food_warmer, static, airy, thermoventilated, grill, cleaning)
7. Help_level_O_cycleTarget: Consisting of the program list instruction manual for the oven ("Basic" or "Advanced")
8. O_CyclePeriod: Consisting of oven cooking times in minutes (Range0 = 5 – 25, Range1 = 30 – 60, Range2 = 65 – 120, Range3 = 125 – 180, Range4 = 185 – 240)
9. Help_level_O_cyclePeriod: Consisting of the cycle period instruction manual for the oven ("Basic" or "Advanced")
10. O_TemperatureTarget: Consisting of the oven temperature list (Range0 = 20-35°, Range1 = 40- 65°, Range2 = 70 – 120°, Range3 = 130 – 160°, Range4 = 170 – 220°, Rnage5 = 230 – 250°)
11. O_StartTime: Consisting of the time of the day on which the delayed start of the oven is set (Range0 = 0.01 - 1.00, Range1 = 1.01 - 2.00, Range2 = 2.01 -

3.00, Range3 = 3.01 - 4.00, Range4 = 4.01 - 5.00, …., Range23 = 23.01 - 0.00)

12. Help_level_O_startTime: Consisting of the delay start instruction manual for the oven (“Basic” or “Advanced”).

13. TemperatureFreezer: Consisting of the freezer temperature list (-18/-10 °C)

14. TemperatureFridge: Consisting of the fridge temperature list (0-7 °C)

15. F_Information_Modality: Consisting of the GUI fridge menu display modes (“Normal” or “Wizard”)

16. D_Information_Modality: Consisting of the GUI dishwasher menu display modes (“Normal” or “Wizard”)

17. D_CycleTarget: Consisting of the dishwasher programs list (pre_wash, fragile, eco, rapid, daily, intensive)

18. D_CycleSetting: Consists of dishwasher wash options (Top, Bottom, Both)

19. Help_level_D_cycleTarget: Consisting of the program list instruction manual for the dishwasher (“Basic” or “Advanced”)

20. D_CycleSoap: Consisting of washing options related to the presence/absence of detergent

21. D_StartTime: Consisting of the time of the day on which the delayed start of the dishwasher is set (Range0 = 0.01 - 1.00, Range1 = 1.01 - 2.00, Range2 = 2.01 - 3.00, Range3 = 3.01 - 4.00, Range4 = 4.01 - 5.00, …., Range23 = 23.01 - 0.00)

22. Help_level_D_startTime: Consisting of the delay start instruction manual for the dishwasher (“Basic” or “Advanced”).

23. Recipes: Consisting of recipes (Appetizier, First_course, Main_course, Side_dishes, Dessert)

24. R_Information_Modality: Consisting of the GUI recipe menu display modes (“Normal” or “Wizard”)

25. Recipes_Type: Consisting of recipes in the database (Recipe 1-5: Appetizier, Recipe 6-10: FirstCourse, Recipes 11-15: MainCourse, Recipes 15-17: SideDishes, Recipes 17-20: Dessert).

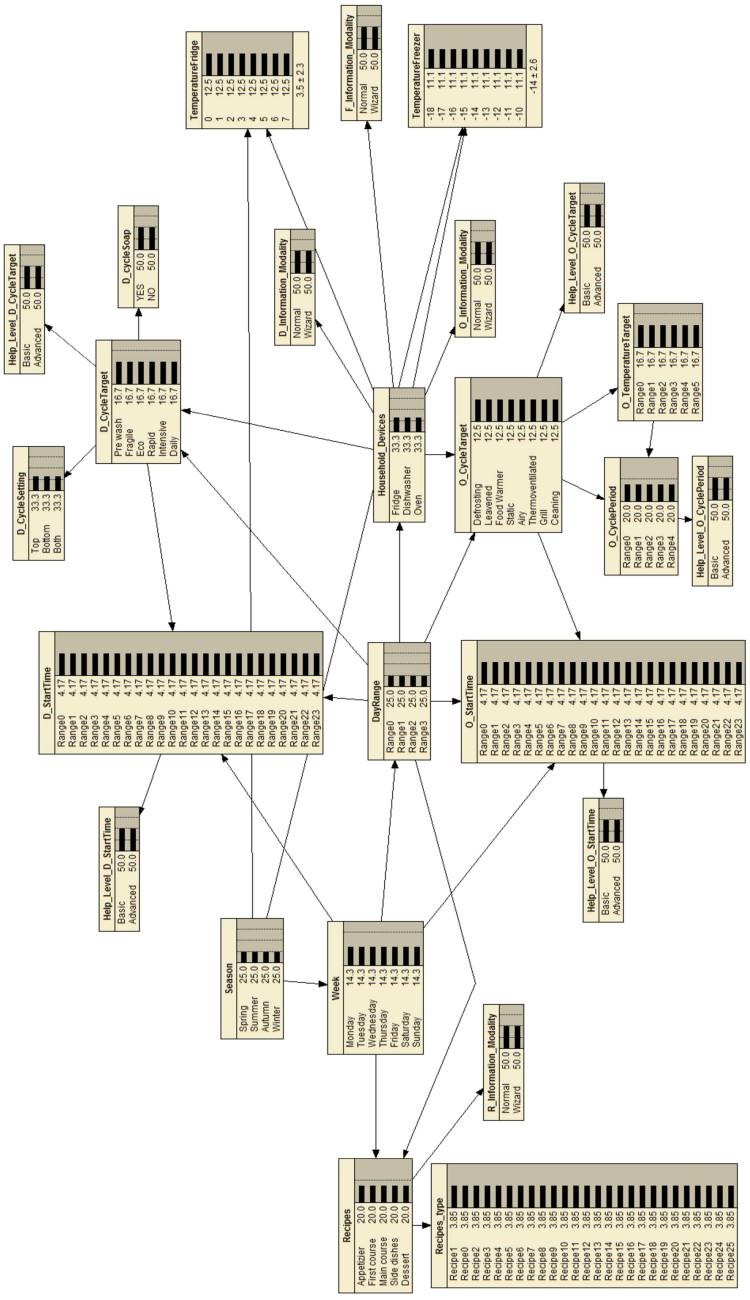


Fig. 52. The Use Case Bayesian Network

5.3.2.2 Adaptable Engine

The adaptable engine is based on the technical accessibility requirements and the corresponding control points for compliance verification defined on the basis of the Standards, Guidelines and success criteria contained in the Recommendation that the World Wide Web Consortium (W3C) - Web Accessibility Initiative (WAI) published on 11 December 2008 and which contains the Web Content Accessibility Guidelines 2.0 (WCAG 2.0) (Brewer & Dardailler 2000).

5.3.3 Interface Application

Finally, the Unified Scheme Functional was used to manage all the variables considered into the interface design [*4.4. Interface*]. It allows to represent in a single scheme the user behavior and the system actions, the interface variants and services and the adaptation and data management (Ceccacci et al., 2016). Using the SFU scheme, the UI functional model and different adaptation mechanisms have been designed. The
Fig. *53* ***s***hows the UI functional model developed for the oven management.
With the aim to support the greater number of end user, two different UI modalities were designed: Normal for users without cognitive dysfunction and characterized by a good technology attitude and Wizard for users who have not familiarity with technology and/or have some cognitive dysfunction. The first one is device-oriented, and it is designed to optimize the user interaction: through the UI, users are able to configure all device settings in the most preferred mode (e.g. program, temperature, duration and time in the oven setting). The second is task-oriented and it is designed to simplify the user interaction: tasks were decomposed into more simple and basic pre-programmed actions according to a specific and preset order and the system concludes each action after asking for users' confirmation. In this modality, the user must not manage and understand all the information of the appliance or a recipe's procedure: the system tries to make up for the lack of some user ability and it helps him/her to accomplish the activity of daily living independently.

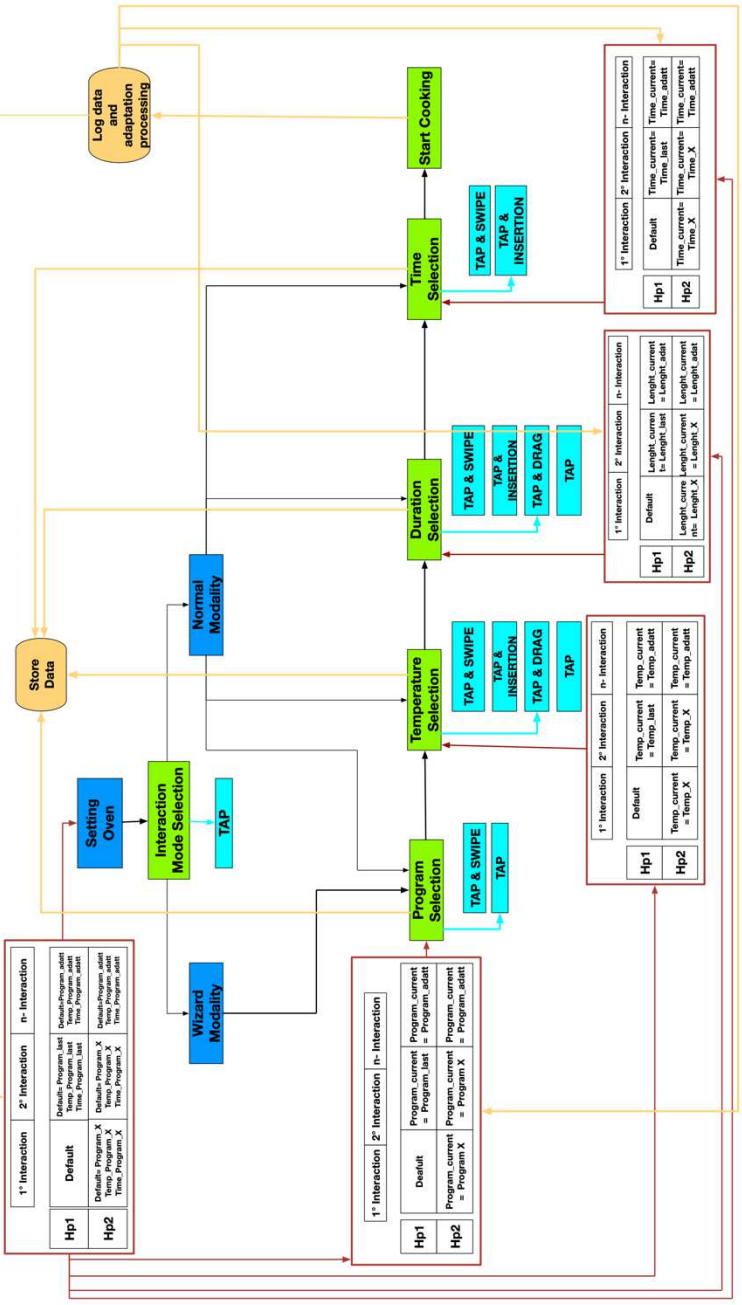


Fig. 53. The Unifed Scheme Functional (SFU)

To improve the usability, initially, three different modes of interaction have been built and integrated into the interface: tap, swipe and drag. Based on user preferences acquired by the system in the profile phase acquisition, the system will provide a specific interaction mode or a combination of these (i.e. tap and insertion, tap and swipe, tap and drag). In addition, in the functional model the system adaptation rules have been inserted. They determine the system response according to user actions and number of interactions.
In detail two functional rules hypotheses have been defined:

HP1

- at the *first interaction* with the device, the **default** Program/Temperature/Duration/Time will be suggested;
- at the *second interaction* with the device, the **last** used Program/Temperature/Duration/Time will be suggested;
- after *n-interactions*, the system will provide the **right adaptation action** to the user by suggesting program, temperature etc.

HP2

- at the *first interaction* with the device, a **specific** Program/Temperature/Duration/Time will be suggested;
- at the *second interaction* with the device, the **same specific** Program/Temperature/Duration/Time will be suggested;
- after *n-interactions*, the system will provide the **right adaptation action** to the user by suggesting program, temperature etc.

In accordance with the standards and the design guidelines (WCAG 2.0; DIS, I., 2009) the GUI has been designed for a touch screen application to provide high accessibility and usability for all kinds of users.
The adaptation mechanism changes both the graphic features and the contents of the interface. The graphics features have been designed to compensate any user's impairments such as color blindness, visual disturbances, motor problem. Using existing guidelines, the graphics features (e.g. color, text and buttons dimension, image size, etc.) and configurations associated with a specific disability have been designed.
Similarly, it has been done for the contents of the interface. The quantity and type of information that the interface gives to the user are designed on the user's abilities. The **HP1** hypothesis was applied during the execution phase.

5.3.3.1 Paper Prototype

On the basis of the SFU, the paper prototype for the oven, dishwasher and for the cookbook was finally developed. In particular, to ensure the accessibility of commands, depending on the motor skills and user preferences, different modes of interaction have been provided:

1) Tap + swipe + rotate
2) Tap + swipe
3) Tap + typing

By way of example, the oven paper prototype will be shown in ***Fig. 54,*** both in Wizard and Normal modalities. Depending on the user's preferences, acquired by the system during the profile acquisition phase, the system will provide a specific interaction modality (Normal or Wizard). At any time, the user can change the interaction mode preferences, through his profile.

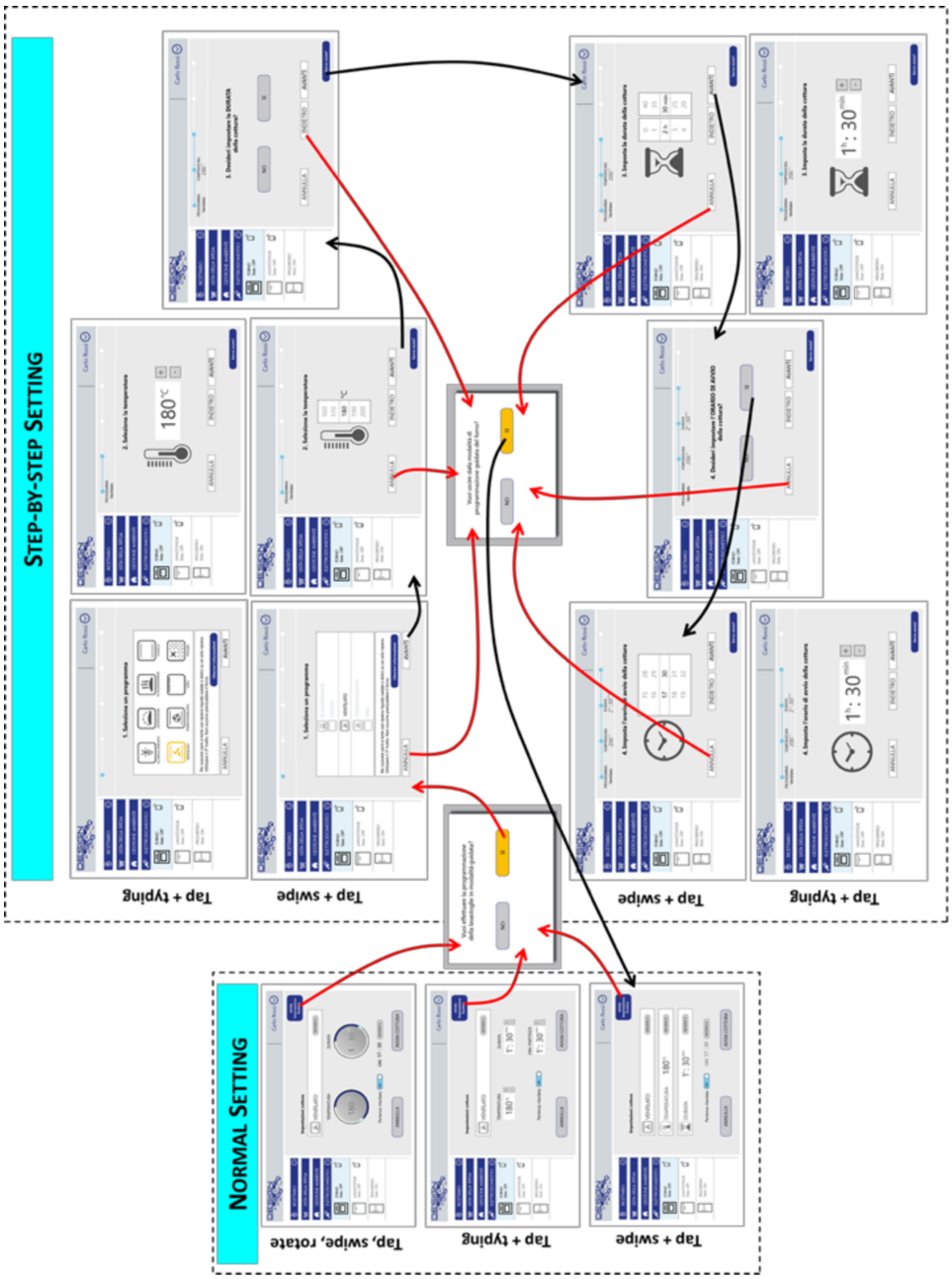


Fig. 54. Paper prototype. "Cooking program setting" menu: alternative ways of presenting information

Considering the Normal Setting, in ***Fig. 55*** the oven setting in "Tap + swipe + rotate" mode is shown. In detail, the user can modify the program by tapping on "modify" button (1), selecting the desired program from the appropriate menu (2) and confirming the choice. Consequently, the default temperature is updated according to the selected program (4). Furthermore, the user can change the temperature and the cooking duration assigned to the selected program using the appropriate knob (4,5). Finally, the user can choose to set the cooking start time (6) (optional). The cooking start time will be set to the current time. The timetable can be changed through a special menu, setting hours and minutes (7,8,9). Once the oven is set, the start cooking button will be enabled (11).

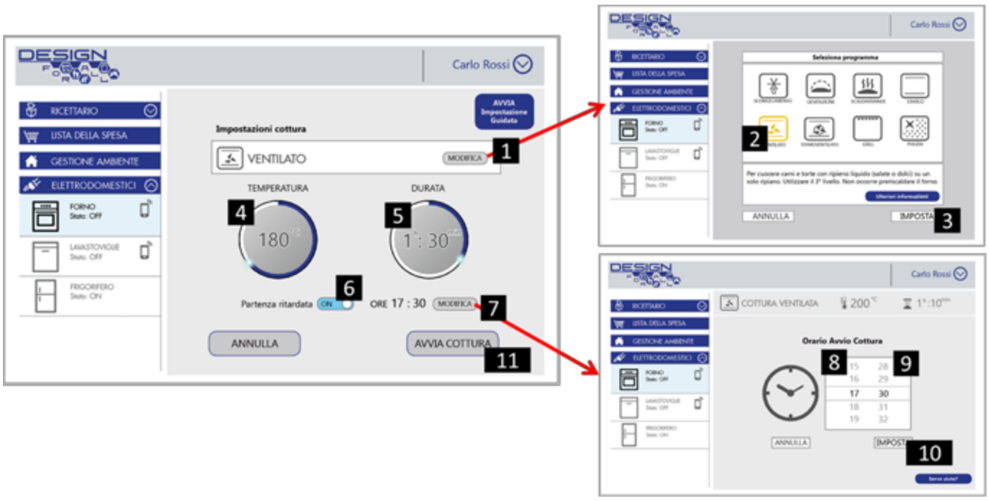


Fig. 55. Paper prototype: oven setting procedure "Normal Setting" mode with "tap + swipe + rotate" commands

In ***Fig. 56*** and ***Fig. 57*** the oven setting in "Tap + typing" and “Tap + swipe’’ mode is shown.

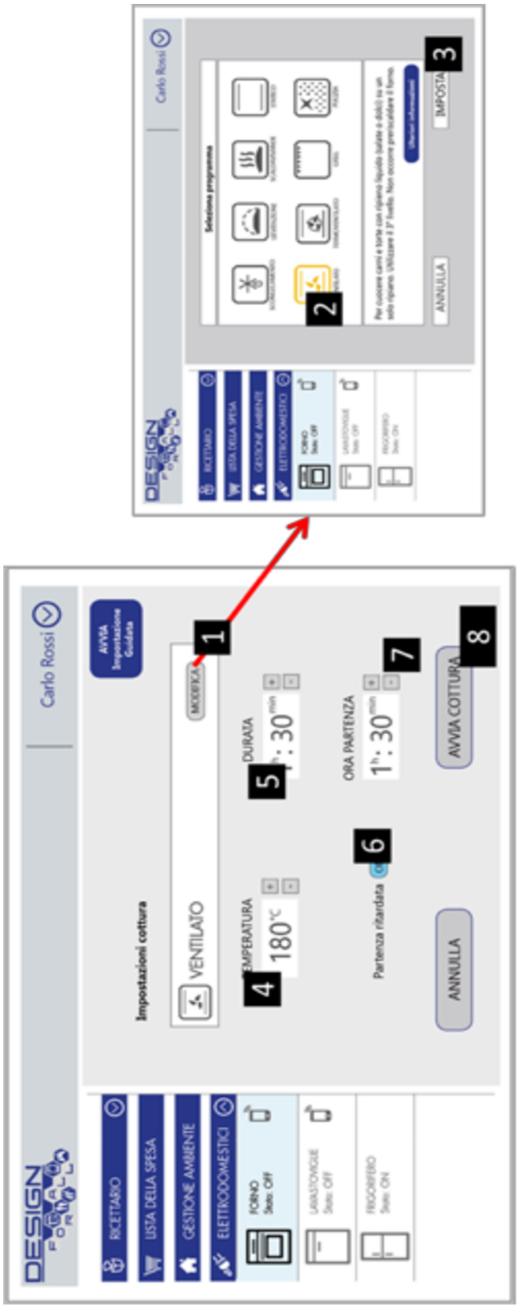


Fig. 56. Paper prototype: oven setting procedure "Normal Setting" mode with "tap + typing" commands

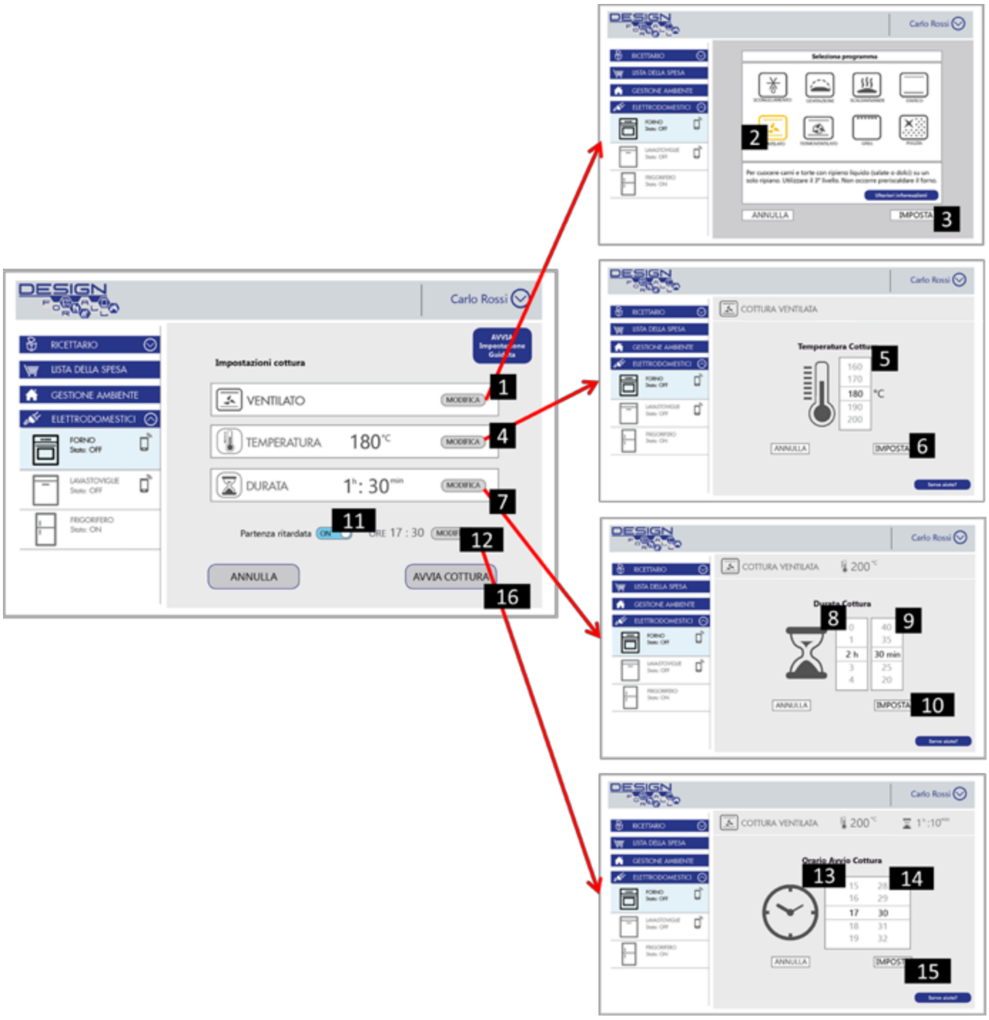


Fig. 57. Paper prototype: oven setting procedure "Normal Setting" mode with "tap + swipe" commands

In the case of the Wizard mode, designed for users with impairment or non-familiarity with technology, only two modes of interaction have been designed: “Tap + typing” and “Tap + swipe” (***Fig. 58***, ***Fig. 59***).

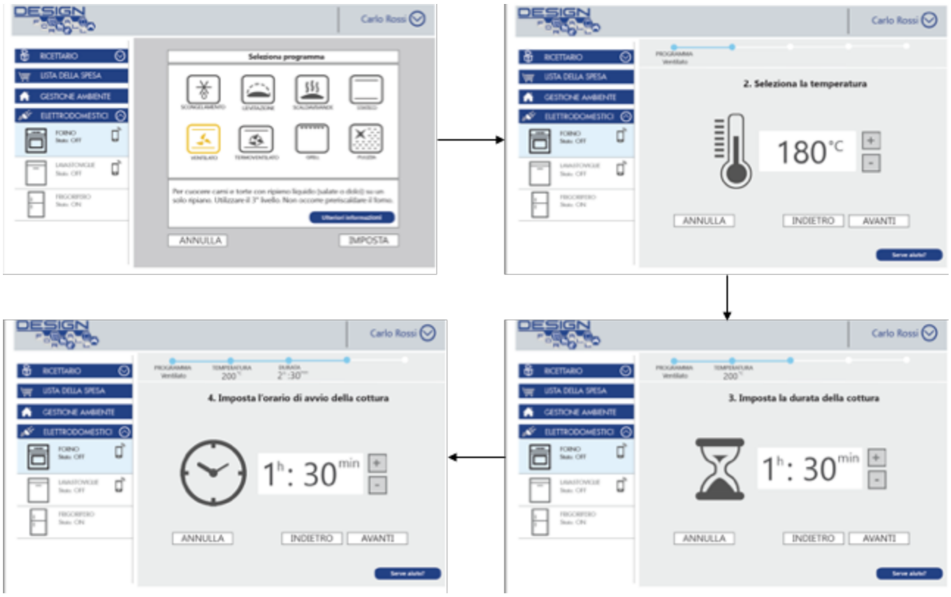


Fig. 58. Paper prototype: "Wizard" mode oven setting procedure with "tap + typing" commands

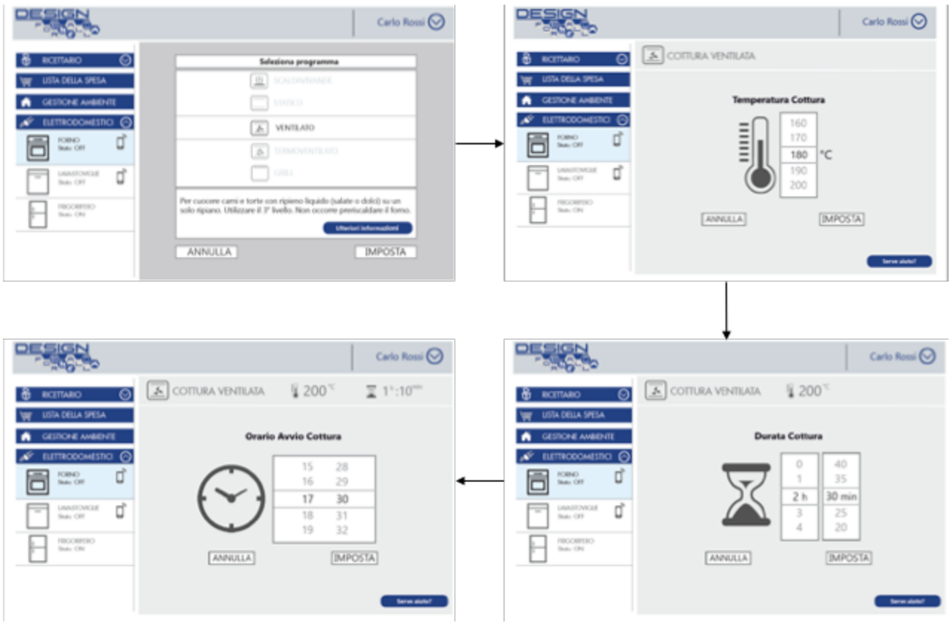


Fig. 59. Paper prototype: "Wizard" mode oven setting procedure with "tap + swipe" commands

5.3.3.2 Preliminary Expert Assessment

The presented paper prototype was subjected to a preliminary assessment of accessibility and usability by experts in the field. The evaluation was conducted at two levels:

- *In terms of accessibility,* the interface has been evaluated to determine the best interaction mode.
- *In terms of usability*, the interface was evaluated by analyzing the percentage of the task completion, the number of errors, the icons layout, the support requests number and the time completion

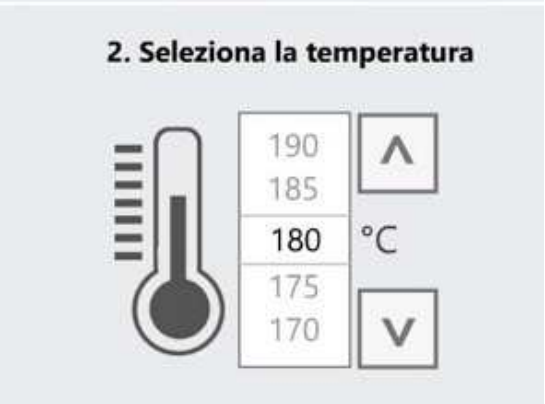


Fig. 60. Interaction mode chosen in the alpha prototype design

The results of this preliminary assessment show how the easiest gesture is the *tap*. In the case of the *swipe and the rotate*, it emerges the difficulty in controlling fine movements that can be problematic to use even for people with good motor control. For this reason, the "Tap + swipe + rotate" interaction mode was not used for the development of the final prototype, and the scroll was replaced with the addition of tap buttons (***Fig. 60***). Moreover, in general larger and more spaced keys increased the accessibility of the interface.

5.4. Final UI Prototype

Following the results obtained from the preliminary test, the final prototype was re-designed. In detail, the CookBook has been created, to search and execute recipes, and The Household Appliances Management to set three different appliances: oven, dishwasher and refrigerator.

5.4.1 CookBook

The Cookbook was developed to assist the user in preparing the meal and was designed for both Normal and Wizard modes. It is divided into two different pages: Recipe Search and Recipe Displaying.

5.4.1.1 Recipe Search

In the Recipe Search the user can select from the left menu the type of flow (e.g. appetizer, first course, second course, side dish and dessert) or can directly search for the recipe.

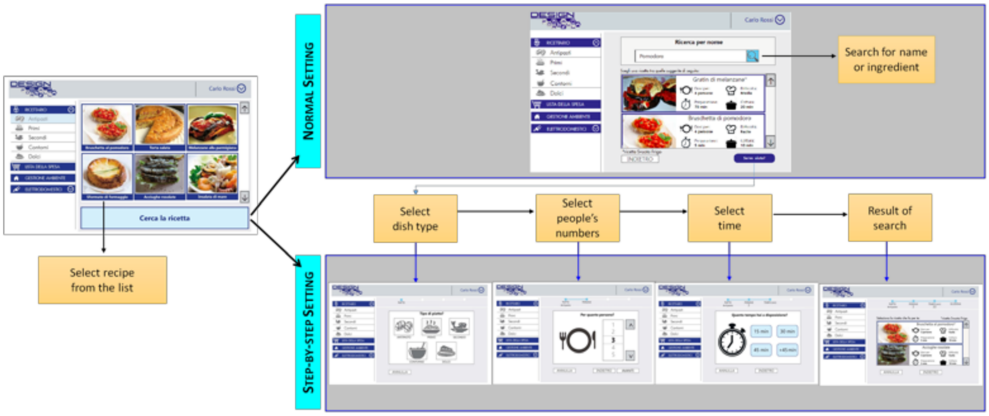


Fig. 61- Recipe book: recipe search in Normal and Wizard Setting

Fig. 61 shows the Recipe Search in the two modes Normal and Wizard. In the first case, the user types the name of the recipe or food and the system will provide a list of the recipes. In the second case, the system guides the user in the Recipe Search through a wizard route based on three different surveys: dish type, number's people, available time.

5.4.1.3 Recipe Displaying

Once the recipe is selected, the user is supported in preparing the meal. In the normal mode all the information is gathered in a single screen so as to make the display of the indications efficient. Specifically, in the left column the picture of the dish with the following specifications: number of persons, difficultiy levels, preparation time

and cooking has been inserted, while in the right column the ingredients and the description of the preparation phases has been introduced.

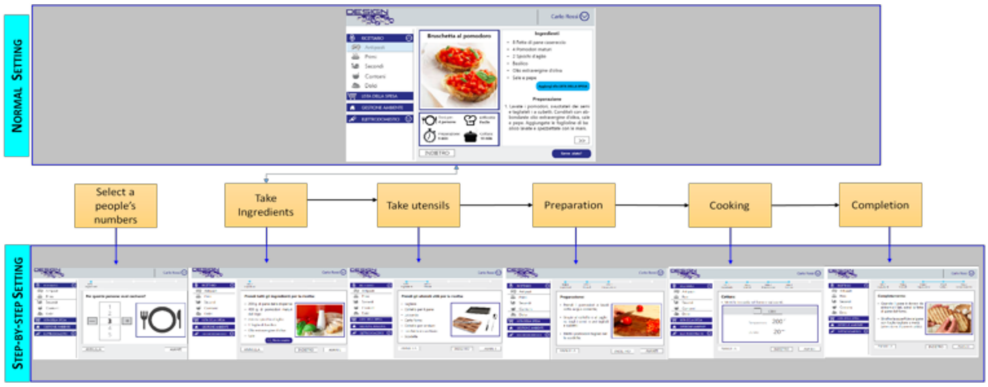


Fig. 62. Recipe book: recipe display in Normal and Wizard Setting

In the step by step setting mode, the recipe display is provided through an interactive guide that can support the user at every stage and ensure correct execution and completion. Once the number of people has been selected, the guide supports the user in the different phases: take the ingredients and the utensils, preparation and cooking the dish. For each complex operation, feedback has been added to allow the system to "control" the user's work.

5.4.2 Household Appliances Management

The Household Appliances Management was developed to assist the user in setting appliances and was designed for both Normal and Wizard modes. It is divided into three different pages: Oven, Dishwasher and Fridge.

5.4.2.1 Oven

The oven management interface has been simplified compared to the Paper prototype to optimize usability and accessibility. The ***Fig. 63*** shows the Oven Setting in the two different modes: Normal and Wizard.

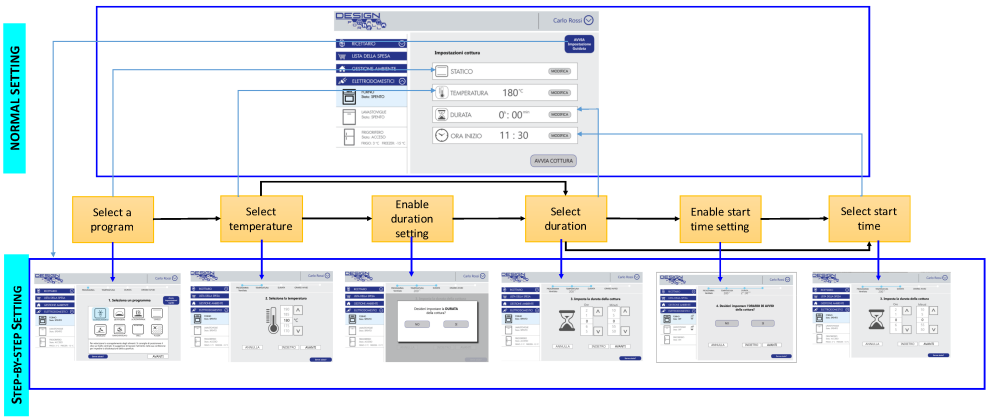


Fig. 63. Oven management in Normal and Wizard Setting

Following the results of the Preliminary Expert Assessment carried out on the paper prototype, the interaction mode "Tap + Swipe" (***Fig. 57***) was chosen for the oven screen in Normal mode. In this way, the commands for changing the temperature and duration, as well as controlling of the start delay, have been removed from the first screen (***Fig. 64***). In this way the user displays all the information for the oven setting in a single screen. The pages created for changing the settings (program, temperature, start time and delayed start) have been redesigned considering a single mode of interaction, more efficient and simpler. As an example, the screen for changing the cooking duration is shown (***Fig. 64*** - *right).* The user can change the hours and minutes through two interaction modes: "Tap + swipe" for the column number or "Tap" for the arrows on the side. The two modes coexist simultaneously on the same screen, so as to allow the user to choose the simplest mode. Moreover, on all the editing pages, help buttons have been inserted to support the user in managing the oven parameters (link to the step by step mode).



Fig. 64. Oven management in Normal Mode (on the left: first screen, on the right: duration management)

In the step by step mode, the interaction mode has been redesigned and has been made more accessible and usable. The system guides the user in the oven setting through four phases: the program (step 1), the temperature (step 2), the duration (step 3) and of the delayed start (step 4). At each stage a new help function has been added. An example is the help in setting the cooking program (***Fig. 65***). The system suggests the duration of cooking associated with the cooked food.



Fig. 65. Oven management in Wizard Mode (help function)

Finally, in the step by step mode, feedback was added for user control. In particular, they are remainders that allow, on the one hand, to improve the security of the system, preventing the oven from turning on without having inserted the pan, on the

other hand to remind the user with cognitive difficulties, to carry out all the necessary actions.

5.4.2.2 Dishwasher

As already described for the oven management, the interaction modes have also been standardized and optimized in the dishwasher control interface.
In the normal setting mode, the user manages the information related to the program, the washing options and the start time on the same screen. In step by step mode, the interface helps the user to set the washing program through a few simple steps: step 1, level of dirt in the dishes, step 2, duration of the wash time, step 3, indication of the dish position, 4 delay start. In this way it is the system that establishes the most correct washing program based on the feedback provided by the user.

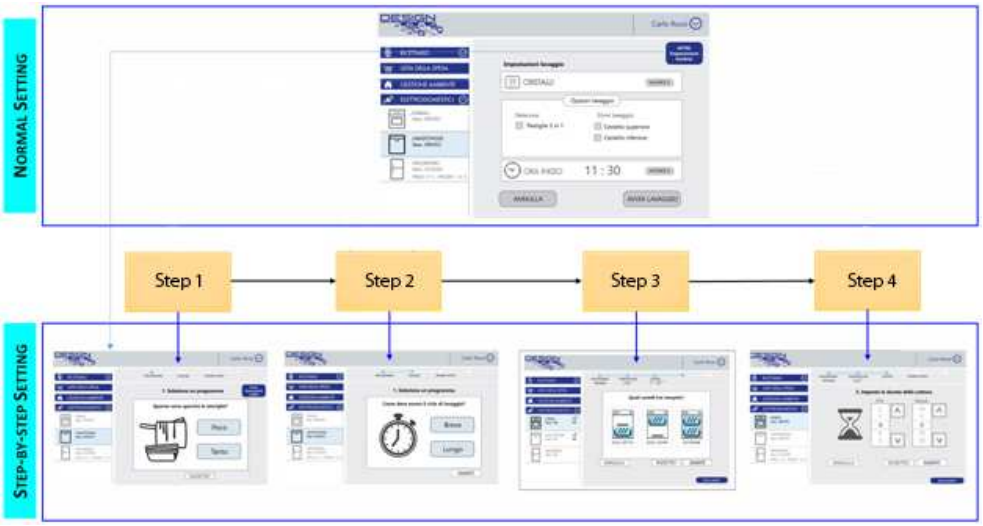


Fig. 66. Dishwasher Managemnt in Normal and Wizard Setting

5.4.2.3 Fridge

Finally, in the case of the fridge, in the normal setting mode it is possible to change the fridge and freezer temperatures directly. In step by step mode, the user is guided in the configuration of the settings. As already described for the other appliances, the management has been implemented with help buttons: the user is thus supported with further information and explanations.

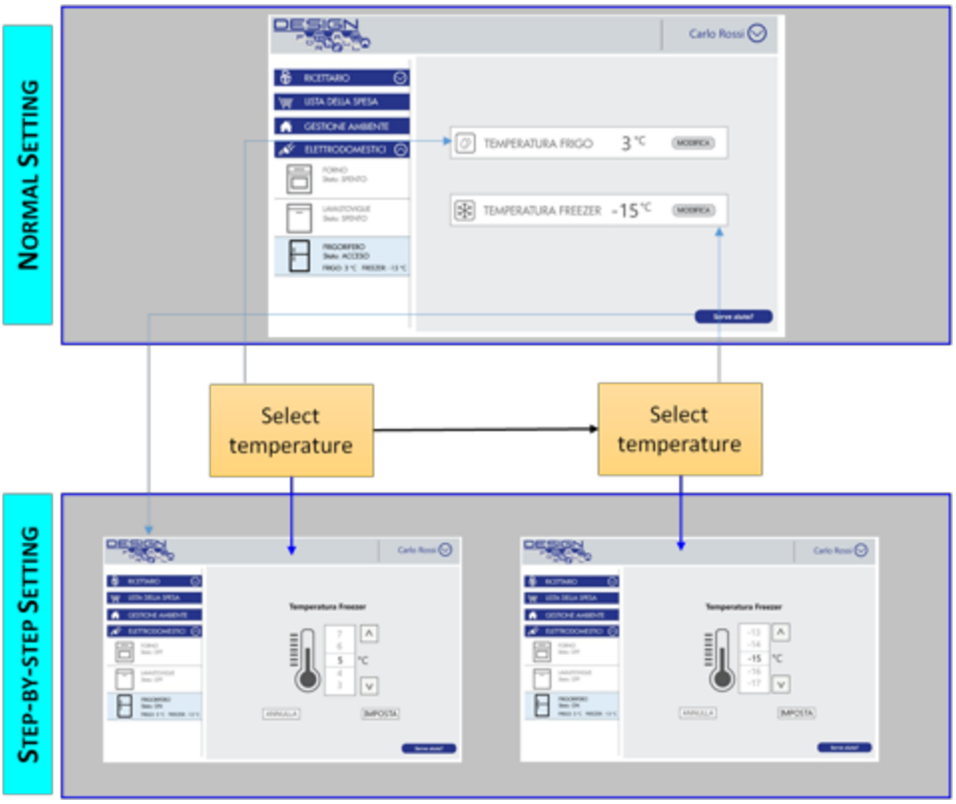


Fig. 67. Fridge Managment in normal and Wizard Setting

5.4.3 Software Architecture

The software architecture of the web application that runs on tablets, developed in collaboration with a Software House is a Single Page Application, which exploits

the AJAX and HTML5 techniques to create a fluid and responsive content without the need to reload the pages.
The user interface management system consists of two components:

- Server-side - WebApi which manages communication between and towards home automation components (simulated)
- Client-side - web application in the phone browser and communicates with the server through a REST API

5.5. Virtual Reality Environment Design

In order to evaluate and test the Graphical User Interface (GUI) and the above-presented system a Virtual Reality interactive environment has been designed and developed.
The VR interactive environment has been used as the main tool for the evaluation of the GUI with subjects instead of a real environment. The reasons are many. Firstly, the VR interactive environment is a flexible tool, which can be modified in order to reproduce and simulate different environments and situations. In particular, once created, a VR interactive environment can be easily edited, shared, visualized with different tools, and can be also enriched with additional items of information. All these activities have a limited cost if compared with similar ones performed using physical prototypes, which in this case are made of whole rooms including furniture, appliances, etc. Moreover, this environment and its components can be managed and controlled both with remote systems (such as, in this case, the GUI) and by the subjects during the evaluation sessions. Furthermore, the use of Virtual Prototypes has proven to be an effective tool to use for the evaluation of objects in other research areas, such as in the product development process, where it is used more and more (Wang 2002; Carulli, et al., 2012; Caruso, et al., 2014; Bordegoni & Carulli 2016).
Consequently, all the components that traditionally constitute an Italian kitchen have been identified, in order to consider their integration into the VR environment: under-counter furniture, with drawers or doors, and wall units, with hinged doors; built-in or free-standing home appliances: refrigerator, freezer, hob, oven, microwave, dishwasher, and washing machine and worktop.
The VR interactive environment was developed using the Unity3D software (https://unity3d.com/), a software tool largely used for the development of VR applications and used in the Virtual Environment laboratories of Istituto Auxologico

Italiano and Università Politecnica delle Marche. The VR environment supported the validation of the adaptive GUI with regard to the control of the household appliances and the control of the environment, both in the case of users without cognitive dysfunction and characterized by a good technology attitude (normal setup), and users who have not familiarity with technology and/or with some cognitive dysfunction (wizard setup). In order to test the adaptive interface in the VR system the GUI and the VR environment are directly connected. In details, the adaptive GUI makes visible the inputs to the VR environment by using updated values of some control variables (as a result of events, such as clicking on the "next" button by the user), so that they can be read in a client/server system (***Fig. 68***). Similarly, the VR environment makes visible the updated values of the variables (according to the events in the scene, such as "open the fridge door" or "illuminated shelves"), so that they can be read in a client/server system from the adaptive GUI. In this case through the Database the data about the users' profile (age, gender, technology attitude, possible cognitive dysfunction etc.) and the real home environment measured by sensors (brightness, temperature, etc.) are collected and analyzed in order to adapt the GUI characteristics. The interaction between the user and the VR environment only occurred through the GUI, due to the fact that the use of other devices for interacting with the environment could influence the evaluation of the GUI. In fact, most of the devices used for interacting with VR environments involves the use of one or both hands (gloves, remote controls, etc.), and can affect the validation of the touch screen application.

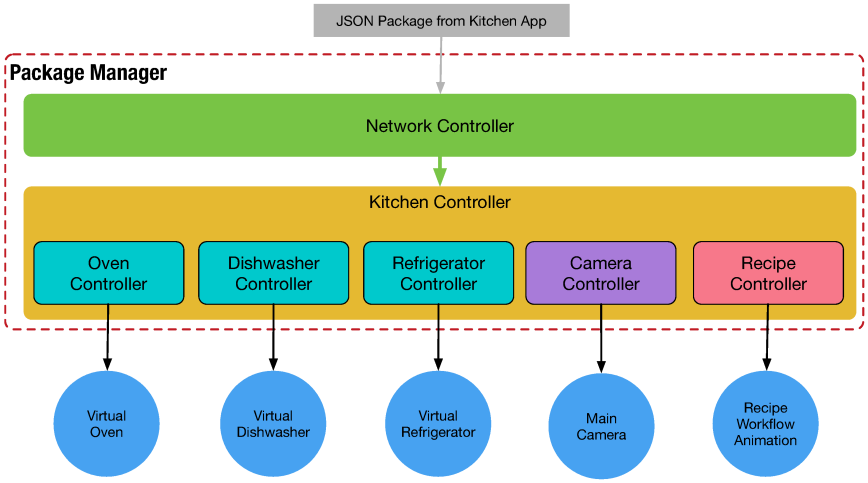


Fig. 68. Package Manger of Virtual Reality Envirnment and GUI communication

The virtual prototypes of the kitchen have been firstly developed by using various 3D modelling software tools (i.e., 3DStudioMax, Rhinoceros, Solidworks), depending on the characteristics of the following objects: the structural elements of the environment (floor, ceiling, walls, window, door, lights, etc.); home appliances (fridge, oven, dishwasher, hob and hood); furniture (under-counter furniture, shelves, table and chairs); accessories (plates, glasses, bowls, pots, cutlery, etc.); food items (some in the fridge, like eggs, cheese and vegetables, and others in the cabinets, like potatoes and oil).
Then, these have been imported into the Unity3D software using various interchange data formats. Therefore, they have been assembled in order to create a static VR kitchen environment. Consequently, materials and textures have been applied to it, and several lights have been installed into the kitchen, to simulate artificial lights, and outside it, for simulating the sun.
Finally, the three home appliances (refrigerator, oven and dishwasher) have been made dynamic, that is, manageable by the adaptive GUI. Specifically, using the "home appliances" section of the adaptive interface, it is possible to individually select the refrigerator, the oven and the dishwasher and, for each one it is possible to set the program, the temperature, the delay start and time (***Fig. 69***).

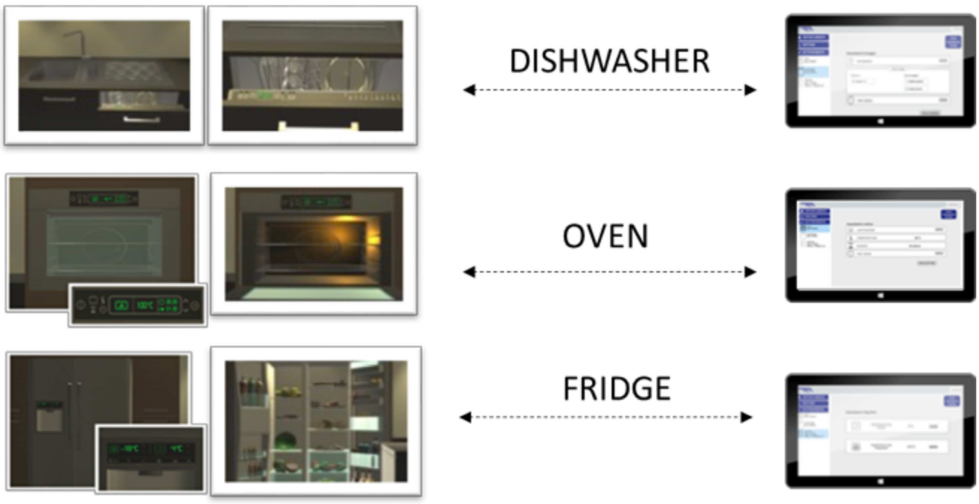


Fig. 69. Communication between VR and UI



Fig. 70. The Virtual Reality Environment

5.6. Validation

5.6.1 Introduction

To verify the functioning of the entire adaptive system, simulation tests have been carried out to test the adaptation of the network in the kitchen environment in VR managed by the Tablet.

5.6.2 Bayesian Network Simulation

On the basis of the user profile and its interactions with the system, a trained network has been associated with each subject (U1, U2, U3). A total of 6 behaviors (two behaviors per subject) were simulated. The network learning lasted about 1 month, during which the network stored the user's behavior and interactions with the system without implementing adaptation. To allow the adaptation of the network to the user needs, avoiding inefficient behavior, the specific value of the thr_perc is defined. The threshold value chosen for the DM algorithm that correctly fix false positives and false negatives is *thr_perc = 20*.
The simulated behavior for the U3 profile in Normal mode and the Wizard behavior simulated for the U1 profile are shown in the following chapters.

5.6.2.1 Normal Modality

As an example, the trained network for the U3 profile is shown in ***Fig. 71***; the user prefers the oven and dishwasher appliances in the time window ranging from 18.00 to 00.00 pm, in the Normal mode. In particular, for the oven setting, the user prefers the static program at a temperature of 230-250°C, while for the dishwasher setting it prefers the Rapid program and the "Top" cycle setting. Once learning the system, a simulation test has been carried out to test the network and its adaptation.

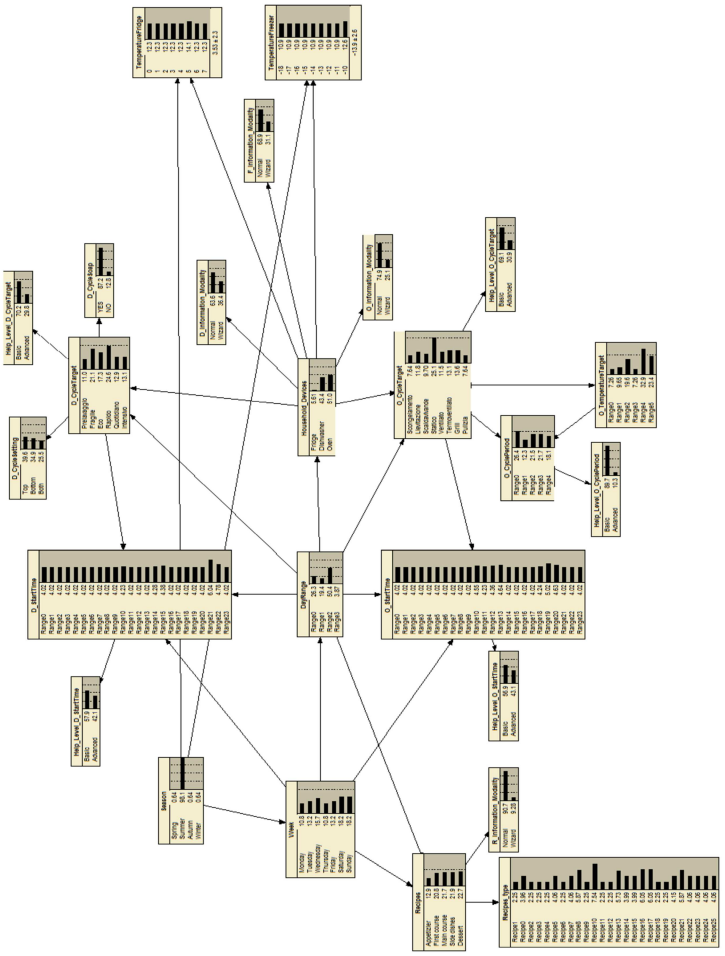


Fig. 71. The Normal Bayesian Network

In detail, by analyzing the probabilities existing between the nodes characterizing the BBN, it can be seen how the thr_perc set for the DM algorithm manages the correct adaptation action. Indeed, as can be seen in the ***Fig. 72*** and ***Fig. 73*****,** two are the simulation output: the "Filtered nodes" and the "Distributions with PP". In particular, the unfiltered knots are represented by probabilities that do not satisfy the specific condition (d>thr) such as Week, TemperatureFridge, O_start_time, Recipes and D_cycleSetting, and do not suggest any adaptation action.
Instead, the nodes with PP Distributions are represented by probabilities that satisfy the specific conditions and suggest the right adaptation action such as Season, DayRange, Household_Devices, etc., as shown in ***Fig. 73***.

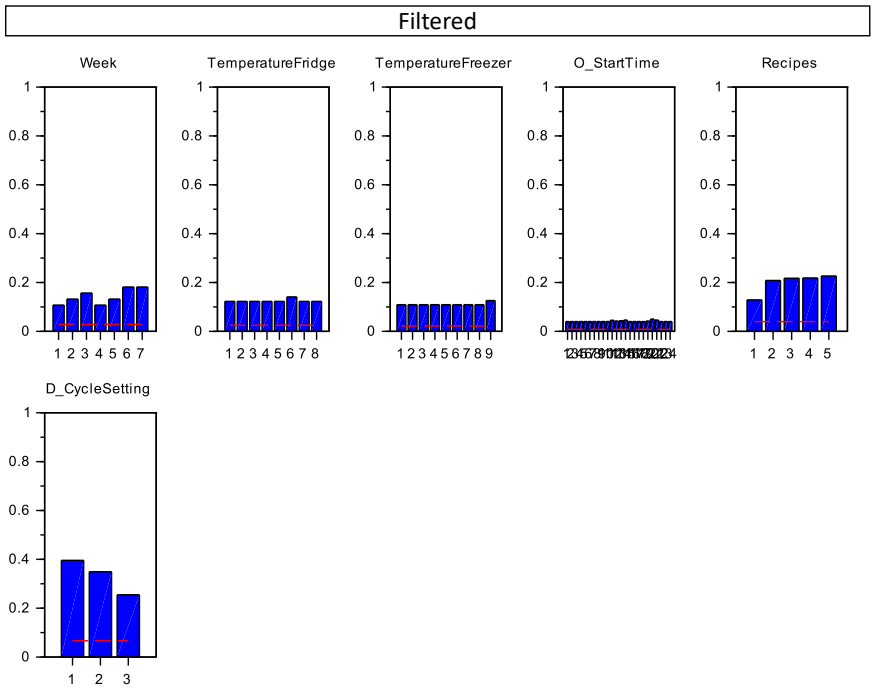


Fig. 72. Simulation with Filtered nodes in the Normal Mode

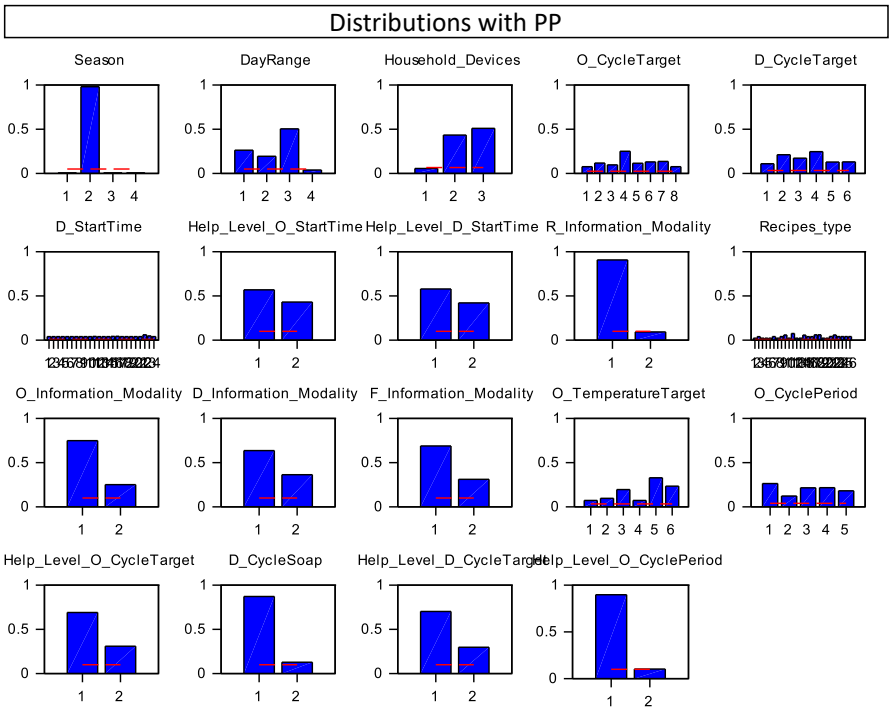


Fig. 73. Simulation with PP Distributions in the Normal Mode

5.6.2.2 Wizard Modality

The trained network for the U1 profile is shown in ***Fig. 74***; the user prefers the oven and dishwasher appliances in the time range from 18.00 to 00.00 pm, of the Wizard mode. In particular, for the oven setting, the user prefers the ventilated program at a temperature of 170-220°C, while for the dishwasher setting it prefers the Fragile program and the "Top" cycle setting. For the recipes, a preference for desserts is shown. Once learning the system, to test the network running and its adaptation a simulation test has been carried out.

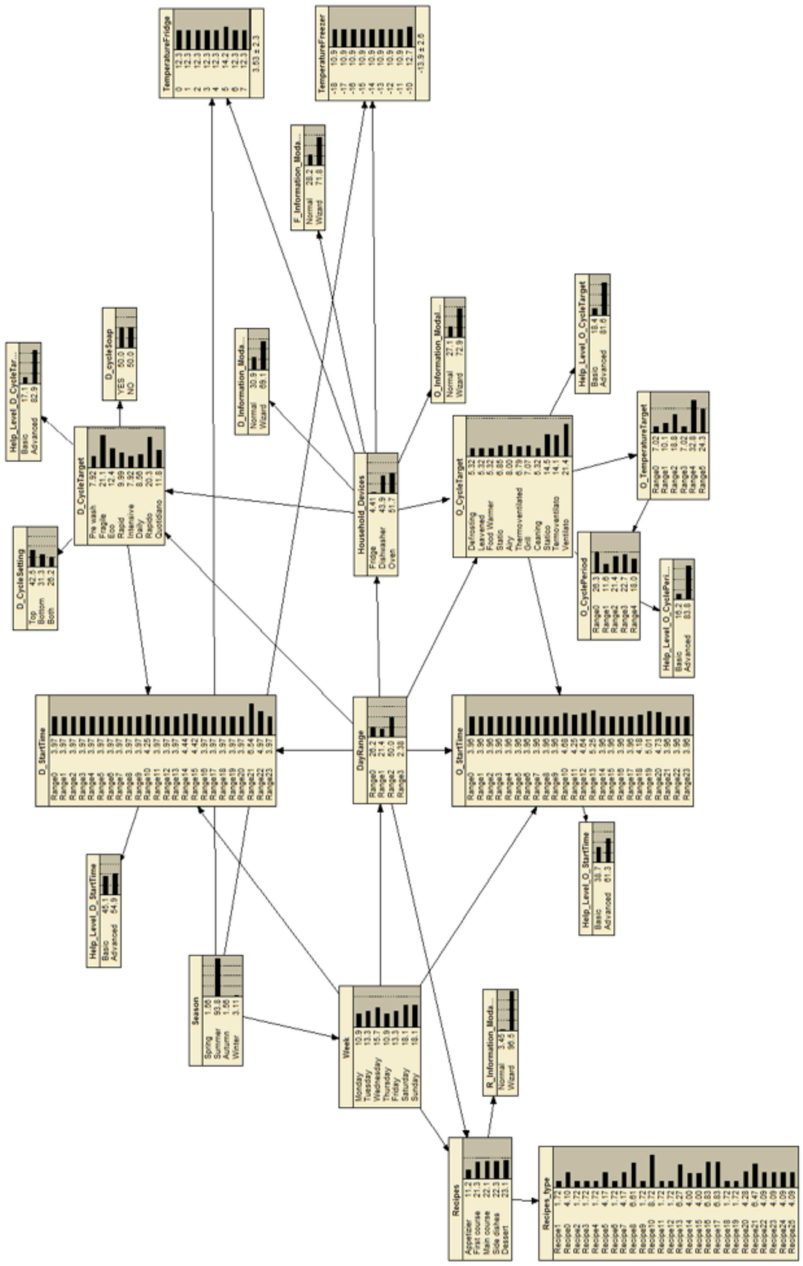


Fig. 74. The Wizard Bayesian Network

In detail, by analyzing the probabilities existing between the nodes characterizing the BBN, it can be seen how the thr_perc set for the DM algorithm manages the correct adaptation action. Indeed, as shown in the ***Fig. 75*****,** the unfiltered knots are Week, TemperatureFridge, O_start_time, Recipes, D_cycleTarget, O_CyclePeriod, etc.

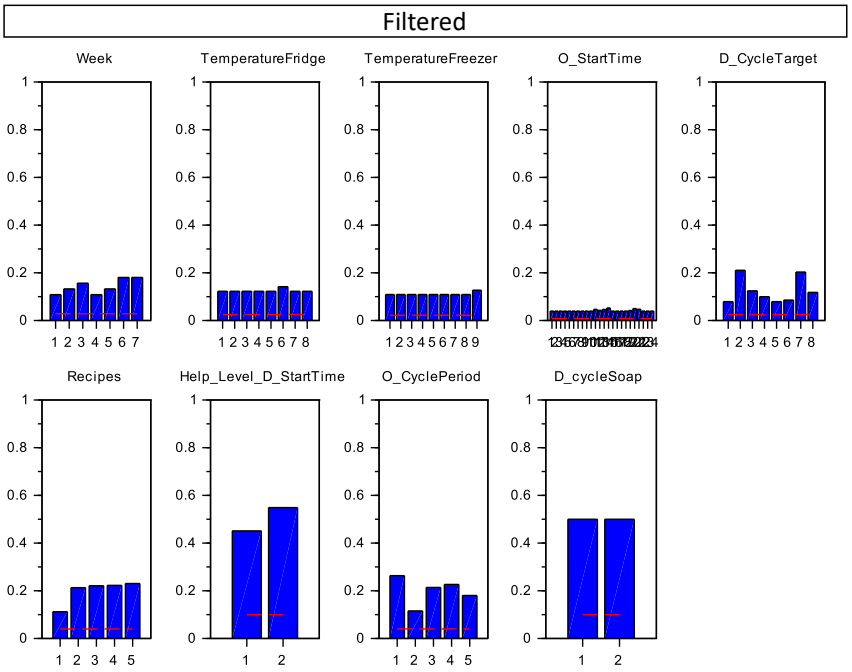


Fig. 75. Simulation with Filtered nodes in the Wizrad Mode

Instead, the nodes that suggest the right adaptation action are Season, DayRange, Household_Devices, Help_Level, etc., as shown below.

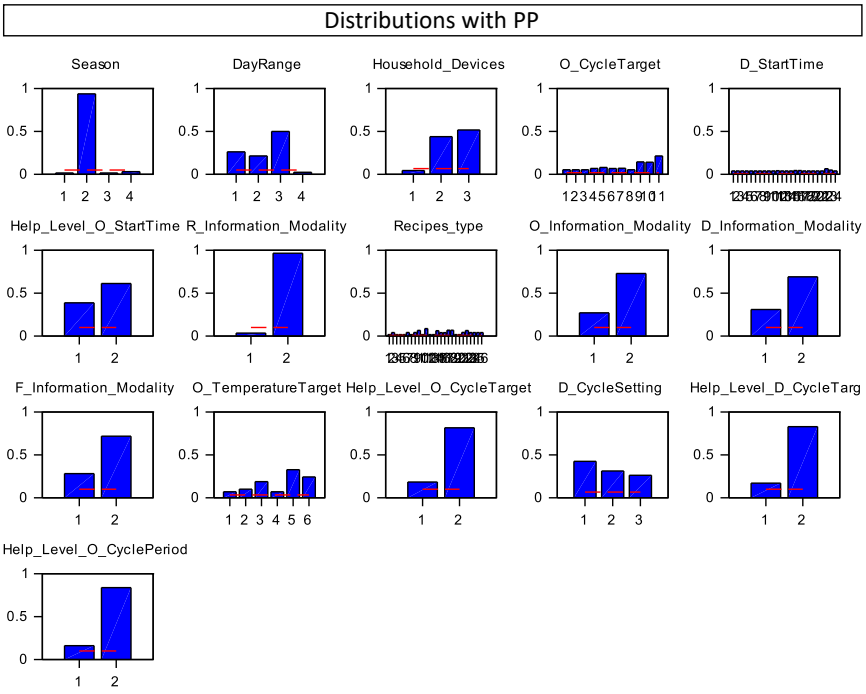


Fig. 76. Simulation with PP Distributions in the Wizard Mode

5.7.2.3 Normal - Wizard Modality Transition

Lastly, a final simulation test was carried out to test the transition from one mode to another (Normal-Wizard) during the system use phase (run-time). As initially specified, the user can switch from one mode to another through the special button located on the interface. This action is recorded by the system and stored in the User Use Profile.

The network shown in ***Fig. 77***, depicts how the nodes probabilities change during the use of the system and determine adaptation.

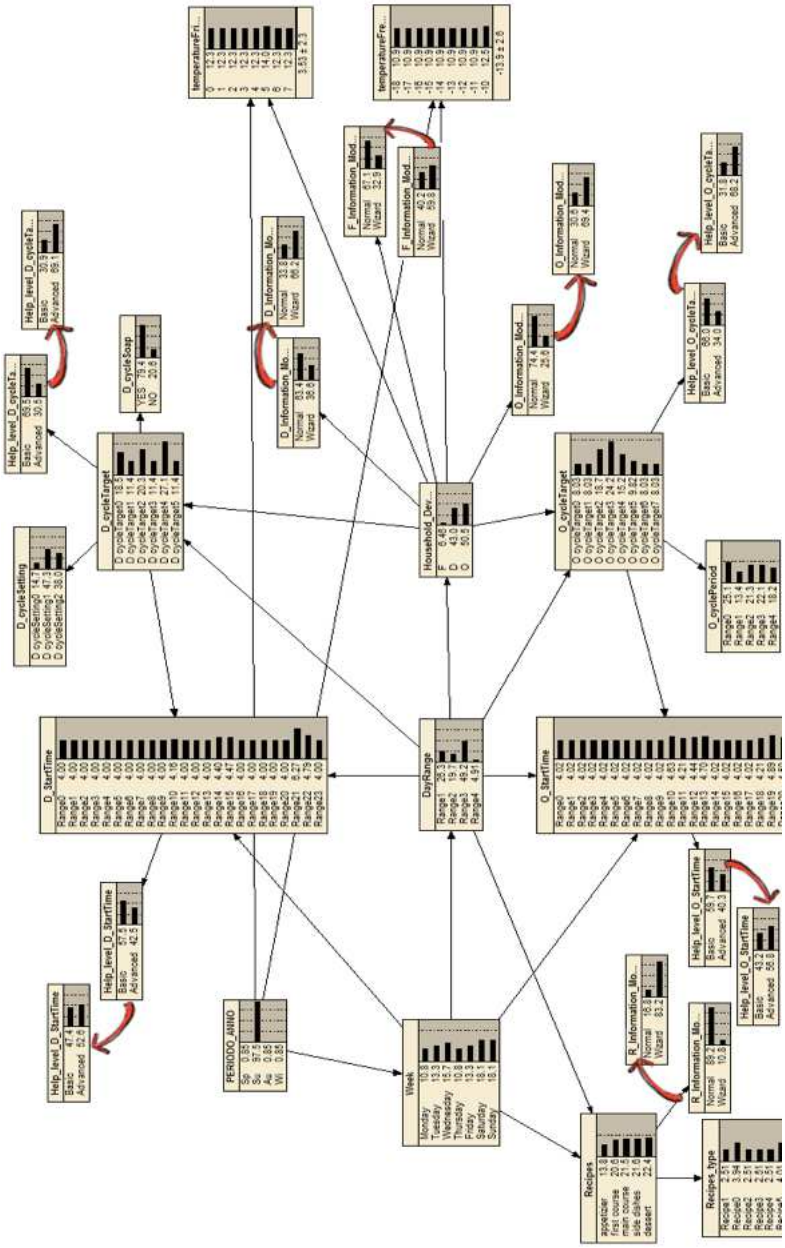


Fig. 77 The Adaptive Network: Normal and Wizard

In particular, the figure shows how the probabilities belonging to the control nodes are subjected to increases such as to generate adaptation: R_Information_Modality, Help_level_O_startTime, Help_level_D_startTime, Help_level_D_cycleTarget, D_Information_Modality, F_information_Modality, O_Information_Modality and Help_level_O_cycleTarget.

5.6.3 System Usability Evaluation

Once the goodness of the adaptation provided by the system has been simulated, a usability evaluation test has been carried out to assess the system in terms of usability for each user target category. This evaluation aimed to measure usability perceived by users during the first use of the smart system (***Fig. 78***).

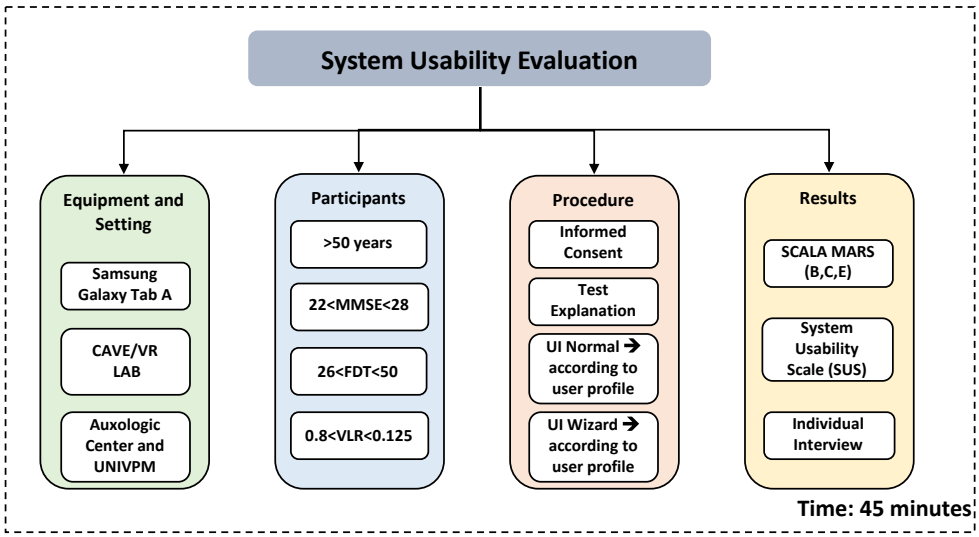


Fig. 78. System Usability Evaluation

5.6.3.1 Equipment and Setting

The adaptive GUI was presented on a Samsung Galaxy Tab A (10.1'') . The experiment took place both in the "Geriatric-Cardiovascular Department and Experimental Laboratory of Rehabilitation Research and Cerebrovascular Disease Collaboration", and at the Department of Industrial Engineering and Mathematical Science at the Università Politecnica delle Marche in an immersive virtual reality system called CAVE. The CAVE is a projection room consisting of four screens

placed in a mutual position on which the computer-generated environments are rear-projected and of different position sensors (trackers) applied for detecting patient's movements and transmit them to the computer, so that can modify the three-dimensional image according to the patient's point of view.

5.6.3.2 Participants

For the purpose of the study, a total of **6 patients** were recruited in this observational study (Nielsen and Landauer 1993). To be enrolled, the following inclusion criteria were fixed:

- Age 50 and older;
- Previous experience in the use of touch-screen technologies.
- For **Cognitive Impairment**: Mini Mental State Examination (MMSE) score between 22 and 28 (early and mild impaired patients who retained the physical and functional ability to participate in a cognitive or motor stimulation program) (U1, U3).
- For **Dexterity Problem:** Functional Dexterity Test score between 26-50 (Moderately and Minimally Functional) (U5, U6).
- For **Visual Problem**: Ranges of Vision Loss (based on visual acuity) score between Mild (<0.8 and >0.3) and Moderate (<0.3 and >0.125) Vision loss (U2, U4).

The total sample was composed of 6 participants, 3 males and 3 females, with a mean age of 60,33 years old (**SD±6.15**).

5.6.3.3 Procedure

Participants were welcomed in one of the laboratory facilities and were debriefed regarding the overall activity and goals. They read and signed the informed consent, the experimenter explicitly invited them to ask for any doubt at any time during the activity.

The experimenter then walked patients to the immersive reality system, CAVE, where the experiment set up was illustrated how, then patients received a hands-on explanation of how the adaptive GUI was structured and how to operate it.

The participants in the research were exposed to simulations of everyday living environments ("virtual kitchen environment"), through the aforementioned systems of immersive virtual reality. During the session, the participants could move within the simulated environment carrying out tasks indicated by the research operator. Specifically, subjects were asked to interact with the GUI to control and manage

some components of the kitchen, such as lightings and household appliances, and perform some activities, such as fill in a list of food items to be purchased, have suggestions and follow recipes. By using this approach, the authors kept the focus of the evaluation of the GUI, and not on the VR interactive environment, and on the interaction with it.
In detail, the system was presented to users and based on their profile and usage will provide the best UI mode (normal or wizard). At the end of the simulation, the participants filled out questionnaires that evaluated the user's experience in the interaction with the simulated interfaces (e.g. measuring the ease of use degree, learning difficulties, etc.). In detail the following surveys were carried out:

1. *MARS scale* (sections B, C, E);
2. *System Usability Scale* (SUS) which evaluates the usability of the interfaces;
3. *Individual interview* aimed at deeply examining in depth the dimensions of usability considered by the SUS questionnaire.

The session lasted *45 minutes* approximately.

5.6.3.4 Results

By analysing the results, we can observe that the proposed adaptive system can be considered usable for all considered user profile. Users performed a usability test of the Think Aloud Protocol based interface consisting of a sequence of 13 tasks designed in such a way as to test the main functionalities of the system. In detail, the System Usability Scale (SUS) questionnaire (Sauro 2011), calculated using the scores of all four subjects, reports an average score of **70.42** which indicates a good level of usability of the system, as indicated in Figure below.

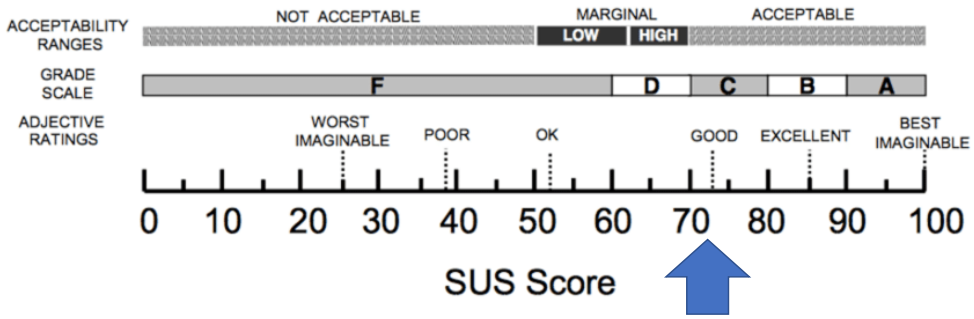


Fig. 79. The System Usability Scale (SUS) Score

Furthermore, the Mobile Rating Scoring App (MARS) (sections B, C, E) provides an overall positive evaluation of the interface. The average score of sample responses to section B, evaluating the functionality of the application, is 2.75 / 5 indicating a sufficient level of functionality. The average score of the sample's answers to section C, evaluating the application's aesthetics, is 3.23 / 5 indicating a discrete level of pleasantness. The average score of the sample responses to section E, evaluating the subjective quality of the application, is 2.79 / 5 indicating a sufficient level (***Table 12***).
With respect to the patients with Cognitive Impairments, the usability test shows that the interface is sufficiently usable and adaptable, but the most commons problems are comprehension and interaction with technology (scroll modality). Most of the patients showed great difficulties in using the tablet, in particular, they struggle to scroll the pages and to search the correct program. This problem is transversal to the various tasks and is present for all the subjects. It can be easily solved by implementing a complete and interactive tutorial able to explain both the use of the tablet and the various functionalities of the interface itself.
Alternatively, all the information could be entered on the same page, to the detriment of the images that for these patients were confusing. Furthermore, the introduction of more visible buttons and the insertion of some labels to clarify some commands would increase usability and error prevention.

Table 12. *The SUS and Mars Score*

	User 1	**User 2**	**User 3**	**User 4**	**User 5**	**User 6**	**Average Value**
SUS_1	3	3	3	2	4	3	
SUS_2	3	2	1	2	1	1	
SUS_3	3	4	4	4	4	5	
SUS_4	4	3	3	3	1	3	
SUS_5	4	5	2	2	4	4	
SUS_6	2	1	1	1	1	1	
SUS_7	3	5	3	3	3	4	
SUS_8	1	1	2	3	1	1	
SUS_9	3	4	5	5	4	4	
SUS_10	3	4	2	3	2	3	
SUS_TOT	57,5	75	70	60	82,5	77,5	**70,42**
MARS_B6	2	4	2	3	4	4	
MARS_B7	2	4	1	3	3	4	
MARS_B8	2	4	1	3	2	3	
MARS_B9	3	4	2	4	4	4	
MARS_B_Mean	2,25	4	1,5	3,25	3,25	3,75	**2,75**
MARS_C10	4	4	0	4	4	4	
MARS_C11	3	3	0	3	3	3	
MARS_C12	3	3	3	3	3	3	
MARS_C_Mean	3,3	3,3	3	3,3	3,3	3,3	**3,23**
MARS_E_20	2	5	2	3	4	3	
MARS_E_21	3	5	2	2	3	3	
MARS_E_22	2	4	1	2	1	2	
MARS_E_23	3	5	3	1	3	3	
MARS_E_Mean	2,5	4,75	2	2	2,75	2,75	**2,79**

The Table below summarizes problems that emerged from the two patients with Cognitive Deficit, the impact that these could have on usability, the solutions that could be put in place to solve these problems and the level of applicability of the same **(**

Table 13). By way of example, the oven setting task is shown.

Table 13. *Usability Evaluation with Cognitive Impairments Patients*

Task	*Problems*	*Impact*	*Suggested Solution*	*Solution Feasibility*
Oven Section	Slight difficulty in understanding how to select the section	MEDIUM	Insert Tutorial	Medium applicable
Set Static Program	Buttons too close	MEDIUM	Insert the Label	Easily applicable
Set 200°C Temperature	-	-	-	-
Set 2:45 Duration Time	-	-	-	-
Set the Delay Start	Difficulty in understanding the Delay Start setting	MEDIUM	Insert Tutorial/Help	Easily
Start the Cooking	The start cooking button is not detected immediately	LOW	Move to the top most key and highlight it	Easily applicable

With respect to the patients with Motor Deficit, the usability test shows that the interface is usable, and minor usability problems have been identified. The most common problems are due to the high proximity between buttons, which did not allow easy interaction with the tablet, causing errors such as selecting the wrong program or activating the help function. It has already been partially solved with the insertion of the adaptivity, but it can be further increased by the inserting white spaces to avoid the page overloaded. In addition, the zoom function accessible by double clicking on the tablet made the interaction with the technology problematic. For users with neuromotor deficit the zoom modality must be deactivated. Moreover, also in this case the introduction of a complete and interactive tutorial able to explain both the use of the tablet and the various functionalities of the interface itself has been requested by the users. Finally, during the interaction, one of the subjects has repeatedly reported the pleasure of the exercise, describing the tools as usable in a clear and intuitive way. He also appreciated what the practical aspects could be, highlighting how the application could be useful “not only for elderly people with difficulty in juggle in the kitchen but also for young people, very skilled in technology but often heavily engaged in daily work activities” (***Table 14***).

Table 14. *Usability Evaluation with Dexterity Problems Patients*

Task	*Problems*	*Impact*	*Suggested Solution*	*Solution Feasibility*
Oven Section	-	-	-	- -
Set Static Program	Buttons too close	MEDIUM	Insert the Label	Easily applicable
Set 200°C Temperature	Too small icons	-	Resize the icons size	Easily applicable
Set 2:45 Duration Time	-	-	-	-
Start the Cooking	The start cooking button is not detected immediately	LOW	Move to the top most key and highlight it	Easily applicable

Lastly, the interface has been evaluated with patients with Visual Impairment and a good system usability has emerged. Only few usability problems have been identified thanks to the UI adaptation with high contrast text and colour inversion. The most common problem is due to the slight difficulty in understanding how to open the household appliances menu. It has already been solved by leaving the menu bar fixed and always displaying the household appliances links (i.e., oven, dishwasher and fridge). Finally, the introduction of more visible buttons (e.g., more colour contrast with the text background colour) and the introduction of a complete tutorial (to explain both the use of the tablet and the interface) would increase usability and error prevention (***Table 15***).

Table 15. *Usability Evaluation with Visual Deficit Patients*

Task	*Problems*	*Impact*	*Suggested Solution*	*Solution Feasibility*
Oven Section	Slight difficulty in opening the appliance menu	LOW	Fix the drop-down menu	Medium applicable
Set Static Program	-	-	-	-
Set 200°C Temperature	-	-	-	-
Set 2:45 Duration Time	-	-	-	-
Start the Cooking	The start cooking button is not detected immediately	LOW	Move to the top most key and highlight it	Easily applicable

5.6.4 Heuristic Evaluation

A heuristic evaluation is a usability inspection method for computer software that helps to identify usability problems in the user interface design (UI) (Nielsen &

Molich 1990). In particular, this evaluation was conducted using Jakob Nielsen's heuristics (Nielsen 1994): visibility of system status, match between system and the real world, user control and freedom, consistency and standards, error prevention, recognition rather than recall, flexibility and efficiency of use, aesthetic and minimalist design, help users recognize, diagnose, and recover from errors and lastly help and documentation.

In order to test the usability and the accessibility of the system, in collaboration with the Italian Auxologic Institute, a team of five experts with the following profiles, has been recruited to conduct the evaluation: two geriatricians, two psychologists and an expert in human-computer interaction. Results, which are reported in the table below, highlight that the proposed solutions are suitable for all the considered profile (i.e., profile 1, cognitive impairment; profile 2, dexterity problems; profile 3, visually impaired).

Related to “Recognition rather than recall” heuristic, the same minor usability issue has been identified for Profile 2 and Profile 3: the knob, used for oven temperature and duration changing, may not be intuitive for an inexpert user. Such problem may be solved by adopting a different interaction modality. In addition, for Profile 1 a relevant usability issue has been detected, as the amount of information about each program is probably too much to correctly manage: this can be recover improving the wizard.

Table 16. *Nielsen's Heuristic Evaluation*

Nielsen's Heuristics	Profile 1		Profile 2		Profile 3	
	Home Appliances	Cookbook	Home Appliances	Cookbook	Home Appliances	Cookbooks
Visibility of system status	Yes	Yes	Yes	Yes	Perception Problem	Perception Problem
Match between system and the real world	Yes	Yes	Yes	Yes	Yes	Yes
User control and freedom	Yes	Yes	Yes	Yes	Yes	Yes
Consistency and standards	Excellent	Excellent	Excellent	Excellent	The difference in the interaction mode (tap, swipe and rotate) could generate user frustration	Excellent
Error prevention	Excellent	Excellent	Excellent	Excellent	Excellent	Excellent
Recognition rather than recall	The amount of information about each program are probably too much to be correctly manage	The amount of information about each recipe are probably too much to be correctly manage	The knob, used for oven temperature and duration changing, may not be intuitive for a inexpert user		The KNOB, used for oven temperature and duration changing, may not be intuitive for a inexpert user Some items may not be seen by the user	The images do not logically accompany the steps described in the preparation of the recipe. It could cause confusion
Flexibility and efficiency of use	Yes	Yes	Yes	Yes	Yes	Yes
Aesthetic and minimalist design	Functional Design	Functional Design	Functional Design	Functional Design	Functional Design	Functional Design
Help users recognize	No, there is no way to check for an error, but you can cancel any action.	No, there is no way to check for an error, but you can cancel any action.	No, there is no way to check for an error, but you can cancel any action.	No, there is no way to check for an error, but you can cancel any action.	No, there is no way to check for an error, but you can cancel any action.	No, there is no way to check for an error, but you can cancel any action.
Help and documentation	Yes	To be improve	Yes	To be improve	Excellent	Absent

Chapter 6.

The Second Case Study: Re-design of the Hoover Wizard Application to make it an Adaptive System

6.1. Introduction

Adaptive Interface (AI) represents a new way to improve the GUI usability and to help the users to carry out their tasks efficiently and enjoyable (Lavie & Meyer 2010). Indeed, the AI adapts and customizes automatically its features and structure to the user characteristics, to facilitate users with different skill levels and impairments. (Haas & Hettinger 2001). In order to make the user experience more enjoyable and efficient, AI is designed and employed in many different contexts: for example, in domestic environments (Kushwaha et al., 2004; Gullà, et al. 2016), in workplaces application (Letsu-Dake & Ntuen 2010; Evans, et al., 2013) and healthcare domain (Vasilyeva et al., 2005; Shakshuki, et al., 2015). However, all these applications propose specific approaches for the development of new systems, without considering the opportunity to convert existing smart systems into adaptive systems.

For this reason, this second case study aims to redesign the existing Hoover Application, in order to make it adaptive, so it can automatically customize the content and how it interacts with the user by monitoring the user's status, the system state and the current situation. The Hoover Wizard App is an application designed and developed by the Candy-Hoover Group that allows dialogue with the wide range of Hoover Group appliances. The application helps the user in carrying out daily activities and adds valuable features that make the product or set of products installed more intelligent and performing.

In this context, the insertion of the adaptive component thus allows to:

- Improve the use of domestic appliances through an advanced user interface and provide timely suggestions based on previous interactions, using behavioral prediction algorithms;

- Simplify the use and increase the efficiency of use over time, through Bayesian networks that can offer immediate and compliant practice to the user, leading to a familiar and practical interface;
- Automatically customize the content and the mode of interaction with the user by monitoring: system status, user status and current situation.

In this study we limited ourselves to redesign the system considering three connected devices: *Oven, Dishwasher and Washing Machine*.
Unlike the previous case study in which the presented methodology and architecture were applied in the design of an adaptive ex-novo system, this second case aims to demonstrate how to make an existing system adaptive through the re-design. Therefore, considering the constraints given by the existing system, and starting from the methodology and architecture previously described, a re-design of the Hoover Wizard Application, consisting of the different following steps, has been developed (***Fig. 80***):

- ***System Specifications Definition*:** The first step consists in studying the System and its specifications in terms of Architecture, Mobile Application and Database;
- ***Adaptation Goals and Action Definition***: The second step consists in defining the final goal of the project and the specific adaptation actions;
- ***Re-Design***: The re-design phase consists in redesigning the existing architecture and the Database to adapt it to the new structure;
- ***Back-end Translation Definition***: This module is responsible for reading the updated Database, filtering and translating it;
- ***Adaptive Engine Design***: This phase consists of the design of the three Bayesian networks built specifically for selected appliances;
- ***Cluster Definition***: This phase consists of the creation of different user behavior clusters, based on the real data extracted from the databases and used during the initialization phase of the system;
- ***DM Algorithm & Network Learning***; This phase consists of the description of the decision-making algorithm used to make the adaptation;

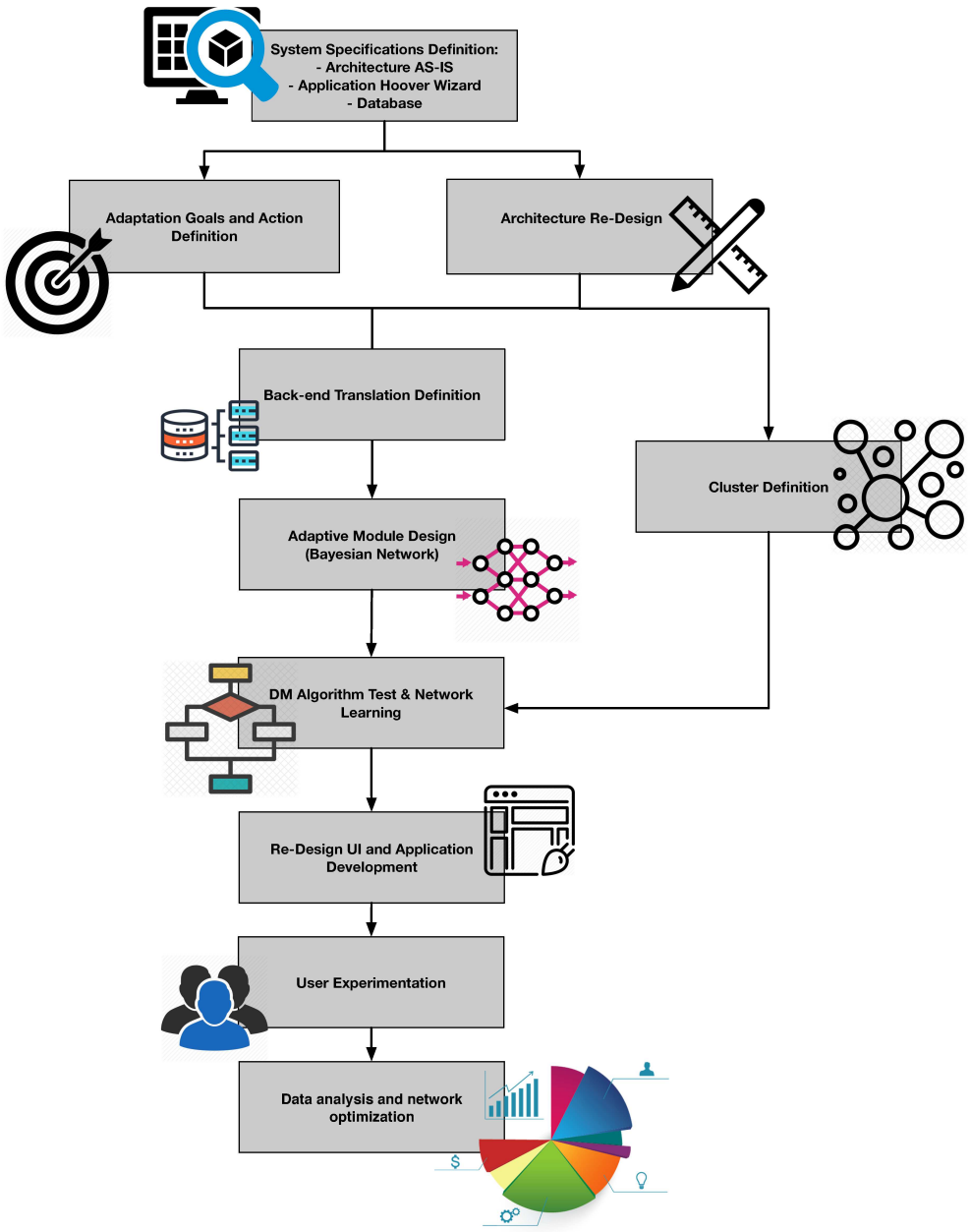


Fig. 80. The Work Flow

- ***Re-Design UI and Application Development***: This phase consists of the redesign of the graphic user interface to insert the adaptive feature;
- ***User Experimentation***: Is the experimentation phase with real users, to test the system;
- ***Data analysis***: The last step consists of the analysis of the data obtained following the experimentation and the subsequent optimization of the system / network.

6.2. System Specifications Definition

Not every system can just be turned into an adaptive one, so it is worth to specify some requirements that the existing system should meet for the proposed method to be applicable. They are the following:

- Communication and storing capabilities should be supported by the system to ensure the proper data flow;
- Capabilities of sensing contextual data from the user, the environment, the system itself. The system should be able to recognize the user and human-system interaction information;
- Adaptable elements and actions should be supported by the system. They allow introducing variability in system behavior based on the changeable requirements.

For this reason, the main functionalities of the existing system must be clear. The latter involves the study of the main system features such as architecture, framework, database, script code and layout, etc. It includes the characteristics of the application, the devices, and the physical features of a system as well as the relationships between elements *[3.3. Environment Model].* The three main elements of the system: Hoover Wizard Application, Architecture and Databases are better described in the following chapters.

6.2.1 Hoover Wizard Application Mapping

The mobile application allows the user to simultaneously manage a range of connected appliances such as washing machines, dishwashers, refrigerators, ovens, cooktops and hoods. Depending on the selected product, both the common and the more specific features are available; the user decides whether to use the default and recommended settings or to activate favorite functions.

Specifically, the transversal functions present on all products are:

- *Remote control*: the appliance is controlled even outside the home;
- *Constant monitoring*: allows to check what the appliance is doing at any time of the day;
- *Monitoring and management of consumption*, with suggestions on the most economical use based on energy tariffs;
- *Warnings and messages*: the appliance send the user information on its operation;
- *Customization of functions* according to their habits and needs;
- *Self-diagnosis*: the appliance constantly informs the user about its conditions.

Once the functions of the system have been identified, a complete mapping of the "Hoover Wizard" has been carried out. This phase represents a fundamental part of the thesis project because it has allowed us to study the structure of the application from both a logical and a functional point of view. Once logged, the user accesses a new application screen in which he selects the appliance (***Fig. 81***).



Fig. 81: The Home Screen Wizard App

Once the appliance has been recruited and selected access to a first screen is allowed in which it is possible to select the program, set the delayed start and run the household appliance. As an example, the Start page of the washing machine is shown in ***Fig. 82 a)***.

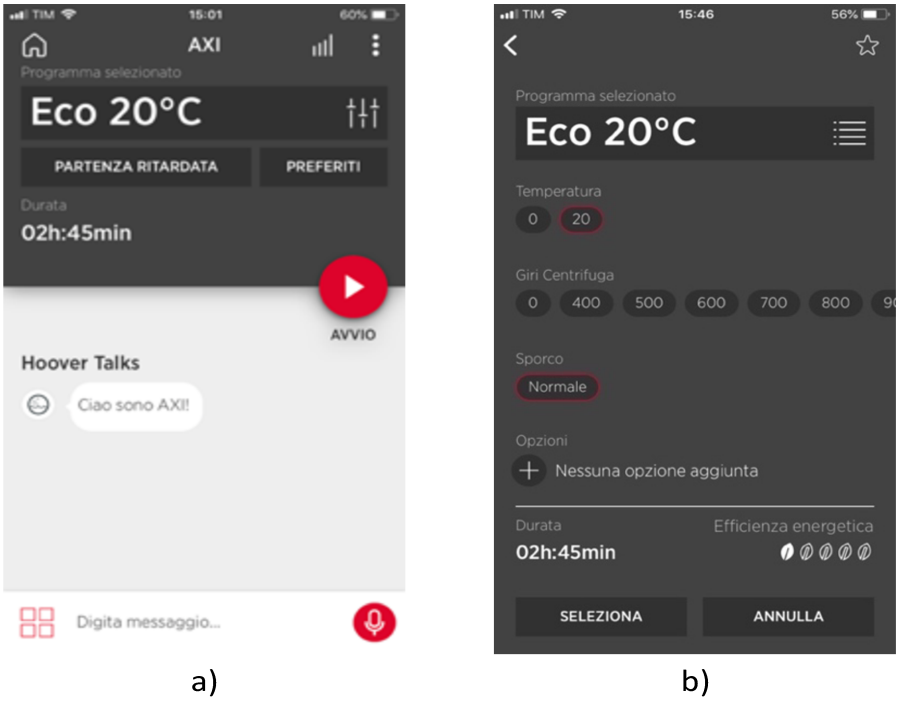


Fig. 82: a) The Hoover Wizard App: The Washing Start Page; b) The Hoover Wizard App: The Washer setting page

To change the program, it is necessary to select the appropriate "*Program selected*" box. In this way a new screen in which it will be possible to modify the program settings will be shown: the program type, temperature, etc. In ***Fig. 82 b)*** the washer setting page is shown. In detail, selecting the "*Selected program*" box from the Setting Page, a screen dedicated to selecting the device program among the standard program will be offered. In ***Fig. 83*** the selected washer program box is shown. Furthermore, for each selected appliance it can possibly add cooking or washing options and set the delayed start.

Fig. 84 and ***Fig. 85*** show an example in the case of washing machine of the options and delay start page.



Fig. 83: The Hoover Wizard App: The selected Washer Program box

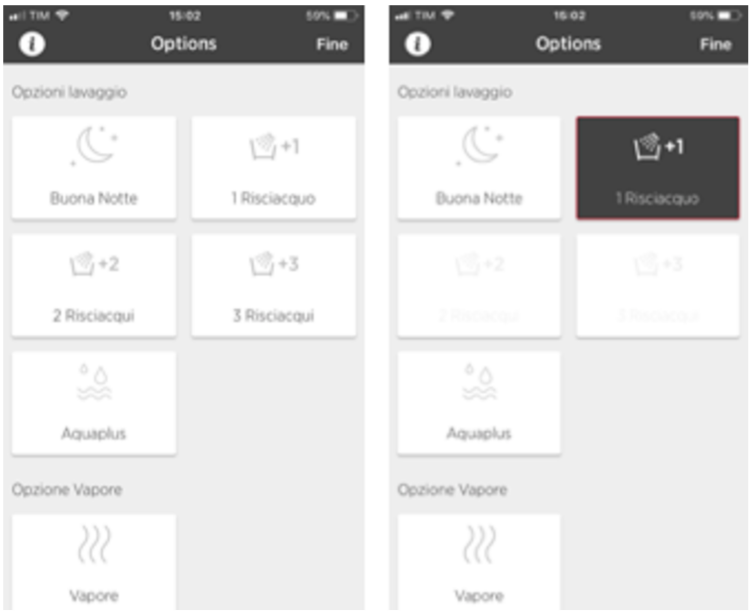


Fig. 84: The Hoover Wizard App: The Washer Options page



Fig. 85: The Hoover Wizard App: The delayed start setting page

Once the program has been set, it is possible to start the household appliance, by means of a special start command. ***Fig. 86*** shows the washer program start page.

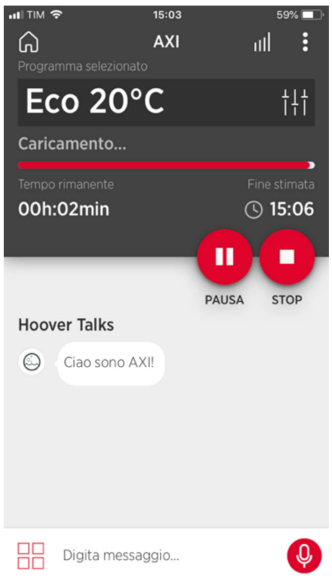


Fig. 86: The Hoover Wixard App: The Washer Program start Page

6.2.2 Architecture AS-IS

The current system architecture is built on three different modules: the mobile application, the household appliances and the Cloud (Server-DB) (***Fig. 87***).
The user-household appliance interaction takes place in two different ways: Direct Interaction (performed on the household appliance) or Mobile Interaction (performed on the mobile application). The application allows the user-household appliance communication both in the home (through Local Network) and outside (through Remote-Control). In the first case, queries are sent directly to the household appliances and saved on the Cloud DB (View HAs status and HAs Programming).

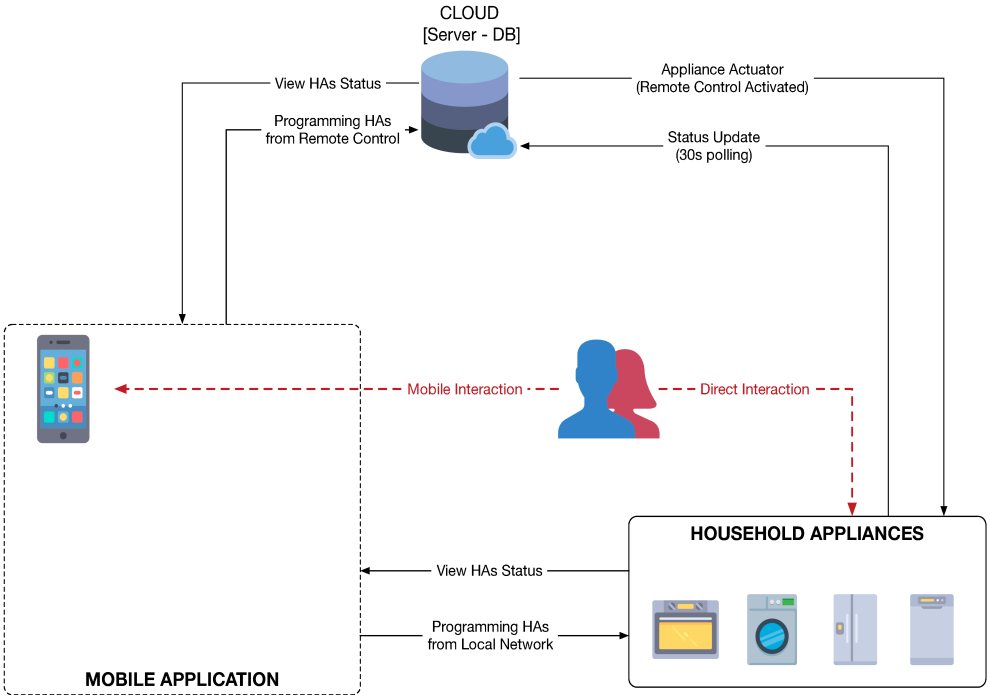


Fig. 87. The AS-IS Architecture

In the case of Remote-Control use, however, the queries are sent to the Household Appliances through the Server on Cloud. In detail, the HAs programming queries will be sent through the Appliance Actuator in Active Remote-Control mode, while the status will be displayed on the Cloud Database. The update status on the Database takes place every 30s (polling).

On the basis of the defined adaptation goals, the constrained components (CC) and editable components (EC) have been identified. Specifically, the defined CC are: User-Household appliances communication modality, Home-appliance Program Temperature and Cycle, Mobile Application Interface color and font. Instead, the defined EC are: Database, Mobile Application Interface contrast and layout, mobile application browsing mode, adaptive functions addition (i.e., programs or temperature suggestion).

6.2.3 Database

The Database aims to collect all the data deriving from the different system inputs, such as both the mobile and the appliance. In details, the information is structured in different semantic areas. In this thesis project we used the following eight tables:

Table 17. *Database Description*

ID	TABLE NAME	DESCRIPTION
1	Appliance_Activities	Activities performed by appliances. An activity (for example, a washer cycle) is characterized by the status of the appliance at the start and the end of the activity, by the program_id, and by the date (day,week,month). Saved activities are: oven cooking, washing/ washing cycles, dishwasher washing cycles.
2	Appliance_models	Appliance model (identifier = first 8 characters of the 16-digit code)
3	Appliance_status_changes	It contains all the states sent by the appliances that are considered significant by the system (for example: start and end of a wash cycle). Data historicization table.
4	Appliance_types	Type of appliance (washing machine, dishwasher, oven, ...)
5	Programs	Programs of all appliances.
6	Users	Registered users
7	Appliances	Contains all the information regarding the enrolled appliances. In particular, only the Appliances currently enrolled. If an appliance is "misplaced", the system deletes the record from this table and moves it to the "trash" = "versions"
8	Client_devices	List of all the devices used by users (tablet/smarthopne, operating system, version, ..)

For reasons of corporate confidentiality, it is not possible to show the structure of the Database in detail.

6.3. Adaptation Goals and Action Definition

Once the current system has been described in terms of architecture, database and mobile application, the final adaptation goal and actions have been defined. The final

goal of the project is to make the adaptive system based on user behavior and needs. To achieve this milestone, the general objectives and the specific adaptation actions have been defined *[3.4. Interaction Model]*.

Table 18. *Adaptation Goals and Action Definition*

Current UI	Adaptation goal	UI Adaptation Action
UI contrast and zoom: fixed	Increase UI accessibility and readability	Adapt UI contrast and zoom at the User Login
-	Improve user experience	Add/replace image with text to help user in programming the device
Single page layout	Optimize the layout (to highlight relevant information and minimize irrelevant and potentially distracting information)	Change UI layout at the User Login (i.e. propose the most fitting option for the user to provide an easy interaction and usage)
No Program suggestion	Best Program Suggestion	Depending on the user login, day and time range, the adaptive network suggests the best Program to the end user. The application will set as the first choice the Program suggested by the network, as a second choice the Program suggested by the Company and in cascade all the others.
No Temperature suggestion	Best Temperature Suggestion	Depending on the user login, day, time range and suggested Program, the adaptive network highlights the best Temperature to the end user. The application will set as the first choice the Temperature suggested by the network, as a second choice the Temperature suggested by the Company and in cascade all the others.
No Delay Start suggestion	Delay Start and cooking duration Suggestion	Depending on the user login, day, time range and suggested Program, the adaptive network suggest the delay start and cooking duration (oven case). The application will set as the first choice the most used delay start time and cooking duration.

In detail, the identified adaptation goals aim to increase the accessibility and readability of the mobile application, improve the user experience, optimize the layout and suggest the household-appliance functions (e.g. temperature, program, delay start etc.,) to the user based on the history of previous interactions. For each identified objective a list of adaptation actions to be implemented has been defined in
***Table** 18*.

6.4. Architecture Re-Design

Starting from the current Hoover system architecture (***Fig. 87***) and from the proposed architecture [*Chapter 4. Architecture]* and based on the defined requirements, the new Hoover architecture has been re-designed.
Specifically, the Adaptive engine has been added to the AS-IS Architecture and consists of two different modules:

- **Back-end Translation for Bayesian Network.** This module has been added as responsible for reading the existing Hoover Database, its filtering and its translation. Specifically, the filtering action selects from the DB the states needed to perform the adaptation. Instead, the translation action consists of translating the states, previously filtered, into a compatible language to be sent to the Adaptive Engine module.

- **Adaptive Engine Design**
  1. **Adaptive Middleware.** This second module is characterized by the Decision-Making Algorithm (DM). The adaptive middleware is responsible for Bayesian network learning; through this module data received from the Back-End Translation are processed and sent to the network. Once trained the network and obtained the specific conditional probabilities, these will be elaborated through the Decision-Making algorithm. The DM will identify the appropriate adaptation actions to be taken on the system by upgrading network node dependencies on the mobile application (*4.3.2.2 Decision Making Algorithm (DMA)*).
  2. **Mechanism Inference,** this module consists of the Bayesian Network customized for the System. The nodes number constituting the

network is affected by the devices number involved and by the user's monitoring actions. The network learning is made via the adaptive middleware.

Finally, the system architecture has been re-designed. In details, the Adaptive module introduction has been carried out on two different levels: the first level involves the DB Server on the Cloud, the second level involves the mobile application. The adaptation actions suggested by the adaptive module will be shown on the mobile device through the user interface.

Fig. 88 shows the adaptive module addition in the system architecture.

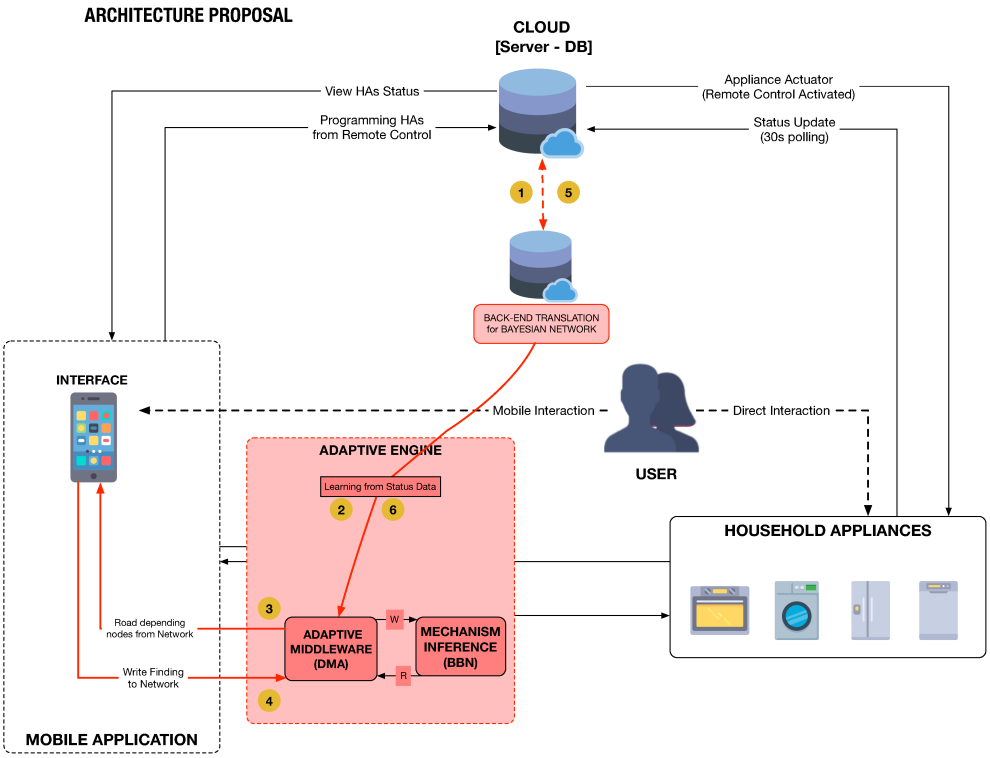


Fig. 88. The Architecture re-design (in red the Adaptive Module addition)

6.5. Back-end Translation Definition

Starting from the current Database and based on the entry requirements of the BBNs and on the defined adaptation goals, a Database upgrade has been made. Specifically, starting from the three main databases: User Total, Appliance Total and Status Changes, three different databases have been re-built for:
- Oven;
- Washing machine;
- Dishwasher.

Specifically, the structure of the three different databases consists of **5 phases:**

1. ***Initialization***: is an initial pre-processing phase in which the data are prepared to be cleaned and manipulated in the subsequent phases. In particular, at this stage the following actions are carried out:
    - Construction of a list of models of the electrical appliance associated with user information;
    - Elimination of duplicate information;
    - Filtering the Status_Changes through appliance information associated with users;
    - User Database Construction divided by household appliance.
2. ***Cleaning****:* this phase consists of a cleaning phase in which the no valid rows are deleted and in which the main parameters are cleaned.
3. ***Partition:*** this phase consists of partitioning columns useful for network training.
4. ***Manipulation:*** this phase consists of the manipulation of dates and times in Range for a better network training.
5. ***Hoover Filter:*** this phase consists in filtering only **Hoover** appliances.

In this way the communication and the data exchange between the Hoover database and the network was made compliant.

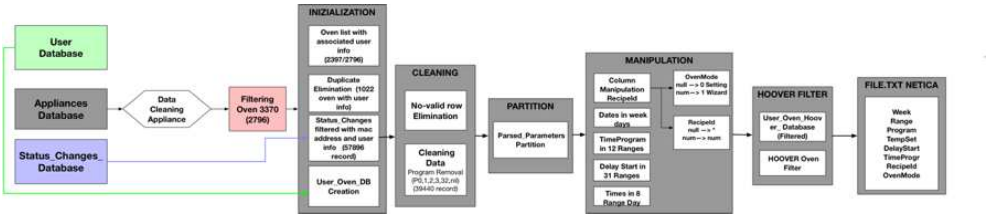


Fig. 89. Cleaning and Filtering Process of the Oven Database

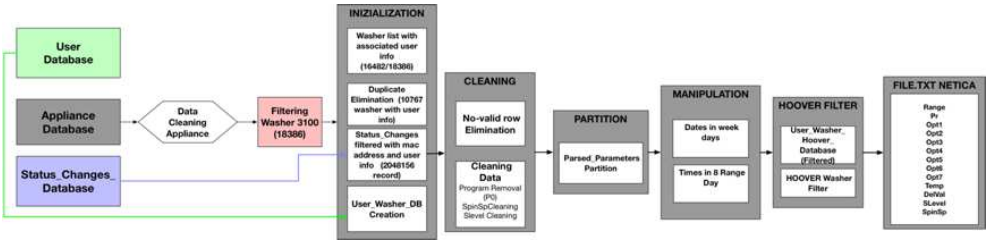


Fig. 90. Cleaning and Filtering Process of the Washer Database

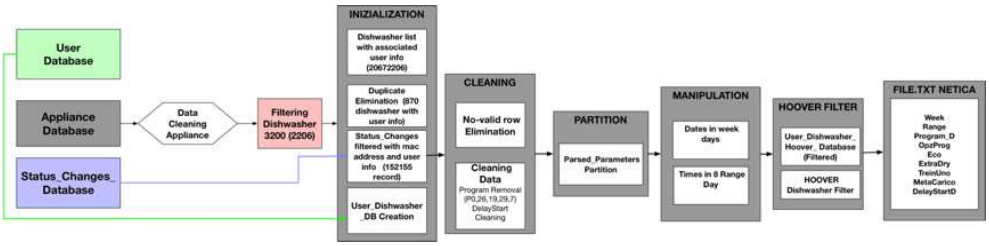


Fig. 91. Cleaning and Filtering Process of the Dishwasher Database

6.6. Adaptive Engine Design

Subsequently, the Bayesian network has been designed by assuming functioning of the Hoover mobile Application in a home environment, featuring three household appliances: oven, dishwasher and washer. In order to test the network functioning for each considered appliances, three different networks have been developed.
In the case of the **Oven**, the network was built on the basis of the possible actions that can be performed on the oven and suggested by the mobile application.
In detail the network, consisting of a total of 8 knots (***Fig. 92***), is structured as follow:

1. **Week**: Consisting of seven days of the week (Monday, Tuesday, Thursday, Friday, Saturday, Sunday);
2. **Range**: Consisting of time slots of the day (Eight three-hour Ranges maximum 1 day);

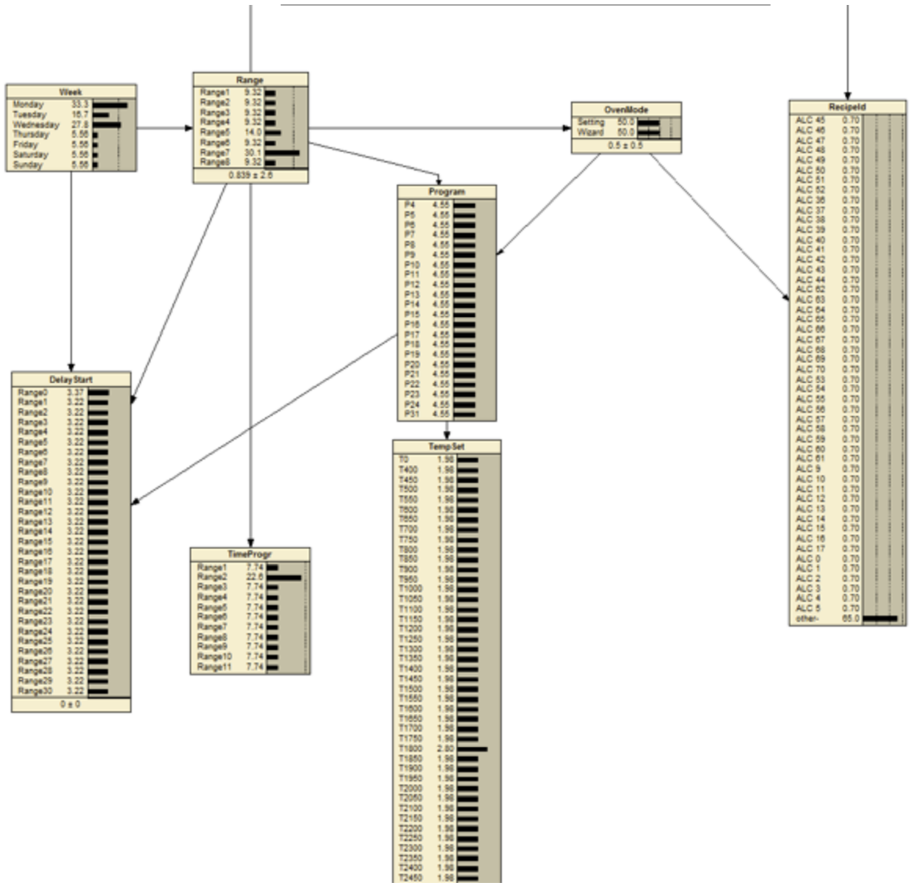


Fig. 92. The Oven Bayesian Network

3. **OvenMode**: Consisting of the GUI oven menu display modes (Setting or Wizrad mode);
4. **Program**: Consisting of the oven programs list (Static, Grill, Fan, Pyro, Eco, Defrost etc.,);
5. **RecipeId**: Consisting of the oven recipe list (First_course, vegetables, meat_and_main_course, fish, bread_and_cakes);

6. **TempSet**: Consisting of the oven temperature list (40°-280°);
7. **TimeProgr**: Consisting of the oven cooking duration and composed of eleven-time slots (Eleven half-hour Ranges, maximum six hours);
8. **DelayStart**: Consisting of the time of the day on which the oven delayed start is set (Thirty-ones half-hour Ranges, maximum fifteen hours).

In the same way the nets for the washing machine and the dishwasher have been build; in both cases the Week and Range nodes remain the same.
In particular, in the case of **Dishwasher**, the network consists of a total of 9 knots (***Fig. 93***) and is structured as follows:

1. **Program**: Consisting of the dishwasher programs list (Eco, Intensive, Quick, Night, Daily Car etc.,);
2. **OpzProg**: Consisting of the dishwasher options programs (0= o setting water, p= + setting water, m= - setting water);
3. **ExtraDry Function:** Consisting of the dishwasher options (0-1);
4. **TreInUno Function:** Consisting of the dishwasher options (0-1);
5. **MetaCarico Function:** Consisting of the dishwasher options (0-1);
6. **Eco Function:** Consisting of the dishwasher options (0-1);
7. **DelayStart:** Consisting of the time of the day on which the dishwasher delayed start is set (Twenty-three hour Ranges, maximum one day).

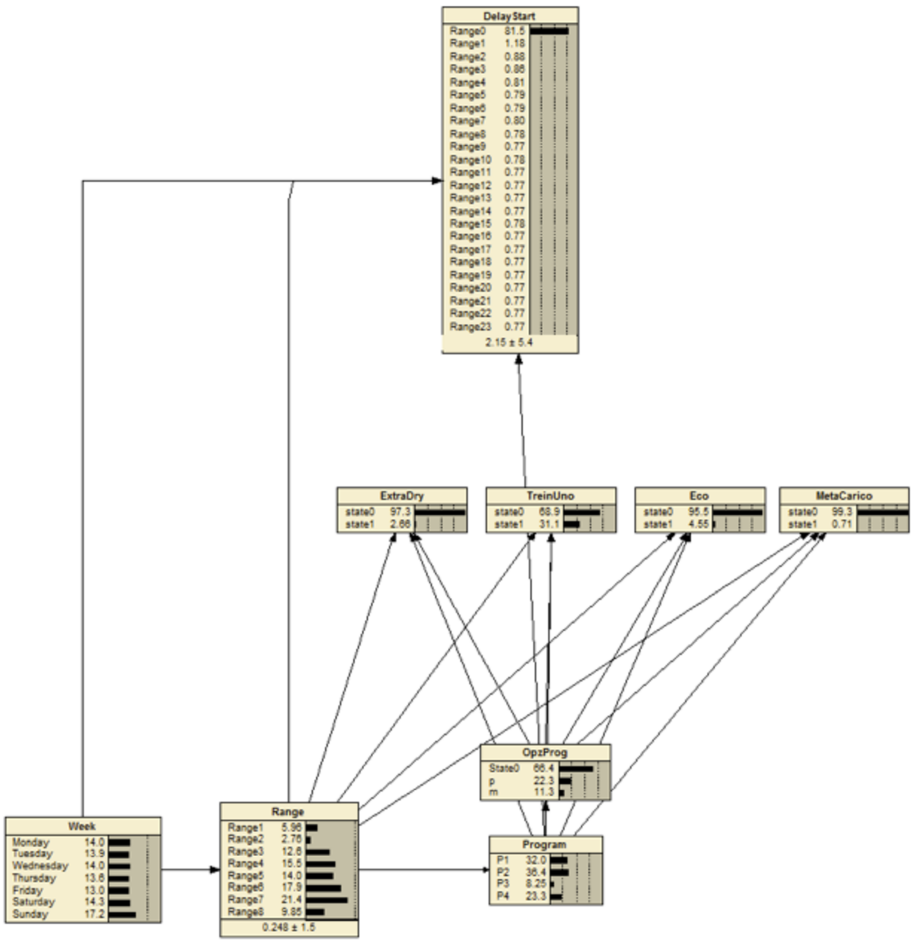


Fig. 93. The Dishwasher Bayesian Network

Finally, in the case of **Washer**, the network consists of a total of 14 knots (*Fig.* ***94***) and is structured as follows:

1. **Pr**: Consisting of the washer programs list (Perfect 20°, Perfect Rapid, Intensive, Baby, Rapid, Delicate, etc.,);
2. **SLevel**: Consisting of the soil level of the clothes (0= no soil, 1=min soil, 2=med soil, 3=max soil);
3. **SpinSp**: Consisting of the washer spin speed (0=no spin, 4=400 rpm, 5=500 rpm, .. , 16=1600 rpm, 255 = spin speed not applicable);

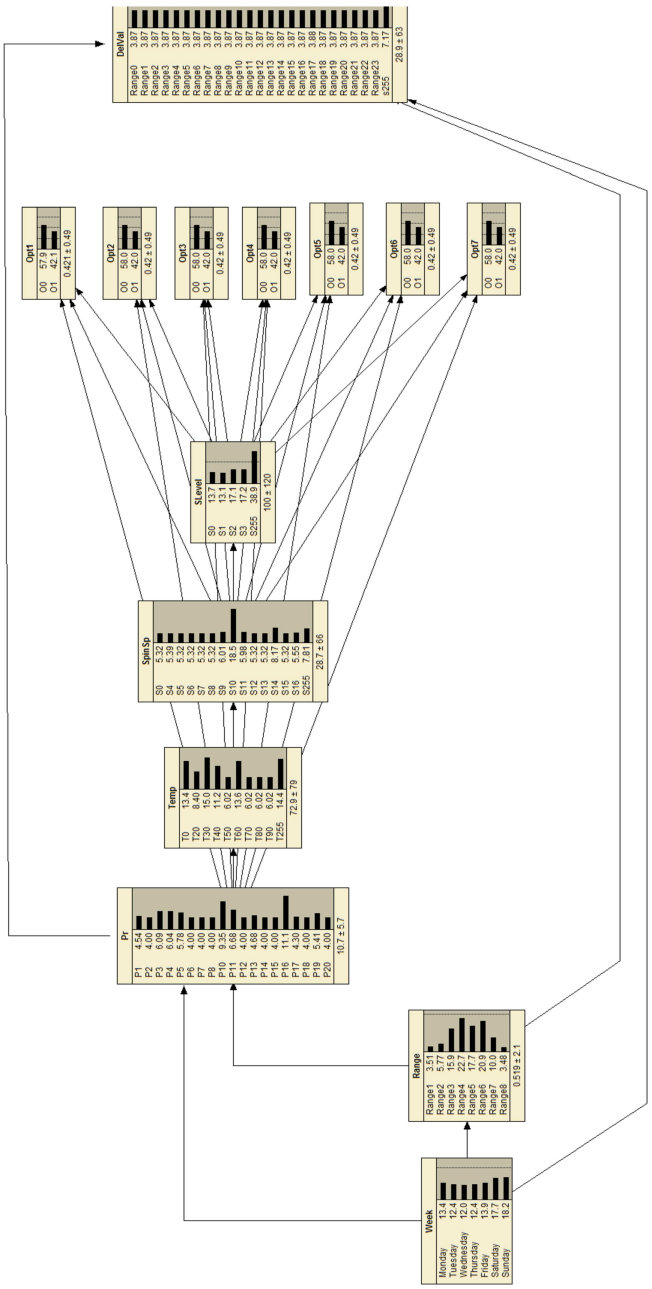


Fig. 94. The Washer Bayesian Network

4. **Temp**: Consisting of the washer temperature list (0-90°, 255= temperature not applicable);
5. **Opt1-Op7:** Consisting of the washer options (0-1);
6. **DelVal:** Consisting of the time of the day on which the washer delayed start is set (Twenty-three hour Ranges, maximum one day, b255= not applicable).

6.7. Cluster Definition

In order to support the design phase, a reliable and realistic representation of target user groups for reference has been created. It is based on the ethnographic analysis. It allowed giving a clear picture of the users in terms of needs and expectations, from their backgrounds.
In particular, the data present in the *User* database were analyzed. Users have been classified by Country, Brand, Type of appliance (Oven, Washing Machine and Dishwasher) and Model (identification code). For business reasons, England and Italy were chosen as survey countries; all others have been identified as "other".
This classification made it possible to associate a cluster of users to each appliance model. Each cluster consists of a subset of events (states) related to user behavior: through the user_id it is possible to select in the status_changes database the behaviors related to the identified cluster.
In this way, a total of 37 clusters were created: 10 representatives of the oven, 10 of the dishwashers and 17 for the washing machine.
In the specific case of the oven, five models were selected, as shown in the Table below.

Table 19: *Oven Cluster*

Brand	Country	Device	Model	Cluster
Hoover	**GB**	**Oven**	33701511	**1° Cluster_Oven**
Hoover	**GB**	**Oven**	33701681	**2° Cluster_Oven**
Hoover	**GB**	**Oven**	33701751	**3° Cluster_Oven**
Hoover	**GB**	**Oven**	33701775	**4° Cluster_Oven**
Hoover	**GB**	**Oven**	33701860	**5° Cluster_Oven**
Hoover	**IT**	**Oven**	33701511	**6° Cluster_Oven**
Hoover	**IT**	**Oven**	33701681	**7° Cluster_Oven**
Hoover	**IT**	**Oven**	33701751	**8° Cluster_Oven**
Hoover	**OTHER**	**Oven**	33701511	**9° Cluster_Oven**
Hoover	**OTHER**	**Oven**	33701751	**10° Cluster_Oven**

In the specific case of the dishwasher, six models were selected, as shown in the *: Table 20*:

Table 20: *Dishwasher Cluster*

Brand	Country	Device	Model	Cluster
Hoover	**GB**	Dishwasher	32000956	**1° Cluster_Dishwasher**
Hoover	**GB**	Dishwasher	32000957	**2° Cluster_Dishwasher**
Hoover	**IT**	Dishwasher	32000946	**3° Cluster_Dishwasher**
Hoover	**IT**	Dishwasher	32000945	**4° Cluster_Dishwasher**
Hoover	**OTHER**	Dishwasher	32000945	**5° Cluster_Dishwasher**
Hoover	**OTHER**	Dishwasher	32000946	**6° Cluster_Dishwasher**
Hoover	**OTHER**	Dishwasher	32000956	**7° Cluster_Dishwasher**
Hoover	**OTHER**	Dishwasher	32000957	**8° Cluster_Dishwasher**
Hoover	**OTHER**	Dishwasher	32001111	**9° Cluster_Dishwasher**
Hoover	**OTHER**	Dishwasher	32001112	**10° Cluster_Dishwasher**

In the case of the washing machine, the total clusters are seventeen. In detail, washing machines differ from each other for load capacity, which is why they have been in turn clasterized:

Table 21: *Washer Cluster*

Cluster	Kg
Cluster A	7
Cluster B	8
Cluster C	9
Cluster D	10/11
Cluster E	12
Cluster F	13

Brand	Country	Device	Model	Cluster
Hoover	GB	Washer	CLUSTER B	1° Cluster_Washer
Hoover	GB	Washer	CLUSTER C	2° Cluster_Washer
Hoover	GB	Washer	CLUSTER D	3° Cluster_Washer
Hoover	GB	Washer	CLUSTER E	4° Cluster_Washer
Hoover	GB	Washer	CLUSTER F	5° Cluster_Washer
Hoover	IT	Washer	CLUSTER A	6° Cluster_Washer
Hoover	IT	Washer	CLUSTER B	7° Cluster_Washer
Hoover	IT	Washer	CLUSTER C	8° Cluster_Washer
Hoover	IT	Washer	CLUSTER D	9° Cluster_Washer
Hoover	IT	Washer	CLUSTER E	10° Cluster_Washer
Hoover	IT	Washer	CLUSTER F	11° Cluster_Washer
Hoover	OTHER	Washer	CLUSTER A	12° Cluster_Washer
Hoover	OTHER	Washer	CLUSTER B	13° Cluster_Washer
Hoover	OTHER	Washer	CLUSTER C	14° Cluster_Washer
Hoover	OTHER	Washer	CLUSTER D	15° Cluster_Washer
Hoover	OTHER	Washer	CLUSTER E	16° Cluster_Washer
Hoover	OTHER	Washer	CLUSTER F	17° Cluster_Washer

The construction of specific clusters based on real usage data allows the consumer using the system for the first time to access an already initialized system characterized by data that reflect the behavioral trend of a similar population (in common with the type and model of home appliance and the country of belonging). Each cluster is characterized by 200 records.
The relatively low number of records associated with each class depends on whether the system is designed to provide an initial learning phase to the customer, in which the user's usage, behavior and preferences are learned. During this initial learning phase, the system provides suggestions based on the behavior of the associated cluster. The learning phase ends when the number of records associated with the new user is greater than 200: for each new user record a cluster record is deleted (ratio of 1:1). Once a record number associated with the user (> 200) has been reached, the system definitively bypasses the cluster, adapting to the user's behavior.

6.8. DM Algorithm Test & Learning Network

To verify the functioning of the new adaptive system, simulation tests have been carried out to test the adaptation of the three different Bayesian Network. The Decision Making Algorithm (*4.3.2.2 Decision Making Algorithm (DMA)*) was tested with two different threshold values: 20% and 35%. A total of 37 networks were tested (10 for oven and dishwasher, 17 for the washing machine) for both threshold values.

By way of example, in the case of the oven, the behavior of the 4th Cluster_Washer has been simulated for both thresholds (***Fig. 95***).

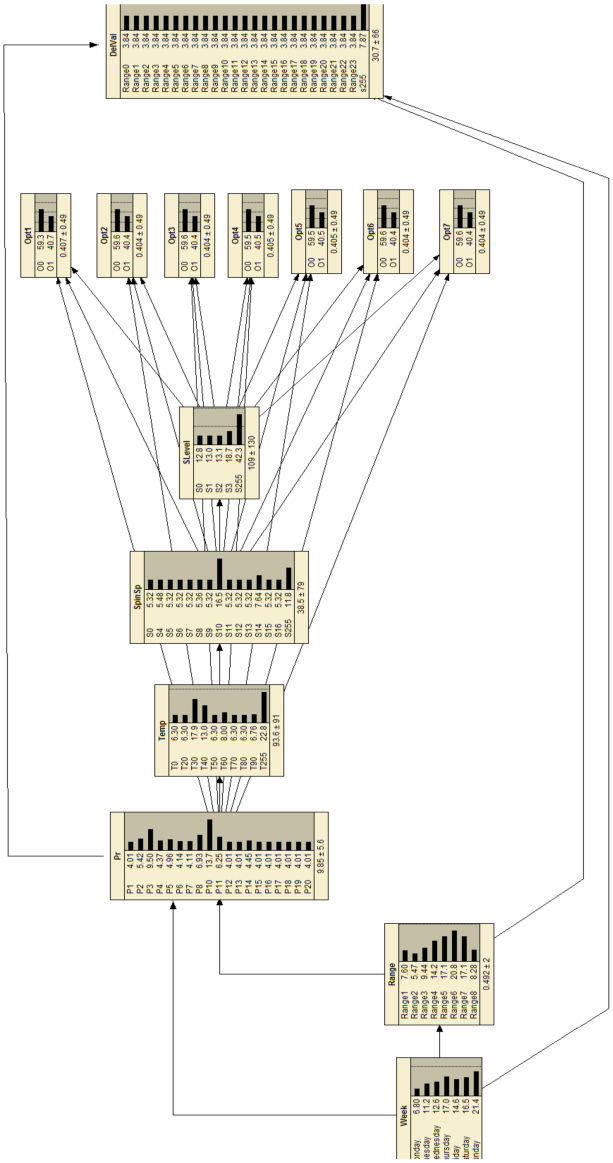


Fig. 95: 4°Cluster_Washer Bayesian Networ

Using the 20% threshold, all nodes satisfy the adaptation condition. (***Fig. 96***).

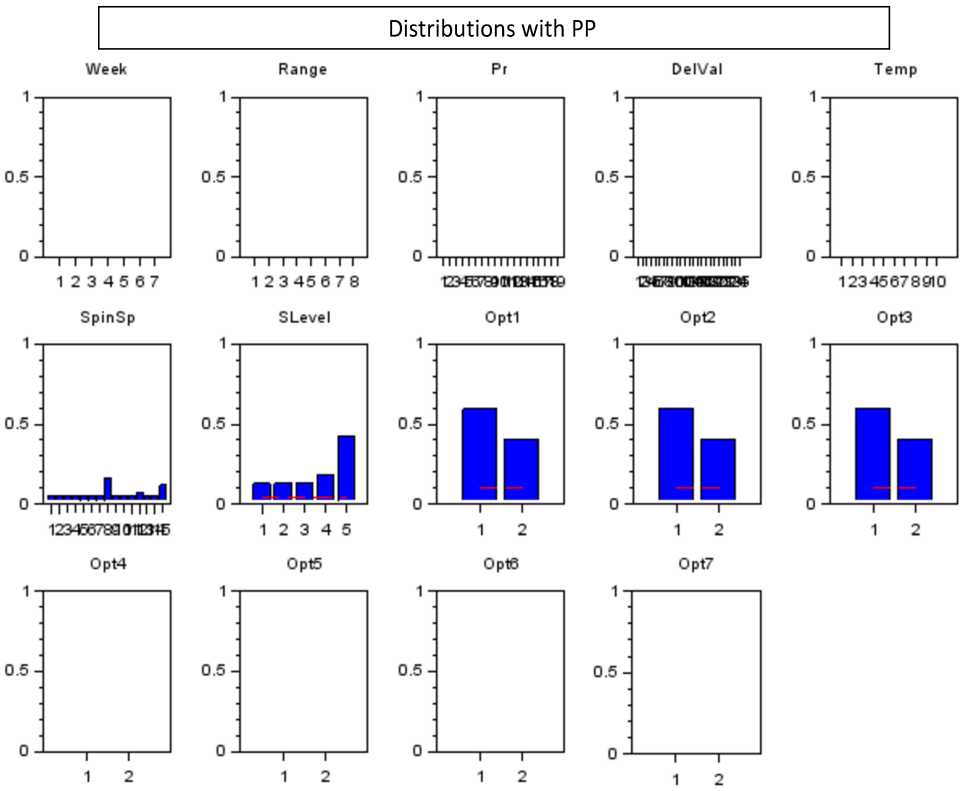


Fig. 96. Algorithm Test for the 4°Cluster_Washer with a threshold of 20%

By raising the threshold to 35%, it can be seen that the algorithm applies a restrictive action on the nodes. In fact, the condition ***d> thr*** is not satisfied for the nodes Week and Range. The remaining nodes satisfy the condition (***Fig. 97***).

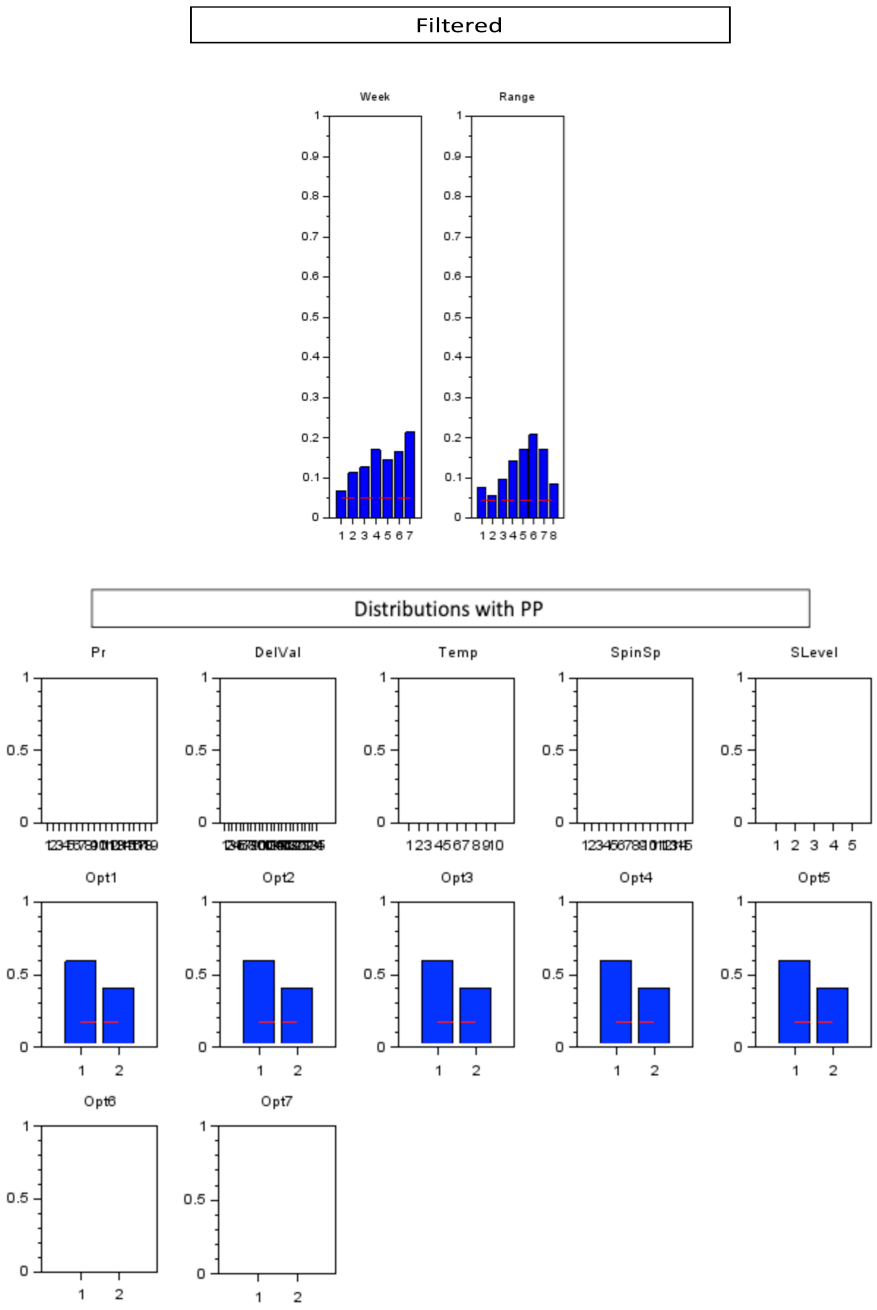


Fig. 97. Algorithm Test for the 4°Cluster_Washer with a threshold of 35%: at the top the filtered nodes; at the bottom the nodes that satisfy the condition

The table below shows in detail the behavior related to the nodes belonging to the seventeen clusters of the washer, simulated for both thresholds. Nodes that do not meet the adaptation condition have been highlighted in orange, depending on the identified problem (**d<thr or no! Pmax**).

Table 22: *Simulation Thresholds for the washer*

thr_perc=20	Week	Range	Pr	Opt1	Opt2	Opt3	Opt4	Opt5	Opt6	Opt7	Temp	DelVal	SLevel	SpinSp
1°Cluster_washer	x	x	x	x	x	x	x	x	x	x	x	x	x	x
2°Cluster_washer	x	x	x	x	x	x	x	x	x	x	x	x	x	x
3°Cluster_washer	x	x	x	x	x	x	x	x	x	x	x	x	x	x
4°Cluster_washer	x	x	x	x	x	x	x	x	x	x	d<thr	x	x	x
5°Cluster_washer	x	x	x	x	x	x	x	x	x	x	x	x	x	x
6°Cluster_washer	x	x	d<thr	x	x	x	x	x	x	x	x	x	x	x
7°Cluster_washer	x	x	x	x	x	x	x	x	x	x	x	x	x	x
8°Cluster_washer	x	x	d<thr	x	x	x	x	x	x	x	x	x	x	x
9°Cluster_washer	x	x	x	x	x	x	x	x	x	x	x	x	x	x
10°Cluster_washer	x	x	x	x	x	x	x	x	x	x	x	x	x	x
11°Cluster_washer	x	x	x	x	x	x	x	x	x	x	x	x	x	x
12°Cluster_washer	x	x	x	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	x
13°Cluster_washer	x	x	x	x	x	x	x	x	x	x	x	x	x	x
14°Cluster_washer	x	x	x	x	x	x	x	x	x	x	x	x	x	x
15°Cluster_washer	x	x	x	x	x	x	x	x	x	x	x	x	x	x
16°Cluster_washer	x	x	x	x	x	x	x	x	x	x	x	x	x	x
17°Cluster_washer	x	x	x	x	x	x	x	x	x	x	x	x	x	x

thr_perc=35	Week	Range	Pr	Opt1	Opt2	Opt3	Opt4	Opt5	Opt6	Opt7	Temp	DelVal	SLevel	SpinSp
1°Cluster_washer	d<thr	d<thr	x	x	x	x	x	x	x	x	x	x	x	d<thr
2°Cluster_washer	d<thr	d<thr	x	x	x	x	x	x	x	x	x	x	x	x
3°Cluster_washer	d<thr	d<thr	x	x	x	x	x	x	x	x	x	x	x	x
4°Cluster_washer	d<thr	d<thr	x	x	x	x	x	x	x	x	d<thr	x	x	x
5°Cluster_washer	d<thr	d<thr	x	x	x	x	x	x	x	x	x	x	x	d<thr
6°Cluster_washer	d<thr	d<thr	x	x	x	x	x	x	x	x	x	x	x	x
7°Cluster_washer	d<thr	d<thr	x	x	x	x	x	x	x	x	x	x	x	x
8°Cluster_washer	d<thr	d<thr	x	x	x	x	x	x	x	x	x	x	x	x
9°Cluster_washer	x	x	x	x	x	x	x	x	x	x	x	x	x	x
10°Cluster_washer	d<thr	d<thr	x	x	x	x	x	x	x	x	x	x	x	x
11°Cluster_washer	d<thr	d<thr	x	x	x	x	x	x	x	x	x	x	x	x
12°Cluster_washer	x	d<thr	x	d<thr	d<thr	d<thr	d<thr	d<thr	x	x	d<thr	d<thr	x	x
13°Cluster_washer	d<thr	d<thr	x	x	x	x	x	x	x	x	x	x	x	x
14°Cluster_washer	x	d<thr	x	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	x	x	x
15°Cluster_washer	d<thr	d<thr	x	x	x	x	x	x	x	x	x	x	x	x
16°Cluster_washer	d<thr	d<thr	x	x	x	x	x	x	x	x	x	x	x	x
17°Cluster_washer	d<thr	d<thr	x	x	x	x	x	x	x	x	x	x	x	x

Therefore, the chosen threshold represents a restriction value to be attributed to the algorithm and enable or not adaptation. This allows the company to calibrate the adaptation according to the final goal: for a less adaptive network, higher thresholds

should be chosen, on the contrary for a more adaptive and dynamic network, lower thresholds will have to be chosen.
A further advantage can be associated with the possibility of choosing different thresholds associated with different appliances. Thus, allowing to have dynamic networks associated with the most used appliances, and more static networks for less frequently used appliances.
For completeness, the tables related to the simulations carried out in the case of the oven and the dishwasher are shown below.

Table 23: *Simulation Thresholds for the oven*

thr_perc=20	Week	Range	Program	TempSet	DelayStart	RecipeId	TimeProgr	OvenMode
1°Cluster_Oven	x	x	x	x	x	d<thr	No ! Pmax	x
2°Cluster_Oven	x	x	x	x	x	d<thr	No ! Pmax	x
3°Cluster_Oven	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr
4°Cluster_Oven	x	x	x	x	x	d<thr	x	x
5°Cluster_Oven	x	x	x	x	x	x	d<thr	x
6°Cluster_Oven	x	x	x	x	x	d<thr	No ! Pmax	x
7°Cluster_Oven	x	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	x
8°Cluster_Oven	x	x	x	x	x	x	x	x
9°Cluster_Oven	x	x	x	x	x	x	No ! Pmax	x
10°Cluster_Oven	x	x	d<thr	d<thr	d<thr	d<thr	d<thr	x

thr_perc=35	Week	Range	Program	TempSet	DelayStart	RecipeId	TimeProgr	OvenMode
1°Cluster_Oven	x	x	x	x	x	d<thr	No ! Pmax	x
2°Cluster_Oven	x	x	x	x	x	d<thr	No ! Pmax	x
3°Cluster_Oven	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr
4°Cluster_Oven	x	d<thr	x	x	x	d<thr	x	x
5°Cluster_Oven	x	x	x	d<thr	d<thr	d<thr	d<thr	x
6°Cluster_Oven	d<thr	x	x	x	x	d<thr	No ! Pmax	x
7°Cluster_Oven	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	x
8°Cluster_Oven	x	x	x	x	d<thr	x	x	x
9°Cluster_Oven	d<thr	x	d<thr	x	d<thr	x	No ! Pmax	x
10°Cluster_Oven	x	d<thr	d<thr	d<thr	d<thr	d<thr	d<thr	x

Table 24: *Simulation Thresholds for the dishwasher*

thr_perc=20	Week	Range	Eco	OpzProg	Program	ExtraDry	TreinUno	DelayStart	MetaCarico
1°Cluster_Dishwasher	x	d<thr	x	x	x	x	x	x	x
2°Cluster_Dishwasher	d<thr	x	x	x	d<thr	x	x	x	x
3°Cluster_Dishwasher	d<thr	d<thr	x	x	x	x	x	x	x
4°Cluster_Dishwasher	d<thr	d<thr	x	x	d<thr	x	x	x	x
5°Cluster_Dishwasher	d<thr	d<thr	x	x	x	x	x	x	x
6°Cluster_Dishwasher	x	x	x	x	x	x	x	x	x
7°Cluster_Dishwasher	x	x	x	x	x	x	x	x	x
8°Cluster_Dishwasher	x	x	d<thr	d<thr	x	x	x	x	x
9°Cluster_Dishwasher	d<thr	d<thr	x	x	x	x	d<thr	x	x
10°Cluster_Dishwasher	x	x	d<thr	x	x	d<thr	d<thr	x	d<thr

thr_perc=35	Week	Range	Eco	OpzProg	Program	ExtraDry	TreinUno	DelayStart	MetaCarico
1°Cluster_Dishwasher	d<thr	d<thr	x	x	x	x	x	x	x
2°Cluster_Dishwasher	d<thr	x	x	x	d<thr	x	x	x	x
3°Cluster_Dishwasher	d<thr	d<thr	x	x	x	x	x	x	x
4°Cluster_Dishwasher	d<thr	d<thr	x	x	d<thr	x	x	x	x
5°Cluster_Dishwasher	d<thr	d<thr	x	x	x	x	x	x	x
6°Cluster_Dishwasher	x	x	x	x	d<thr	x	x	x	x
7°Cluster_Dishwasher	x	x	x	x	d<thr	x	x	x	x
8°Cluster_Dishwasher	x	x	d<thr	d<thr	x	x	d<thr	x	x
9°Cluster_Dishwasher	d<thr	d<thr	d<thr	x	x	x	d<thr	x	x
10°Cluster_Dishwasher	x	x	d<thr	x	x	d<thr	d<thr	d<thr	d<thr

6.9. UI Re-Design

In order to simplify the control and the setting of household appliances, the mobile application UI has been re-designed.

The UI re-designing consists in removing features that are not required by the user and adapting others to his preferences. Based on the user profile, the previously interaction and the monitored actions, the adaptive module introduction allows to find the optimal features combination for the interface.

The UI adaptations concern two main features: the first one consists of changing the interface's characteristics, and the last consists of changing the content displayed by the interface [*4.4. Interface*]. The first UI adaptations is based on varying the graphical features or its layout configuration. In this case, the platform is aware of the user experience with the interface and it makes the corresponding adaptation based on these interactions. The second UI adaptation deals with the interface

contents. Contents represent all interface items editable according to uses and actions the user acts on interface, own preferences and needs.

6.9.1 Adaptive Behavior

The new adaptive Hoover Wizard will provide personalized suggestions to the user. Specifically, the insertion of the adaptive component in the new system has introduced some changes with respect to the existing system. These are the following:
(1) Initial Program Suggestion;
(2) Initial Suggestion of the Program Setting;
(3) Dynamic Suggestion during the selection of the Program;
(4) Parameter update by Tree structure.
In detail, from the adaptive point of view, at the first access of the user, according to time and day of the week the most used program associated with the reference cluster will be suggested (1).
Subsequently:

- If the phone is in offline mode, the last settings entered will be proposed;
- If the phone is in online mode, an asynchronous call will be made with input Weekday and Range of use, for a prediction of the values of Program, Temperature, Spin Speed and options list (2);
- If the selected program has never been used, the program default settings will be selected (no additional options).

The suggestion provided by the system allows the user to start the machine faster; if instead the user wants to ignore the suggestion and start a different program, the system will provide a new program prediction to the end-user; in this last case, in fact, a new call will be made to the service with the following input parameters (inferences) Day, Range and Program, to have a prediction of the values of temperature and options list (3).
In detail, once the program has been chosen, the prediction system will modify all the fields after the selected one (4). At any time, the user could exit the system or cancel the choices made through the appropriate icon on the screen.
In conclusion, we can assume that:

1. Each time the user changes the program, the prediction system will be reset;
2. The system does not make predictions of the fields that are at the top of the modified field;

3. The system does not predict if the field has already been modified voluntarily by the user.

By way of example, the new Adaptive Hoover Wizard App and the Hoover Wizard App are compared in the case of the **washing machine**.
At the first log-in performed by the user we can see in ***Fig. 98*** the different behavior of the two Apps. In detail, the adaptive app, depending on the day and time slot, suggests the best program to be used; on the other hand, the normal app, shows the default program at the first access, Total Care, (1).

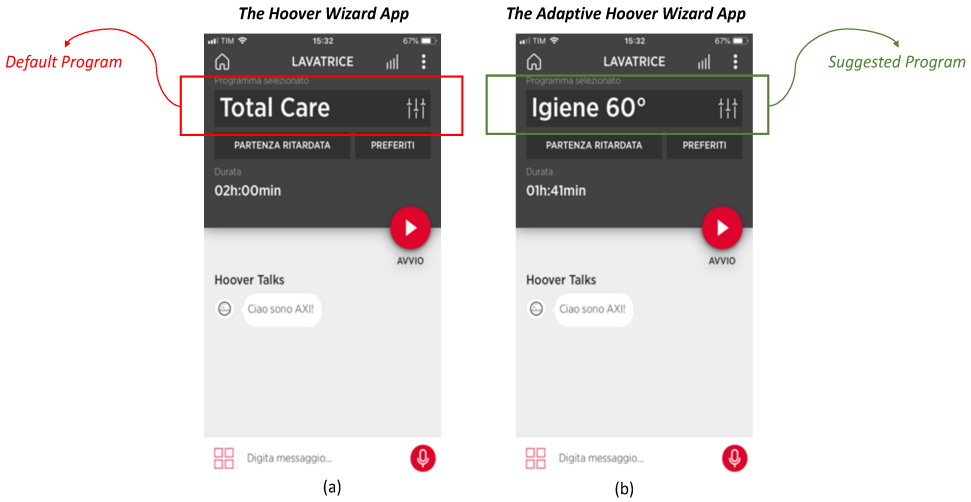


Fig. 98: The two App compared: (a) the initial page with the default program of the Hoover Wizard; (b) the initial page with the suggested program of Adaptive Hoover Wizrad.

In detail, by selecting the "Hygiene 60" program on both Apps, we note that in the normal version the default settings, recommended by the company, are highlighted. While in the adaptive version, based on user preferences, temperatures, speeds and options are suggested (2) ***Fig. 99***.

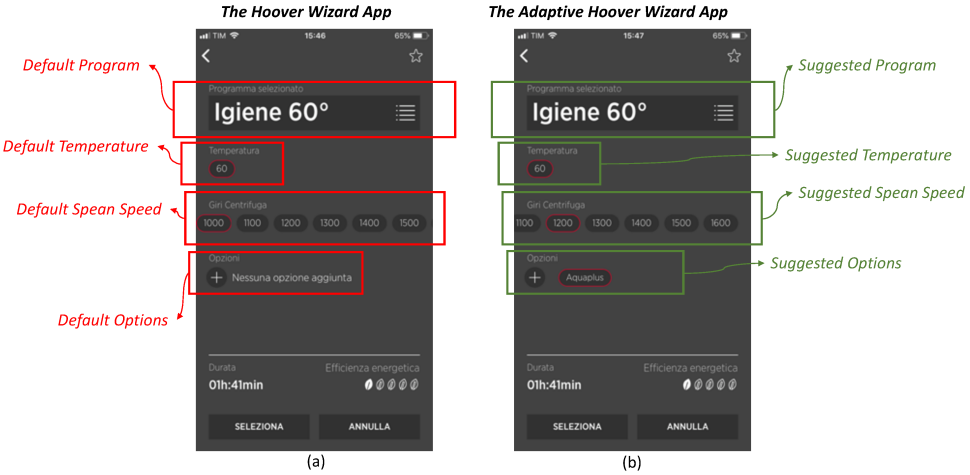


Fig. 99: The two App compared: on the left the program setting page of the Hoover Wizard, on the right the program setting page of the Adaptive Hoover Wizard.

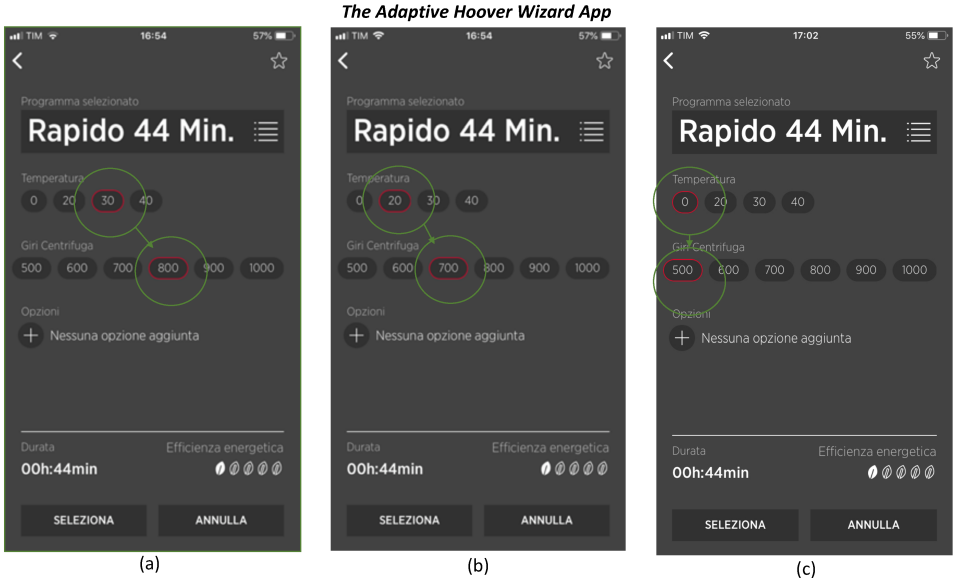


Fig. 100: The Adaptive Hoover Wizard App - Dynamic change of setting: (a) at the temperature of 30 ° a spin speed of 800 is associated; (b) at the temperature of 20 ° a spin speed of 700 is associated; (c) at the temperature of 0 ° a spin speed of 500 is associated.

If the user decides to change the program suggested by the adaptive version, selecting a different program from the program list, the system will make a new call

to the service having input parameters Day, Range and Program, to have prediction of temperature, spin speed and options list (3).
Furthermore:

- If the user selects the temperature, the spin speed and the options will be inferred (4) ***Fig. 100***;
- If the user selects the spin Speed, the temperature set as inference will be considered and the prediction will change only the option data (4) ***Fig. 100***;
- If the user selects the option data, the preferred option on that day will be suggested (4) ***Fig. 100***.

On the contrary in the normal version, the choice of a different program by the user does not influence the change in temperature, spin speed or options ***Fig. 101***.

The Hoover Wizard App



Fig. 101: The Hoover Wizard App: the change in the program does not affect the temperature or the spin speed. (a) at the temperature of 30 ° a spin speed of 1000 is associated;(b) at the temperature of 40 ° a spin speed of 1000 is associated;

Finally, if the user selects the delayed start, the preferred time for a specific day on the adaptive interface is displayed. While in the normal system mode, the current time will be highlighted.



Fig. 102: The two App compared. (a) In the Hoover Wizard App the timer starts at the current time; (b) in the Adaptive version the most used time of the delayed start is suggested.

The software architecture of the mobile application has been developed in collaboration with NTNext and Coodermine.

6.10. User Experimentation

A user's experimentation test was carried out to evaluate the system in terms of adaptivity and user satisfaction.
This evaluation aims to measure the adaptivity perceived by users during the first use of the adaptive system compared to the non-adaptive Wizard system. For business reasons, testing was conducted on washing machines.

6.10.1 Participants

For the purpose of the study, volunteer subjects who own the AXI washing machine were recruited. To be enrolled, the following inclusion criteria were fixed:

- Age 18 and older;
- Native speakers of the trial language;
- Ability to make time commitment;
- Previous experience in the use of Hoover Wizard Application.

The total sample was composed of 5 users, 2 males and 3 females, with a 47 years old average age (SD±3.76) (Nielsen and Landauer 1993).

6.10.2 Procedure

In detail, the users involved were asked to use the Adaptive Hoover Wizard Application for the management of household appliances in the home, for a period of about one month. In addition, the users have been interviewed by using a pre-test questionnaire designed to collect background information and to evaluate the current Hoover Wizard App.

The pre-test questionnaire is shown in the ***Table 25*****.**

Table 25: *The pre-test questionnaire*

QUESTIONS - 1° PART				
id	**SCORE (1)**	**SCORE**	**SCORE (5)**	
1	Complicated		Easy	
2	Unlikable		Pleasing	
3	Slow		Fast	
4	Conventional		Inventive	
5	Obstructive		Supportive	
6	Demotivating to use		Motivating to use	
7	Does not meet expectations		Meets exepectations	
8	Confusing		Clear	
9	Inefficient		Efficient	
10	Static (Default)		Dynamic (Custom)	
	QUESTIONS - 2° PART			**SCORE (1=little,…,5=much)**
1	How habitual you are in setting up your appliances			
2	How often did you use your favorite programs?			
3	How often did you use the program suggested by the application?			
4	How many times have you been satisfied with the setting suggested by the application (Temperature-Centrifuge-Liv dirty etc.)?			

At the end of the experiment the pre-test (post-test) questionnaire was resubmitted to them to evaluate the level of satisfaction and to test the adaptivity perceived by

the new system. In the post-experimentation phase, a question related to the improvement of the app was submitted to the users.
The post-test questionnaire is shown in ***Table 26***.

Table 26: *The Post-test questionnaires*

QUESTIONS - 1° PART			
id	**SCORE (1)**	**SCORE**	**SCORE (5)**
1	Complicated		Easy
2	Unlikable		Pleasing
3	Slow		Fast
4	Conventional		Inventive
5	Obstructive		Supportive
6	Demotivating to use		Motivating to use
7	Does not meet expectations		Meets exepectations
8	Confusing		Clear
9	Inefficient		Efficient
10	Static (Default)		Dynamic (Custom)

	QUESTIONS - 2° PART	SCORE (1=little,...,5=much)
1	How habitual you are in setting up your appliances	
2	How often did you use your favorite programs?	
3	How often did you use the program suggested by the application?	
4	How many times have you been satisfied with the setting suggested by the application (Temperature-Centrifuge-Liv dirty etc.)?	
5	How has the application improved with this adaptive update?	

The questionnaires were also formulated to investigate aspects related to the usability, the design and the quality of the experience of use with the product and its interface. In total, the questionnaires consist of several items for which participants were asked to mark their level of agreement on a 5-point Likert scale.
To accelerate the learning phase during the test period, it was decided to eliminate the associated cluster records with a ratio of 1:10 (instead of 1:1, ***6.7. Cluster Definition***). For each new user record, 10 cluster records are deleted. In this way we have accelerated the learning phase of the system, providing a responsive system to the end-user.

6.10.3 Results

This section shows the results of the evaluation of the two Hoover Wizard Application. Two evaluation processes have been performed: the first one is related to the objective evaluation by analyzing the data coming from the database of each

involved user and the second one is a subjective evaluation obtained through ad-hoc questionnaires submitted to end users.
The main goal is to demonstrate the adaptability and versatility of the proposed application by evaluating the user satisfaction and the perceived adaptability with respect to the native application.
From the objective point of view, for each user the data present in databases were analyzed and the records relating to the interactions with the application were interpreted.
Based on the information coming from user-app interactions, we analyzed the deviations between the two applications in terms of:

- Number of times the suggested **Program** has been changed;
- Number of times the **Temperature** has been changed;
- Number of times the **Spin Sped** has been changed;
- Number of times the **Level of dirt** (Slivel) has been changed;
- Number of times the **Options** have been changed.

The results obtained through the objective evaluation are shown in ***Fig. 103***.
Their comparison shows the achievement of very different score rankings for each section. In particular, the maximum deviation is about 60% and mainly refers to the number of times that the program has been changed. In fact, analyzing the achieved scores related to the **Program**, it is possible to affirm that all the users involved have drastically reduced the number of program changes, thanks to the personalized suggestion provided by the adaptive system (with an average of about 20% compared to 80% resulting from the use of the non-adaptive app). From a **Temperature** point of view, users choose the suggestion provided by the app by reducing the average percentage from about 46% to 15%. **Spin speed** suggestions also met user needs, which reduce the change number average percentage from 33% to 8%. However, smaller deviations between the two apps were recorded regarding the options and the level of dirt. In general, we can affirm that most of the users involved in the study do not use this type of settings in the wash cycle. This negatively affects the behavior provided by the adaptive system, which has no information on user interactions and does not affect the system actions.

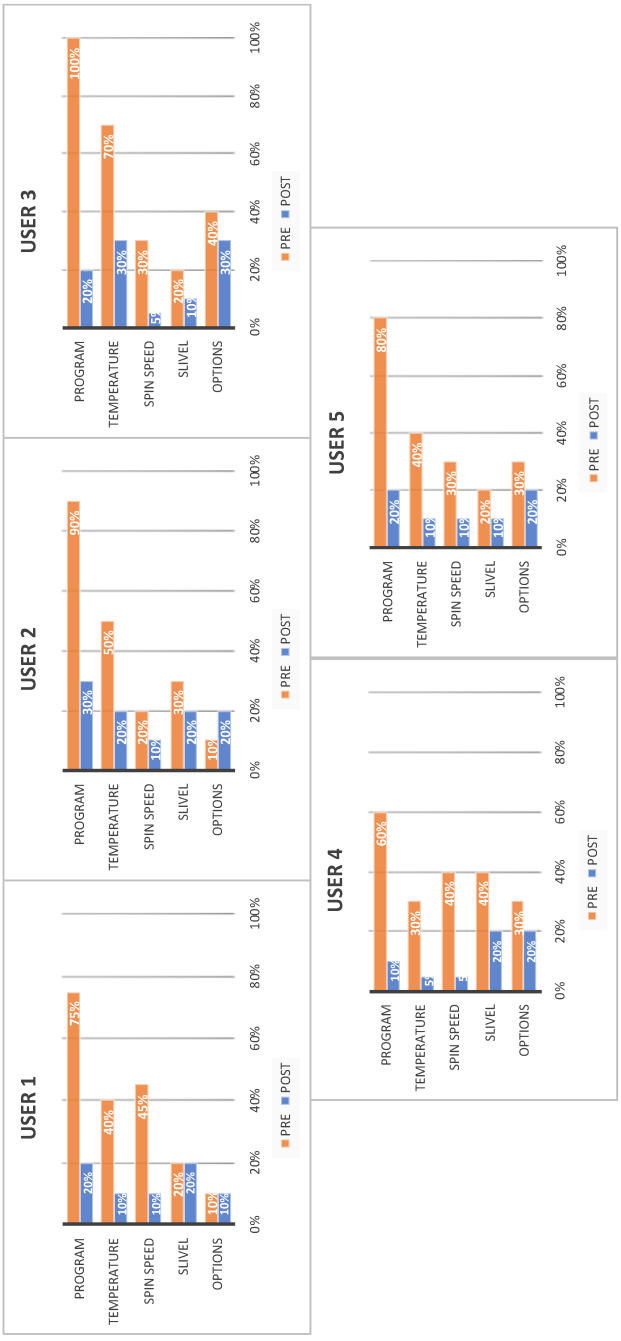
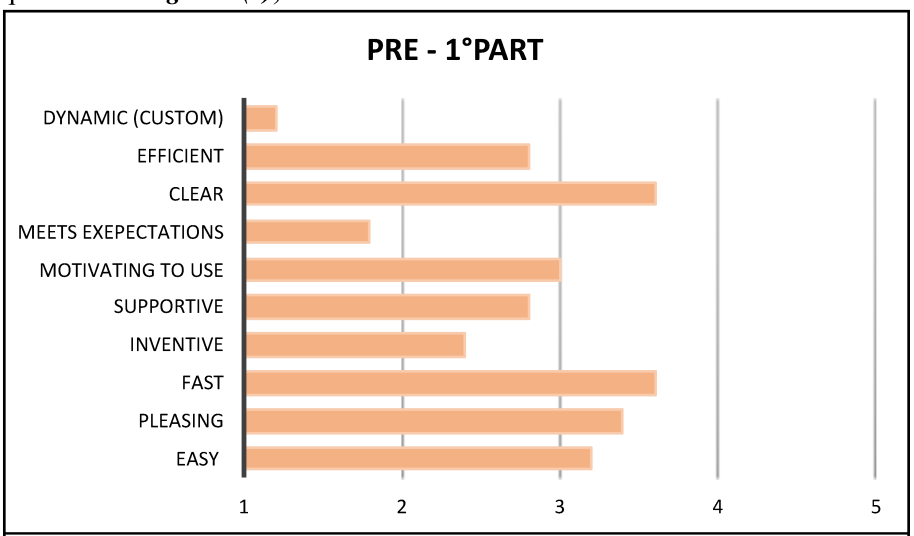


Fig. 103: The objective evaluation

From a subjective point of view, the pre and post-test questionnaires submitted to users were analyzed. From their comparison, the preference of users in using the adaptive system compared to the native system appears. The results obtained through the pre and post-test questionnaire are summarized in ***Fig. 106.*** In detail, the results from the pre-questionnaire calculated using the scores of all five subjects, show that users find the Hoover Wizard app clear, easy and pleasant to use (i.e. scores ≥ 3 in a 5- point scale ***Fig. 104 (1)***).

(1)



(2)

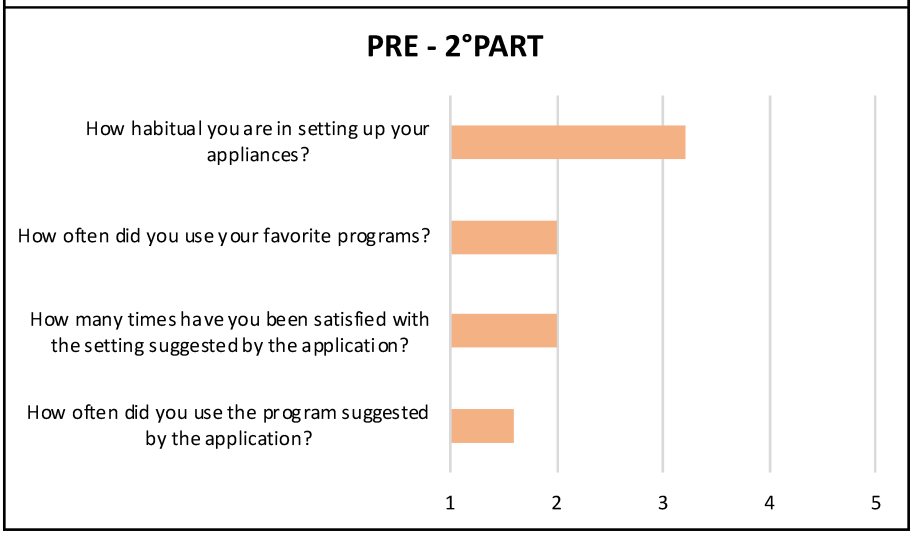


Fig. 104: The pre test questionnaire

However, most users have highlighted a lack of dynamicity and personalization, limited to the addition of programs in the favorite list (i.e., scores ≤ 2 in a 5-point scale ***Fig. 104 (1***)). Moreover, the default program recommended for each log-in does not respect the expectations of the user who is forced to manually select the preferred program (i.e., scores ≤ 2 in a 5-point scale ***Fig. 104 (2)***).
Finally, suggestions of default settings by the app (temperature, spin speed, etc.) do not meet the users expectations, who want to change and customize them (i.e., scores ≤ 2 in a 5-point scale ***Fig. 104 (2)***).
At the end of the experiment, the questionnaire was re-submitted to the users. The post-questionnaire using the scores of all five subjects shows how the new adaptive application retains attributes such as clarity and ease of use, and achieves higher scores on customization, efficiency, and support requirements (i.e. scores ≥ 3 in a 5- point scale ***Fig. 105 Fig. 104(1)***).
In detail, all users have found the benefit of using the adaptive app, which provides valid support during the setting of the appliance. Suggestions about the programs given by the app based on the day and hour of use respected the user's expectations, which reduced the number of changes accordingly. Moreover, the dynamic suggestions provided by the app in the setting phase (temperature and spin speed, etc.,) have met the user's expectations, obtaining an average of 4.6 / 5 in a 5- point scale ***Fig. 105 Fig. 104(2)***.

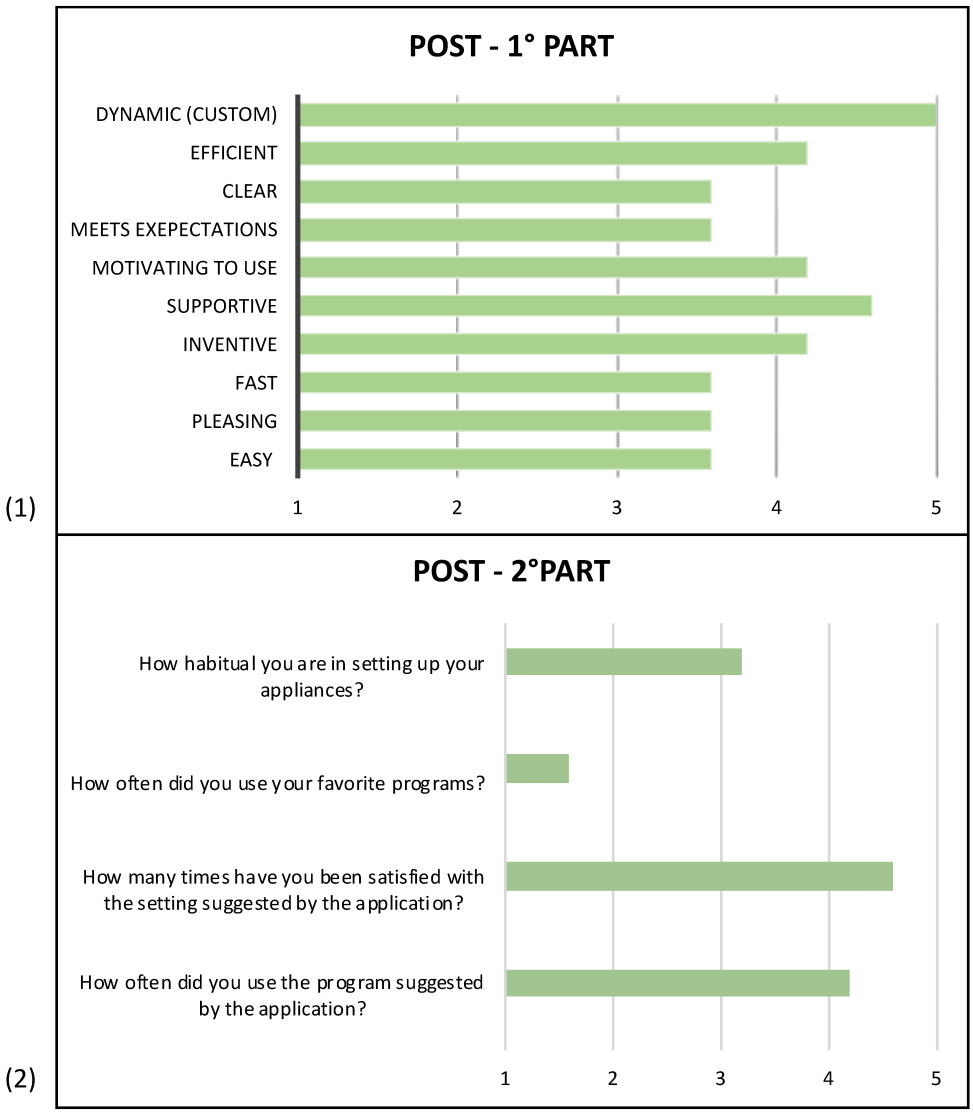


Fig. 105: The Post test questionaire

A comparison between the two Apps is showed in ***Fig. 106***.

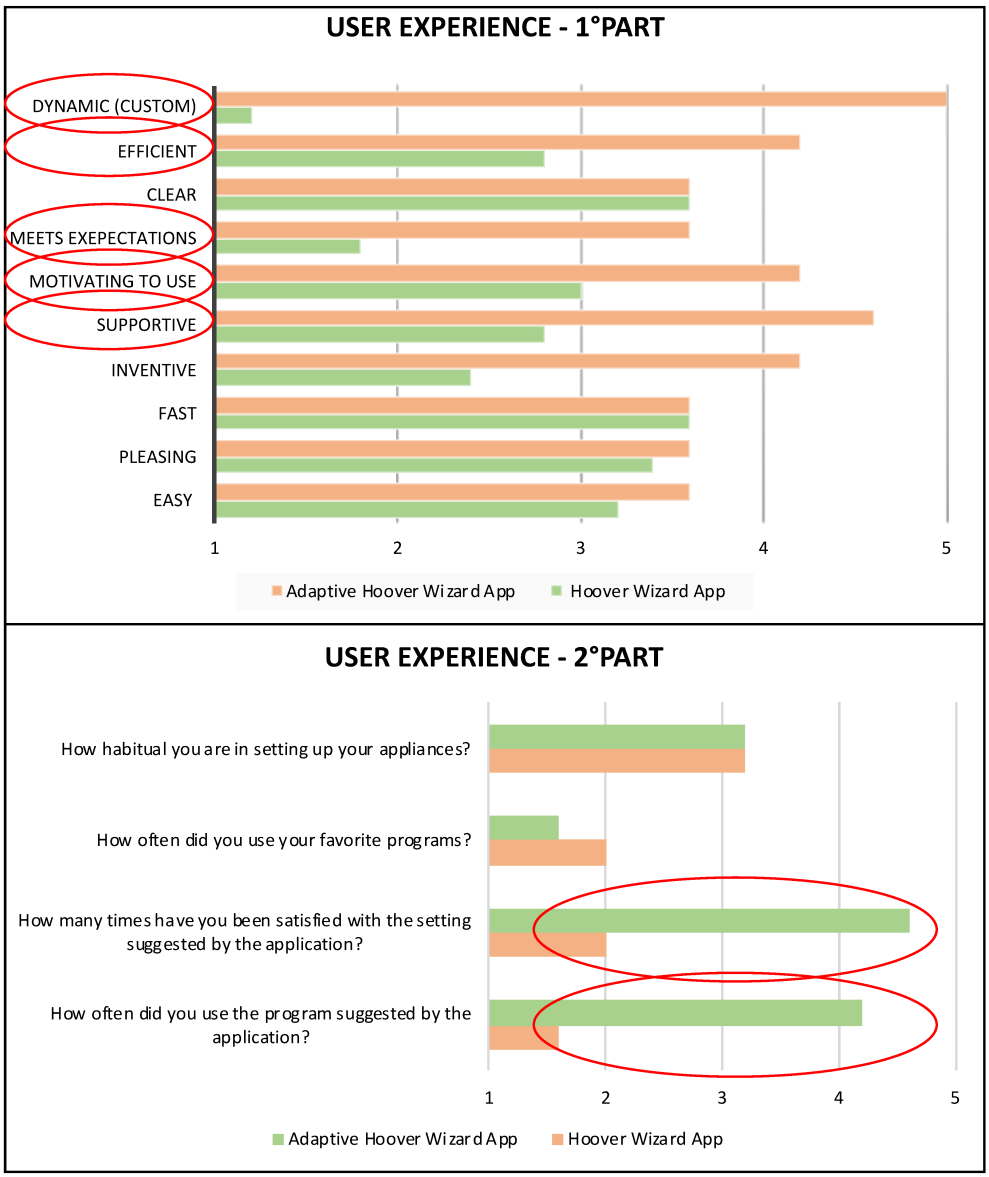


Fig. 106: Comparison of the user's experience with the Adaptive Hoover App and with Hoover App

In detail we see how the insertion of the adaptive component in the Hoover system has greatly improved the user's experience in setting up the washing machine. Finally, users were asked to evaluate the improvement of the new application

compared to the previous one with a score range of 1 to 5. The average of the results obtained is 4.8/ in a 5- point scale.

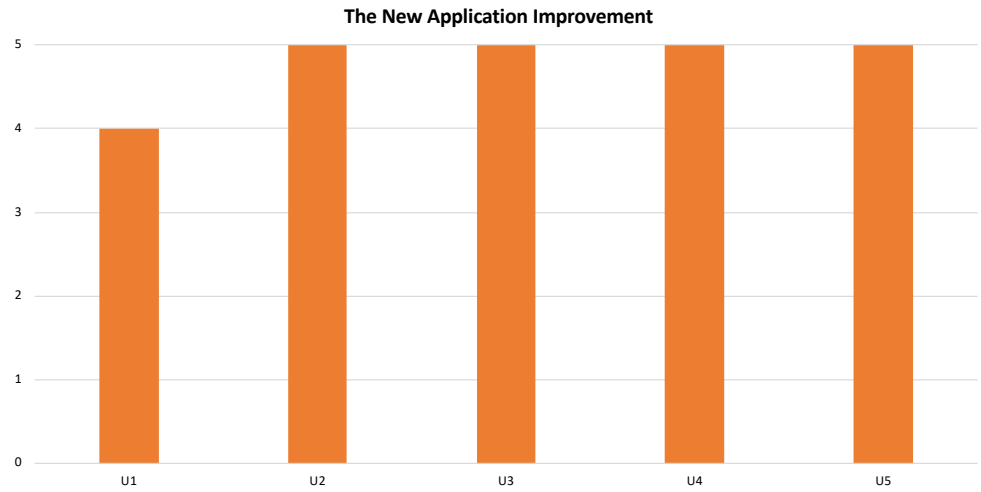


Fig. 107: The New Application Improvement

Chapter 7.

Conclusion

Nowadays, interaction with machines influences much of our daily life. In fact, every day we interact with different intelligent objects which are always richer in functions (mobile phones, appliances, cars, MP3 players, stereo systems, cameras) and which come into contact with a myriad of services managed by software. These software shapes the modalities in which we interact with objects and with other people.

The human-computer interaction discipline proposes methods to center the design of the machine interaction on the user (user-centered design). Solutions created with these technologies must take into account that they will be used by human beings. In order to make the technology both easy and useful to use, it is therefore necessary to know the various categories of people who will use the service. The already complex challenge of designing usable products and services has become more ambitious in recent years, since it began to study and to include in design the set of sensations and emotions that interaction with products or services causes to user (user experience) and in creating systems that adapt the use and preferences to users' (adaptive and adaptale system).

Adaptation and adaptability are on their way to become common features of user system that claim to provide access to users with different abilities (user interfaces for all). Facing this fact, one should expect a sound conceptual and technical background for adaptable or adaptive system design and evaluation. Unfortunately, neither a common terminology structure nor a complete framework for the design and evaluation of adaptive and adaptable user interfaces exists. In detail, the existing general approaches are not reflected in the applications, instead creating specific methodologies closely related to the application domains. Furthermore, this design process represents a difficult

challenge for the designer because it depends on the complexity of human intentions in a particular moment and place.
For these reasons, this thesis prject aims to study and develop a new method and a tool to support the design of adaptive and adaptable user interfaces.
The proposed method and the tool have been applied in:

- the design of a new adaptive and adaptable system created for the control and management of home appliances in a kitchen, both for users with learning and for able-bodied users and tested in a virtual reality environment.
- the re-design of the existing Hoover Wizard Application, in order to make it adaptive.

Experimental results show that the method and the means proposed enable to effectively support the design of adaptive and adaptable user interfaces.
In particular, the results from the first case study show interesting perspectives. As demonstrated by the SUS and MARS scores, the interface is generally usable by users with different impairments, in terms of typology and severity. The execution of several tasks allows to obtain interesting outcomes from different perspectives: usability and adaptability. From one hand, each task allows observing specific usability problems; from the other one, the overall test allows taking into account previous interactions and evaluating the improvement on usability due to the system adaptation. During the usability evaluation, some technical problems hindered the perfect outcome of the task and that could have reduced the perception of the usability of the application. They mainly refer to minor usability problems such as make some buttons more visible, insert some labels to clarify some commands, and rewrite some sentences to improve their understandability.
The major usability problem concerns the error prevention and recovery from errors. Several feasible solutions were hypnotized and will be implemented to solve it. A couple of tasks were not completed by one of the two users with dexterity problems. Some adaptations related to the page's layout (e.g. buttons closeness) allowed the user to complete the following tasks, but some difficulties remained (e.g. double tap zoom). This suggested to modify the interaction modalities and simplify the navigation anyway.

In case of users with cognitive impairments, the benefits of system adaptation (i.e. shift to wizard modality) were fully appreciated and were reflected by users' performance in the following tasks. However, transversal problems such as the use of the scroll function should be necessarily faced to improve the user experience. UI adaptations in terms of high contrast text, color inversion and zoom allows the successful execution of tasks by users with visual impairments.

Finally, the system adaptations, mainly tested in simulation, showed the effective adaptive behavior of the system itself due to the correct operation of the DM algorithm and the chosen threshold.

However, the need to test the adaptive system component with users in real life environments and the possibility of using the proposed tool and methodology with the aim of adapting an existing system emerged from this first case study.

For this reason, it was decided to extend this thesis work, introducing the second case study. The methodology and tool have been experimented with an Italian company, Candy Hoover Group, which aimed at improving its Wi-Fi system for the control and management of household appliances. It led designers in their activity, supporting the rules definition and reducing the overall efforts. Such approach allowed performing adaptation activities (i.e., monitoring, analysis, decision-making, execution, etc.) without affecting system performance and availability.

The results coming from the experimentation have measured the adaptability of the system perceived by the end-users. Furthermore, the results investigated aspects related to the usability, the design and the quality of the experience of use with the product and its interface.

In detail the adaptivity perceived by end-users during the use of the adaptive system compared to the non-adaptive Wizard system has been evaluated.

From an objective point of view, the results show how the users involved have drastically reduced the number of program changes, temperature and spin speed thanks to the fast and personalized suggestion provided by the adaptive system. Less noticeable changes are shown for the options and delayed start setting, due to the little use of these functions by the users involved.

From a subjective point of view, the questionnaires show how the addition of the adaptive component has greatly improved the user experience for using the system, which has appreciated the suggestions provided by the system and its ability to adapt dynamically to their usage preferences.
Despite the excellent results obtained, further analysis to investigate the dynamic adaptive dynamic behavior on a larger number of users should be performed.
Furthermore, the proposed methodology should be tested in different applications and contexts of use to validate it and to highlight its advantages and disadvantages.

References

Abras, C., Maloney-Krichmar, D., & Preece, J. 2004. User-centered design. Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications, 37(4), 445-456.

Alaeddini, A., & Dogan, I. 2011. Using Bayesian networks for root cause analysis in statistical process control. Expert Systems with Applications, 38(9), 11230-11243.

Anitha, G., & Priya, S. B. 2018. Posture based health monitoring and unusual behavior recognition system for elderly using dynamic Bayesian network. Cluster Computing, 1-8.

Application, N. 2004. A complete software Package to solve problem using Bayesian Belief Networks and Influence Diagrams. Norsys Software Corp.

Ardissono, L., Goy, A., Petrone, G., & Segnan, M. (2000). Adaptive user interfaces for on-line shopping. In 2000 AAAI Spring Symposium (pp. 13-18).

Arens, Y., Hovy, E. H., & Vossers, M. 1992. On the knowledge underlying multimedia presentations (No. ISI/RR-93-370). UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST.

Baresi, L., & Pasquale, L., 2011. Adaptation goals for adaptive service-oriented architectures. In Relating Software Requirements and Architectures (pp. 161-181). Springer, Berlin, Heidelberg.

Ben-Ari, M., & Mondada, F., 2018. Fuzzy Logic Control. In Elements of Robotics (pp. 179-183). Springer, Cham.

Bental, D., Cawsey, A., Pearson, J., and Jones, R., 2000. Adapting web-based information to the needs of patients with cancer. In Adaptive Hypermedia and Adaptive Web-Based Systems (pp. 27-37). Springer Berlin

Benyon D, Turner P, Turner, S., 2005. Designing interactive systems: a comprehensive guide to HCI and interaction design, 2nd edn. Pearson Education Limited, Edinburgh"

Benyon, D., and Murray, D., 1993. Applying user modeling to human-computer interaction design. Artificial Intelligence Review, 7(3), 199-225

Bi, S., & Salvendy, G. 1994. Analytical modeling and experimental study of human workload in scheduling of advanced manufacturing systems. International Journal of Human Factors in Manufacturing, 4(2), 205-234.

Bongartz, S., Jin, Y., Paternò, F., Rett, J., Santoro, C., and Spano, L., 2012. Adaptive user interfaces for smart environments with the support of model-based languages. Ambient Intelligence, 33-48.

Bordegoni M., Carulli M. 2016. Evaluating Industrial Products in an Innovative Visual-Olfactory Environment, Journal of Computing and Information Science in Engineering , 16(3):030904-030904-9, DOI:10.1115/1.4033229.

Brewer, J., and Dardailler, D., 2000. World Wide Web Consortium (W3C) Web Accessibility Initiative (WAI). World Wide Web Consortium.

Brown, C. M. 1999. Human-computer interface design guidelines. Intellect Books.

Brusilovsky, P. 1996. Methods and techniques of adaptive hypermedia. User modeling and user-adapted interaction, 6(2-3), 87-129.

Brusilovsky, P., & Millán, E. 2007. User models for adaptive hypermedia and adaptive educational systems. In The adaptive web (pp. 3-53). Springer, Berlin, Heidelberg.

Brusilovsky, P., and Eklund, J., 1998. A study of user model based link annotation in educational hypermedia. Journal of Universal Computer Science, 4(4), 429-448.

Brusilovsky, P., and Pesin, L., 1998. Adaptive navigation support in educational hypermedia: An evaluation of the ISIS-Tutor. CIT. Journal of computing and information technology, 6(1), 27-38.

Caesarendra, W., Pratama, M., Tjahjowidodo, T., Tieud, K., & Kosasih, B. 2018. Parsimonious Network based on Fuzzy Inference System (PANFIS) for Time Series Feature Prediction of Low Speed Slew Bearing Prognosis. arXiv preprint arXiv:1802.09332.

Carulli M., Bordegoni M., Cugini U. 2012. An approach for capturing the Voice of the Customer based on Virtual Prototyping, Journal of Intelligent Manufacturing, DOI: 10.1007/s10845-012-0662-5.

Caruso G., Re G.M., Carulli M., Bordegoni, M. 2014. Novel Augmented Reality system for Contract Design Sector, Computer-Aided Design and Applications, 11(4), pp. 389-398, DOI: 10.1080/16864360.2014.881181

Ceccacci, S., Cavalieri, L., Gullà, F., Menghi, R., & Germani, M. 2016. A universal design method for adaptive smart home environment. In International Conference on Universal Access in Human-Computer Interaction (pp. 359-369). Springer, Cham.

Chaminda, H. T., Klyuev, V., & Naruse, K. 2012. A smart reminder system for complex human activities. In Advanced Communication Technology (ICACT), 2012 14th International Conference on (pp. 235-240). IEEE.

Chin, C. S., & Lin, W. P. 2018. Robust Genetic Algorithm and Fuzzy Inference Mechanism Embedded in a Sliding-Mode Controller for an Uncertain Underwater Robot. IEEE/ASME Transactions on Mechatronics, 23(2), 655-666.

Ciolacu, M., & Beer, R. (2016, October). Adaptive user interface for higher education based on web technology. In Design and Technology in Electronic Packaging (SIITME), 2016 IEEE 22nd International Symposium for (pp. 300-303). IEEE.

Cockton, G. 1987. A new model for separable interactive systems. In Human–Computer Interaction–INTERACT'87 (pp. 1033-1038).

Cook, D., and Das, S. K. 2004. Smart environments: Technology, protocols and applications (Vol. 43). John Wiley & Sons.

Cooper, A., 1999. The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity (Indianapolis, Indiana: SAMS).

Darnell, S. J., Page, D., & Mitchell, J. C. 2007. An automated decision-tree approach to predicting protein interaction hot spots. Proteins: Structure, Function, and Bioinformatics, 68(4), 813-823.

Dieterich, H., Malinowski, U., Kühme, T., & Schneider-Hufschmidt, M. 1993. State of the art in adaptive user interfaces. Human factors in information technology, 10, 13-13.ISO 690 " "DIS, I., 2009. 9241-210: 2010. Ergonomics of human system interaction-Part 210: Human-centred design for interactive systems. International Standardization Organization (ISO). Switzerland.

Duda, J., Botzolakis, E., Chen, P. H., Mohan, S., Nasrallah, I., Rauschecker, A., ... & Cook, T. 2018. Bayesian network interface for assisting radiology interpretation and education. In Medical Imaging 2018: Imaging Informatics for Healthcare, Research, and Applications (Vol. 10579, p. 105790S). International Society for Optics and Photonics.

El-Bakry, H. M., Riad, A. M., Abu-Elsoud, M., Mohamed, S., Hassan, A. E., Kandel, M. S., and Mastorakis, N., 2010. Adaptive user interface for web applications. In Recent Advances in Business Administration: Proceedings of the 4th WSEAS International Conference on Business Administration (ICBA'10) (pp. 20-22).

Endsley, M. R. 1987. SAGAT: A methodology for the measurement of situation awareness (NOR DOC 87-83). Hawthorne, CA: Northrop Corporation. ISO 690

Endsley, M. R., & Strauch, B. 1997. Automation and situation awareness: The accident at Cali, Columbia. In Proceedings of the ninth international symposium on aviation psychology (pp. 877-881).

Evans, D., Arasu, V. A., & Fendley, M. 2013. A Model-Based Framework for an Adaptive User Interface in Supervisory Control. IFAC Proceedings Volumes, 46(15), 37-43

Eyoh, I., John, R., De Maere, G., & Kayacan, E. 2018. Hybrid learning for interval type-2 intuitionistic fuzzy logic systems as applied to identification and prediction problems. IEEE Transactions on Fuzzy Systems.

Feng, J., & Liu, Y. (2015). Intelligent context-aware and adaptive interface for mobile LBS. Computational intelligence and neuroscience, 2015, 5.

Findlater, L., & McGrenere, J. 2004. A comparison of static, adaptive, and adaptable menus. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 89-96). ACM.

Findlater, L., & McGrenere, J. 2008. Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1247-1256). ACM.

Fischer, G., 2001. User modeling in human–computer interaction. User modeling and user-adapted interaction, 11(1), 65-86

Fleishman, Edwin A., et al. "Taxonomic efforts in the description of leader behavior: A synthesis and functional interpretation." The Leadership Quarterly2.4 (1991): 245-287.

Francisco-Revilla, L., & Shipman III, F. M. 2000. Adaptive medical information delivery combining user, task and situation models. In Proceedings of the 5th international conference on Intelligent user interfaces (pp. 94-97). ACM.

Freund, Y., & Mason, L. 1999. The alternating decision tree learning algorithm. In icml (Vol. 99, pp. 124-133).

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. 2000. Using Bayesian networks to analyze expression data. Journal of computational biology, 7(3-4), 601-620.

Gajos, K. Z., Czerwinski, M., Tan, D. S., and Weld, D. S., 2006. Exploring the design space for adaptive graphical user interfaces. In Proceedings of the working conference on Advanced visual interfaces (pp. 201-208). ACM

Gajos, K. Z., Everitt, K., Tan, D. S., Czerwinski, M., and Weld, D. S., 2008. Predictability and accuracy in adaptive user interfaces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1271-1274). ACM.

Ghédira, C., Maret, P., Fayn, J., and Rubel, P., 2002. Adaptive user interface customization through browsing knowledge capitalization. International journal of medical informatics, 68(1), 219-228.

Goh, A. T. 1995. Back-propagation neural networks for modeling complex systems. Artificial Intelligence in Engineering, 9(3), 143-151

Greenberg, S., and Witten, I. H., 1985. Adaptive personalized interfaces—A question of viability. Behaviour and Information Technology, 4(1), 31-45.

Grundy, J., and Hosking, J., 2002. Developing adaptable user interfaces for component-based systems. Interacting with computers, 14(3), 175-194

Gullà, F., Ceccacci, S., Menghi, R., & Germani, M. 2016. An adaptive smart system to foster disabled and elderly people in kitchen-related task. In Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments (p. 27). ACM

Gullà, F., Ceccacci, S., Menghi, R., Cavalieri, L., & Germani, M. 2016. Adaptive Interface for Smart Home: A New Design Approach. In Italian Forum of Ambient Assisted Living (pp. 107-115). Springer, Cham.

Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J., and Cajander, Å., 2003. Key principles for user-centred systems design. Behaviour and Information Technology, 22(6), 397-409

Haas, M. W., & Hettinger, L. J. 2001. Current research in adaptive interfaces. The International Journal of Aviation Psychology, 11(2), 119-121.

Hackos, J. T., & Redish, J. 1998. User and task analysis for interface design.

Hall, T. W., Navvab, M., Maslowski, E., & Petty, S. 2012. Virtual reality as a surrogate sensory environment. In Advances in Robotics and Virtual Reality (pp. 251-273). Springer Berlin Heidelberg.

Hariyanto, D., & Köhler, T. 2017. An Adaptive User Interface for an E-learning System by Accommodating Learning Style and Initial Knowledge.

Hart, S. G. 2006. NASA-task load index (NASA-TLX); 20 years later. In Proceedings of the human factors and ergonomics society annual meeting (Vol. 50, No. 9, pp. 904-908). Sage CA: Los Angeles, CA: Sage Publications.

Hart, S. G., & Staveland, L. E. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In Advances in psychology (Vol. 52, pp. 139-183). North-Holland.

Hayes-Roth, F., Waterman, D. A., & Lenat, D. B. 1983. Building expert system

Haykin, S. S., Haykin, S. S., Haykin, S. S., & Haykin, S. S. 2009. Neural networks and learning machines (Vol. 3). Upper Saddle River, NJ, USA: Pearson

He, J., & Yen, I. L., 2007. Adaptive user interface generation for web services. In e-Business Engineering, 2007. ICEBE 2007. IEEE International Conference on (pp. 536-539). IEEE. ISO 690

He, J., Yen, I. L., Peng, T., Dong, J., and Bastani, F., 2008. An adaptive user interface generation framework for web services. In Congress on Services Part II, 2008. SERVICES-2. IEEE (pp. 175-182). IEEE.

He, X., & Xu, S. 2010. Artificial neural networks. Process Neural Networks: Theory and Applications, 20-42.

Heger, A. S., & Aradhye, H. B. 2002. U.S. Patent No. 6,415,276. Washington, DC: U.S. Patent and Trademark Office.

Hervás, R., and Bravo, J., 2010. Towards the ubiquitous visualization: Adaptive user-interfaces based on the Semantic Web. Interacting with Computers, 23(1), 40-5

Hettinger, L. J., Cress, J. D., Brickman, B. J., & Haas, M. W. 1996. Adaptive interfaces for advanced airborne crew stations. In Human Interaction with Complex Systems, 1996. HICS'96. Proceedings., Third Annual Symposium on (pp. 188-192). IEEE.

Hewett, T. T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., ... & Verplank, W. 1992. ACM SIGCHI curricula for human-computer interaction. ACM.

Hix, D., & Hartson, H. R. 1993. Developing user interfaces: ensuring usability through product & process. John Wiley & Sons, Inc.

Huch, M., Kameas, A., Maitland, J., McCullagh, P., Roberts, J., Sixsmith, A., & Wichert, R. 2012. Handbook of ambient assisted living. J. C. Augusto (Ed.). Amsterdam: IOS press.

ISO/IEC. 9241-14 Ergonomic requirements for office work with visual display terminals (VDT)s - Part 14 Menu dialogues, ISO/IEC 9241-14: 1998 (E), 1998.

Issa, T., & Isaias, P. 2015. Usability and Human Computer Interaction (HCI). In Sustainable Design (pp. 19-36). Springer, London.

Jääskeläinen, R. 2010. Think-aloud protocol. Handbook of translation studies, 1, 371-374.

Jameson, A. 2001. User-adaptive and other smart adaptive systems: Possible synergies. In Proceedings of the first EUNITE Symposium, Tenerife, Spain.

Jang, J. S. 1993. ANFIS: adaptive-network-based fuzzy inference system. IEEE transactions on systems, man, and cybernetics, 23(3), 665-685

Janikow, C. Z. 1998. Fuzzy decision trees: issues and methods. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 28(1), 1-14.

Jensen, F. V. 1996. An introduction to Bayesian networks (Vol. 210, pp. 1-178). London: UCL press

Jensen, F. V., & Nielsen, T. D. 2011. Probabilistic decision graphs for optimization under uncertainty. 4or, 9(1), 1-28.

Jiménez-Mixco, V., Villalar González, J. L., Arca, A., Cabrera-Umpierrez, M. F., Arredondo, M. T., Manchado, P., & García-Robledo, M. 2009. Application of virtual reality technologies in rapid development and assessment of ambient assisted living environments. In Proceedings of the 1st ACM SIGMM international workshop on Media studies and implementations that help improving access to disabled users (pp. 7-12). ACM.

Jimison, H. B., Pavel, M., & Pavel, J. (2003). Adaptive interfaces for home health. In Proceedings of the International Workshop on Ubiquitous Computing for Pervasive Healthcare.

Jokela, T., Iivari, N., Matero, J., & Karukka, M. 2003. The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11. In Proceedings of the Latin American conference on Human-computer interaction (pp. 53-60). ACM.

Jokinen, K., Rissanen, J., Keränen, H., & Kanto, K. (2002). Learning interaction patterns for adaptive user interfaces. In Adjunct Proceedings of the 7th ERCIM Workshop User Interfaces for all (pp. 53-58).

Kalogirou, S. A., & Bojic, M. 2000. Artificial neural networks for the prediction of the energy consumption of a passive solar building. Energy, 25(5), 479-491

Karaboga, D., & Kaya, E. 2018. Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey. Artificial Intelligence Review, 1-3

Karagiannidis, C., & Sampson, D. 2004. Adaptation rules relating learning styles research and learning objects meta-data. In Workshop on Individual Differences in Adaptive Hypermedia. 3rd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH2004), Eindhoven, Netherlands.

Karagiannidis, C., Koumpis, A., & Stephanidis, C. 1995. Supporting adaptivity in intelligent user interfaces: The case of media and modalities allocation. In Proceedings of the 1st ERCIM Workshop on User Interface for All.

Keppel, G, Saufley, WH and Tokunaga, H, 1998, Introduction to Design and Analysis, A Student's Handbook, 2nd edn.

Kim, L. W. 2018. DeepX: Deep learning accelerator for restricted boltzmann machine artificial Neural Networks. IEEE transactions on neural networks and learning systems, 29(5), 1441-1453.

Kobsa, A. 1990. User modeling in dialog systems: Potentials and hazards. AI & society, 4(3), 214-231.

Kobsa, A., & Pohl, W. 1994. The user modeling shell system BGP-MS. User Modeling and User-Adapted Interaction, 4(2), 59-106.

Kononenko, I. 1993. Inductive and Bayesian learning in medical diagnosis. Applied Artificial Intelligence an International Journal, 7(4), 317-337.

Kushwaha, N., Kim, M., Kim, D. Y., & Cho, W. D. 2004. An intelligent agent for ubiquitous computing environments: smart home UT-AGENT. In Software Technologies for Future Embedded and Ubiquitous Systems, 2004. Proceedings. Second IEEE Workshop on (pp. 157-159). IEEE."

Kushwaha, N., Kim, M., Kim, D. Y., & Cho, W. D. 2004. An intelligent agent for ubiquitous computing environments: smart home UT-AGENT. In Software Technologies for Future Embedded and Ubiquitous Systems, 2004. Proceedings. Second IEEE Workshop on (pp. 157-159). IEEE.

Lakho, S., Jalbani, A. H., Vighio, M. S., Memon, I. A., & Soomro, S. S. 2017. Decision support system for hepatitis disease diagnosis using bayesian network. Sukkur IBA Journal of Computing and Mathematical Sciences, 1(2), 11-19.

Langley, P. 1999. User modeling in adaptive interface. In UM99 User Modeling (pp. 357-370). Springer, Vienna.

Lata, S., Madaan, D., & Chaudhary, S. 2011. Review on Human Computer Interaction. Learning From History & Need of Rethinking. International Journal of Computer Science and Management Studies, 11(03).

Lavie, T., & Meyer, J. 2010. Benefits and costs of adaptive user interfaces. International Journal of Human-Computer Studies, 68(8), 508-524.

Letsu-Dake, E., & Ntuen, C. A. 2010. A case study of experimental evaluation of adaptive interfaces. International Journal of Industrial Ergonomics, 40(1), 34-40.

Li, C., Mahadevan, S., Ling, Y., Choze, S., & Wang, L. 2017. Dynamic Bayesian network for aircraft wing health monitoring digital twin. AIAA Journal, 55(3), 930-941

Li, F., Wang, W., Qu, J., & Han, H. (2018, July). The Design of Adaptive User Interface Based on the Grey Relational Grade. In Journal of Physics: Conference Series (Vol. 1060, No. 1, p. 012046). IOP Publishing.

Lin, Y., Breugelmans, J., Iversen, M., and Schmidt, D., 2017. An Adaptive Interface Design (AID) for enhanced computer accessibility and rehabilitation. International Journal of Human-Computer Studies, 98, 14-23.

Liu, J., Wong, C. K., and Hui, K. K., 2003. An adaptive user interface based on personalized learning. IEEE Intelligent Systems, 18(2), 52-5

Machado, E., Singh, D., Cruciani, F., Chen, L., Hanke, S., Salvago, F., ... & Holzinger, A. (2018). A Conceptual framework for Adaptive User Interfaces for older adults. IEEE. ISO 690.

Marshall, S. P. 2002. The index of cognitive activity: Measuring cognitive workload. In Human factors and power plants, 2002. proceedings of the 2002 IEEE 7th conference on (pp. 7-7). IEEE.

Martins, F. 1996. Semi-Automatic Design and Prototyping of Adaptive User-Interfaces. Paper Presented at the Proceedings of 2nd ERCIM Workshop on ‘‘User Interfaces for All’’, Prague.

Mayhew, D. J. 1992. Principles and guidelines in software user interface design (Vol. 1, No. 992, pp. 67-68). Englewood Cliffs, NJ: Prentice Hall.

McGrenere, J., Baecker, R. M., & Booth, K. S. 2002. An evaluation of a multiple interface design solution for bloated software. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 164-170). ACM.

Miñón, R., & Abascal, J. 2011. Supportive adaptive user interfaces inside and outside the home. In International Conference on User Modeling, Adaptation, and Personalization (pp.320-334). Springer,Berlin,Heidelberg. ISO 690

Mitchell, T.M. 1997. Artificial neural networks. Machine learing, 45, 81-127.

Mitrovic, N., and Mena, E., 2002. Adaptive user interface for mobile devices. Lecture notes in computer science, 2545, 29-43.

Mitrović, N., Royo, J. A., & Mena, E. 2008. Performance analysis of an adaptive user interface system based on mobile agents. In Engineering Interactive Systems (pp. 1-17). Springer, Berlin, Heidelberg.

Moore, D. M., Lees, B. G., & Davey, S. M. 1991. A new method for predicting vegetation distributions using decision tree analysis in a geographic information system. Environmental Management, 15(1), 59-71.

Mostafa, S. A., Mustapha, A., Mohammed, M. A., Ahmad, M. S., & Mahmoud, M. A. 2018. A fuzzy logic control in adjustable autonomy of a multi-agent system for an automated elderly movement monitoring application. International journal of medical informatics, 112, 173-184.

Muda, Z., & Mohamed, R. E. K. 2006. Adaptive user interface design in multimedia courseware. In Information and Communication Technologies, 2006. ICTTA'06. 2nd (Vol. 1, pp. 196-199). IEEE.

Murray, D. 1990. Building a user modelling shell. In Empirical Foundations of Information and Software Science V (pp. 209-220). Springer, Boston, MA.

Namgoong, H., Sohn, J. C., Cho, Y. J., and Chung, Y. K., 2006. An adaptive user interface in smart environment exploiting semantic descriptions. In Consumer Electronics, 2006. ISCE'06. 2006 IEEE Tenth International Symposium on (pp. 1-6). IEEE.

Nannapaneni, S., Dubey, A., & Mahadevan, S. 2018. Automated aircraft separation safety assurance using Bayesian networks. In 2018 Aviation Technology, Integration, and Operations Conference (p. 3199).

Nasoz, F., Lisetti, C. L., & Vasilakos, A. V. (2010). Affectively intelligent and adaptive car interfaces. Information Sciences, 180(20), 3817-3836.

Nestler, S., & Klinker, G. (2007). Towards adaptive user-interfaces: Developing mobile user-interfaces for the health care domain. Mobiles Computing in der Medizin (MoCoMed). ISO 690.

Nielsen, J. 2003 Usability 101. http://www.useit.com/alertbox/20030825.html . Accessed 1 May 2015

Nielsen, J. 1994. Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, NY.

Nielsen, J., & Landauer, T. K. 1993. A mathematical model of the finding of usability problems. In Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems (pp. 206-213). ACM.

Nielsen, J., & Molich, R. 1990. Heuristic evaluation of user interfaces. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 249-256). ACM.

Nielsen, T. D., & Jensen, F. V. 2009. Bayesian networks and decision graphs. Springer Science & Business Medi

Nikovski, D. 2000. Constructing Bayesian networks for medical diagnosis from incomplete and partially correct statistics. IEEE Transactions on Knowledge & Data Engineering, (4), 509-516.

Nivethika, M., Vithiya, I., Anntharshika, S., and Deegalla, S., 2013. Personalized and adaptive user interface framework for mobile application. In Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on (pp. 1913-1918). IEEE.

Norman, D. 2013. The design of everyday things: Revised and expanded edition. Constellation

Pannu, M., Anane, R., Odetayo, M., & James, A. 2011. Explicit user profiles in web search personalisation. In Computer Supported Cooperative Work in Design (CSCWD), 2011 15th International Conference on (pp. 416-421). IEEE.

Paramythis, A., Totter, A., & Stephanidis, C. 2001. A modular approach to the evaluation of adaptive user interfaces. In Empirical Evaluation of Adaptive Systems. Proceedings of workshop at the Eighth International Conference on User Modeling, UM2001 (pp. 9-24)

Park, J., and Han, S. H., 2011. Complementary menus: Combining adaptable and adaptive approaches for menu interface. International Journal of Industrial Ergonomics, 41(3), 305-316.

Park, J., Han, S. H., Park, Y. S., and Cho, Y., 2007. Adaptable versus adaptive menus on the desktop: Performance and user satisfaction. International journal of industrial ergonomics, 37(8), 675-684.

Pedrycz, W. 1993. Fuzzy control and fuzzy systems (2nd. Research Studies Press Ltd).

Petoud I., Pigneur Y., An Automatic and Visual Approach for User Interface Design, in Engineering for Human-Computer Interaction, North-Holland, pp 403-420.

Pham, B. T., Bui, D. T., Prakash, I., & Dholakia, M. B. 2017. Hybrid integration of Multilayer Perceptron Neural Networks and machine learning ensembles for landslide susceptibility assessment at Himalayan area (India) using GIS. Catena, 149, 52-63.

Plessas, A., Stefanis, V., Komninos, A., and Garofalakis, J., 2017. Field evaluation of context aware adaptive interfaces for efficient mobile contact retrieval. Pervasive and Mobile Computing, 35, 51-64.

Preece J, Rogers Y, Benyon D, Holland S, Carey T. 1994 Human computer interaction. Addison-Wesley, Wokingham

Puerta, A., & Eisenstein, J. 1998. Towards a general computational framework for model-based interface development systems. In Proceedings of the 4th international conference on Intelligent user interfaces (pp. 171-178). ACM.

Quinlan, J. R. 2014. C4. 5: programs for machine learning. Elsevier.

Razmerita, L., & Lytras, M. D. 2008. Ontology-based user modelling personalization: Analyzing the requirements of a semantic learning portal. In World Summit on Knowledge Society. Springer Berlin Heidelberg (pp. 354-363).

Redmond-Pyle, D., & Moore, A. 1995. Graphical user interface design and evaluation (guide): a practical process. Prentice Hall.

Reid, G. B., & Nygren, T. E. 1988. The subjective workload assessment technique: A scaling procedure for measuring mental workload. In Advances in psychology (Vol. 52, pp. 185-218). North-Holland.

Rich, E. 1979. User modeling via stereotypes. Cognitive science, 3(4), 329-354.

Rich, E. 1983. Users are individuals: individualizing user models. International journal of man-machine studies, 18(3), 199-214.

Rich, E. 1989. Stereotypes and user modeling. In User models in dialog systems (pp. 35-51). Springer, Berlin, Heidelberg.

Rim Rebai, M. A., 2016. Building and evaluating an adaptive user interface using a Bayesian network approach. International Journal of Computer Science and Information Security (IJCSIS), Vol. 14, No. 7,

Rogers, S., Fiechter, C. N., & Thompson, C. 2000. Adaptive user interfaces for automotive environments. In Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE (pp. 662-667). IEEE.

Rothrock, L., Koubek, R., Fuchs, F., Haas, M., and Salvendy, G., 2002. Review and reappraisal of adaptive interfaces: toward biologically inspired paradigms. Theoretical issues in ergonomics science, 3(1), 47-84.

Rouse, W. B., Geddes, N. D., & Curry, R. E. 1987. An architecture for intelligent interfaces: Outline of an approach to supporting operators of complex systems. Human-computer interaction, 3(2), 87-122. ISO 690

Sauro, J. 2011. Measuring usability with the system usability scale (SUS).

Savidis, A., & Stephanidis, C. 2004. Unified user interface design: designing universally accessible interactions. Interacting with computers, 16(2), 243-270.

Scallen, S. F., Hancock, P. A., & Duley, J. A. 1995. Pilot performance and preference for short cycles of automation in adaptive function allocation. Applied ergonomics, 26(6), 397-403.

Schneider-Hufschmidt, M., Malinowski, U., and Kuhme, T., 1993. Adaptive user interfaces: Principles and practice. Elsevier Science Inc.

Schneider, M., Shnaider, E., Kandel, A., & Chew, G. 1998. Automatic construction of FCMs. Fuzzy sets and systems, 93(2), 161-172.

Scime, A. (Ed.). 2005. Web Mining: applications and techniques. IGI Global.

Sears, A., & Shneiderman, B. 1994. Split menus: effectively using selection frequency to organize menus. ACM Transactions on Computer-Human Interaction (TOCHI), 1(1), 27-51.

Shakshuki, E. M., Reid, M., & Sheltami, T. R. 2015. An adaptive user interface in healthcare. Procedia Computer Science, 56, 49-58

Shen, X., Tan, B., & Zhai, C. 2005. Implicit user modeling for personalized search. In Proceedings of the 14th ACM international conference on Information and knowledge management (pp. 824-831). ACM.

Shneiderman, B., and Maes, P., 1997. Direct manipulation vs. interface agents. interactions, 4(6), 42-61.

Song, I. J., and Cho, S. B., 2013. Bayesian and behavior networks for context-adaptive user interface in a ubiquitous home environment. Expert Systems with Applications, 40(5), 1827-1838.

Sosnovsky, S., & Dicheva, D. 2010. Ontological technologies for user modelling. International Journal of Metadata, Semantics and Ontologies, 5(1), 32-71.

Sponselee, A. A. G. 2013. Acceptance and effectiveness of telecare services from the end-user perspective. Eindhoven University of Technology, Eindhoven, The Netherlands

Stephanidis, C., Karagiannidis, C., & Koumpis, A. 1997. Decision making in intelligent user interfaces. In Proceedings of the 2nd international conference on Intelligent user interfaces (pp. 195-202). ACM.

Stephanidis, C., Paramythis, A., Sfyrakis, M., Stergiou, A., Maou, N., Leventis, A., ... & Karagiannidis, C. 1998. Adaptable and adaptive user interfaces for disabled users in the AVANTI project. In International Conference on Intelligence in Services and Networks (pp. 153-166). Springer, Berlin, Heidelberg. ISO 690

Sugeno, M. 1985. An introductory survey of fuzzy control. Information sciences, 36(1-2), 59-83.

Taylor, R. M. 2017. Situational awareness rating technique (SART): The development of a tool for aircrew systems design. In Situational Awareness (pp. 111-128). Routledge.

Tchankue, P., Wesson, J., & Vogts, D. (2011, November). The impact of an adaptive user interface on reducing driver distraction. In Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (pp. 87-94). ACM.

The Interaction Design Foundation https://www.interaction-design.org/literature/topics/user-centered-design

Tso, G. K., & Yau, K. K. 2007. Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. Energy, 32(9), 1761-1768

Tullis, T. S., & Stetson, J. N. 2004. A comparison of questionnaires for assessing website usability. In Usability professional association conference (Vol. 1).

Van den Broek, G., Cavallo, F., & Wehrmann, C. (Eds.). 2010. AALIANCE ambient assisted living roadmap (Vol. 6). IOS press.

Van Velsen, L., Van Der Geest, T., Klaassen, R., & Steehouder, M. (2008). User-centered evaluation of adaptive and adaptable systems: a literature review. The knowledge engineering review, 23(3), 261-281.

Vasilyeva, E., Pechenizkiy, M., & Puuronen, S. 2005. Towards the framework of adaptive user interfaces for eHealth. In Computer-Based Medical Systems, 2005. Proceedings. 18th IEEE Symposium on (pp. 139-144). IEEE

Viano, G., Parodi, A., Alty, J., Khalil, C., Angulo, I., Biglino, D., ... & Lachaud, P. 2000. Adaptive user interface for process control based on multi-agent approach. In Proceedings of the working conference on Advanced visual interfaces (pp. 201-204). ACM.

W3C., W. C. 2017: http://www.w3.org/TR/WCAG20/ [Accessed May 24, 2017].

Wang G., 2002. Definition and review of virtual prototyping. Journal of Computing and Information Science in Engineering, 2(3).

Wang, X., Hua, Q. Y., Zou, F., and Guo, L., 2014. An adaptive user interface model for mobile devices based on perceptual control theory. In Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on (pp. 908-911). IEEE.

Weidl, G., Madsen, A. L., & Israelson, S. 2005. Applications of object-oriented Bayesian networks for condition monitoring, root cause analysis and decision support on operation of complex continuous processes. Computers & chemical engineering, 29(9), 1996-2009.

Wharton, C. 1994. The cognitive walkthrough method: A practitioner's guide. Usability inspection methods.

Whiteside, J, Bennett, J and Holtzblatt, K, 1988, Usability engineering: our experience and evolution in M Helander (ed.) Handbook of Human–Computer Interaction New York: North-Holland, 791–817

Yeh, I. C. 1998. Modeling of strength of high-performance concrete using artificial neural networks. Cement and Concrete research, 28(12), 1797-1808.

Yen, G. G., and Acay, D., 2009. Adaptive user interfaces in complex supervisory tasks. ISA transactions, 48(2), 196-205.

Zudilova-Seinstra, E., 2007. On the role of individual human abilities in the design of adaptive user interfaces for scientific problem-solving environments. Knowledge and Information Systems, 13(2), 243-270.