Open University of Cyprus

Faculty of Pure and Applied Sciences

Master's Degree *Cognitive Systems*

Postgraduate (Master's) Dissertation

Call Assistant- an Application of Natural Language Based Dialogue System Using Machine Coaching and Argumentation

Iraklis Papadopoulos

Supervisor
Loizos Michael

June 2021

Open University of Cyprus

Faculty of Pure and Applied Sciences

Master's Degree *Cognitive Systems*

Master's Thesis

Call Assistant- an Application of Natural Language Based Dialogue System Using Machine Coaching and Argumentation

Iraklis Papadopoulos

Supervisor
Loizos Michael

The present Postgraduate (Master's) Dissertation was submitted in partial fulfilment of the requirements for the postgraduate degree
in Cognitive Systems
Faculty of Pure and Applied Sciences
of the Open University of Cyprus.

June 2021

BLANK PAGE

Summary

The last few decades, scientists have been trying to design and produce machines capable of thinking and acting like humans "embedded" them, in a way, with cognitive abilities and human intelligence. The construction of such machines applies the research that has been conducting by several cognitive psychologists who have tried to describe the way we comprehend the information and how our cognitive functions operate and interacting together. Also, other researchers had tried to examine how humans make inferences and how these inferences are produced under a certain contexts. The results of this research will be used for the construction of a system, the Call Assistant, capable of voice interactions using itself as a personal automation system, designed for managing the phone calls. The agent should be able to learn, and to be improved, from its past interaction(s) with the user, by offering personalized solutions. Constructing the Call Assistant, we will review the hypotheses that argumentation-in the form of rules- is one of the tools which it can establish a common "language" that machines and humans can utilize when interacting through machine coaching

Acknowledgments

I would like to thank to my Professor Loizo Michael, for giving me the opportunity to review a field of this matter and for his support throughout the thesis work.

Also, I would like to thank Christo Rodosthenous, Christodoulo Ioannou and Vassileio Marko for their comments and suggestions, as well as, the participants, who tested the application and to get to position to provide their feedback.

Finally, this thesis is dedicated to the memory of my parents, Stavroula and Haralampos, for their endless love and support.

Table of Contents

Contents

Chapter 1
Introduction

1.1 Research Objective

The last few decades, scientists have been trying to design and produce machines capable of thinking and acting like humans "embedded" them, in a way, with cognitive abilities and human intelligence. This is somehow realized by the development and construction of intelligent agents that are capable of actions based on information they perceive, their own experiences, and their own decisions about which action they perform.

The purpose of the thesis is the construction of an application, the Call Assistant, in order to review the hypotheses that argumentation-in the form of rules- is one of the tools which it can establish a common "language" that machines and humans can utilize when interacting through machine coaching.

The Call Assistant will be capable of voice interactions using itself as a personal automation system, designed for managing the phone calls. Main goal is to use physical interactions between the user and the phone-call system while for assistant's learning purposes, while at the same time, we want to increase its usability. The assistant will operate vocally through a conversation with its user(s) minimizing the need for them to interact each other with any keyboard/touch-screen enabled systems.

1.2 Research's questions

Cognitive science tries to model the human mind in order to simulate its functions. In brief, cognitive systems use "artificial cognitive abilities" or intuitive physics and psychology, the terms that are used by some researchers (Lake, Ullman, Tenenbaum & Gershman, 2016: 17), speech, natural language processing, machine learning and reasoning in order to model the human mind and to simulate the functions of the human brain. The purpose of these systems is to improve humans' decision making and to provide the best possible outcomes. Key elements are the natural language, which humans use for their communication, learning, reasoning and more specific the ways they store, retrieve, and use common sense knowledge.

1.3 Necessity and importance of research

For the last few years we are familiar with the term "life coaching". A rough description of the term is the following, life coaching is a procedure of consultation by a life-coach (expert) who helps a person to reach a goal or to make a number of changes in his life, changes which the person believes that are necessary for success. During this procedure, the life-coach guides the person and evaluates the steps towards to the goal playing in that way the important roles of motivator, strategist and accountability partner. In brief, life-coach helps the person to reach his goal in the most efficient and effective way that is possible.

This concept became a trend. However, the idea is not innovative since John Mc Arthy, an American computer scientist pioneer and inventor, had spoken for coaching but instead of humans he had meant machines. He had developed the idea that through coaching, just as life coaching, yet this time he had referred to machines, we could "guide" or even more to "teach" a machine to reason like a human, making the machine able to "think" and to manage ways of self-improvement. But is that possible? Can we simulate human intelligence? Even more, is this the right question that reflects the real problem?

The term "simulation" is often confused with the term "modelling" and both are used as having the same concept. Yet, they are distinct, though related, in a way that the first premises the second. **Modeling** is the representation of an object or phenomena, which is used by simulation. They may be mathematical, physical, or logical representations of a system, entity, phenomenon, or process. **Simulation** is a representation of the functioning of a system or process. Through simulation, a model may be implanted with unlimited variations, producing complex scenarios. These capabilities allow analysis and understanding of how individual elements interact and affect the simulated environment. In other words modeling represents the system itself, whereas the simulation represents the operation of the system over time. So the right question seems to be the following: can we model the human reasoning and intelligence in order to "construct" them and embedded them in the machines and have a simulation of human intelligence by them?

There is no an easy answer on this and even more, Mc Arthy's idea has a long way to be fulfilled, mainly for two reasons. The first is brittleness and the second is the transparency. The machines are brittle because they do not have the cognitive procedures <u>to form the</u>

context of the various situations that occur in reality. Therefore machines are brittle and break very often when they have to deal with a different condition comparing with the training examples that they have seen. Additionally, the machines are not transparent which means that they can make decisions yet they cannot provide explanations about these decisions. For instance, a machine can "reason" that red sport cars cause more traffic accidents but they cannot provide arguments for this decision (for example a possible assumption could be that the majority of the red sport cars are driven by young drivers who are more aggressive in driving).

Key element for the above concepts is the common sense. Common sense is based on perception understanding and judgment and people with common sense are seen as reasonable, down to earth, reliable, and practical. However we see that people make different decisions under the same states and in fact these decisions and actions can be consider the opposite of what we call common sense. For example drive or not while been drunk, smoking etc. In other words peoples' argumentation, decisions and actions are affected by factors such as knowledge, culture, feelings or states of emotions which many times can make them "unreasonable" and not predictable. On the other hand, machines do not perform in this way. Physical and emotional condition, mood states are out of the question. Under this frame it seems that the term "contains" the set of rules, arguments and actions of what is "efficient" and most logical to be executed according to the conditions and the demands of the environment in which a human has to perform.

Reasoning and argumentation are going together in a natural automation way. Reading comprehension is an example of how these mechanisms work together and provide results. In general, people can easily summarize an article after completely reading it by giving its character, place, process, etc. On the other hand, enabling a machine to complete reading comprehension and to participate in a question-answering procedure in connection with machine coaching and natural language is one of the core difficulties of artificial intelligence and a core difficulty in the current intelligent voice interaction and man-machine dialogues. The research on machine coaching by using natural language is to endow an agent with reading and speaking ability equal to a man, to cooperate with its user, capable to provide explanations and to receive coaching, thus to change behavior and actions in unstable and pre-set environment.

Chapter 2 Fundamentals of Cognitive Technology

2.1 Introduction

Technology development, has succeeded to combine sciences like mathematics, physics, phycology and others (Angel Garrido 2010: 1133, "Physics Boosts Artificial Intelligence Methods", 2021, "Making a thinking machine", 2021) in order to design machines and devices (such as robots or, cognitive assistants) that in some level interact with their environment and perform somehow in an intelligent way similar to those that humans use daily. This is more obvious if we just observe how web's search machines, like Google, function or how our smart phones operate. Even the cinema, by a number of science fiction story films, tries to describe and to give a "perspective" of how this expansion will be evolved. Terms, like cognitive systems, artificial intelligence, computational intelligence, objects recognition, machine learning and others have been arisen and they are embedded more and more in our daily routine. However, the majority of people use these terms like having the same meaning, function or result. Yet, all these terms have different definitions, scope, ways of approach and working, results.

Four critical terms, are Artificial Intelligence (AI), Computational Intelligence (CI), Machine Learning (ML) and Cognitive Computing. Although their general purpose is to make a machine to operate intelligent this happens by using different approaches, methods and procedures.

2.2 Artificial Intelligence and Computational Intelligence

Artificial Intelligence aims to create intelligent agents capable to act in different environments, to perceive and learn from these environments, to use their experiences, to make or change decisions accordingly, to achieve their goals or the best expected outcomes in any case and thus to become "autonomous". In other words, AI tries to establish methods that will "embedded" human cognitive functions, such as perceiving, reasoning and learning, into the machines making them intelligent (Kakas, A & Michael, L., 2016: 14-20).

On the other hand, CI is a branch of AI, a development that aims to create a system capable to reason and thinking similar to the ones that humans use, by using low level knowledge representation and bottom up techniques. There are several cases that data are too big to be processed or they cannot be addressed by mathematics or computer science's algorithms because they might be too fuzzy or uncertain. This is the field area in which CI gives answers mimicking intelligence properties found in humans and nature and by employing techniques and methods CI provides (a level of) intelligence.

Summarizing, both AI and CI are in the same page, they both have the same target, thus to create intelligent agents, however they try to succeed it by using different perspectives, methods and approaches. For example, for both fields have the element of reasoning. In case of AI, reasoning is succeed though the computational argumentation and explain the various forms that it has as well as the frameworks under which this kind of arguments take place(Kakas, A & Michael, L., 2016: 14-20) . We have the use of preference – based argumentation and argumentation – based logic. In the first case, an argument may be preferred to another one when it is stronger or its beliefs have a higher probability. In the second case, the argumentation deals with inconsistent information and its arguments attack each other while there is an evaluation of arguments. In case of CI things do not work this way. In CI the tool for reasoning is Fuzzy logic and it aims the uncertainty. Communication between humans is in linguistics terms in contrast to precise numeric data demanded by computers. Any model of the human decision-making must take into account the structure of information processing within the human organism and such essential features as its ability to handle ambiguity, imprecision, and human language in its highly developed complexity. Fuzzy Set Theory was formalized by Professor Lofti Zadeh at the University of California in 1965. Zadeh proposed a set of rules and regulations which defines boundaries and tells us what to do to be successful in solving problems within these boundaries. In that way CI is capable to work with concepts like "big", "small", "hot", "cold" etc (Zimmermann, 2010: 317-332).

2.3 Machine Learning

The engine behind of the AI/CI development is the machine learning. It provides the underlying technology that drives AI/CI. The primary goal of machine learning is to derive general patterns from a limited amount of data It has the ability to modify itself when exposed to more data is dynamic and does not require human intervention to make certain changes. The learning task is to predict some additional aspect of an input object. Examples of such a task are the simple problem of trying to predict a person's weight or height given some input such age or sex and the more complex task like the one which is described by Mark Gold (Gold, 1967: 447-474). For instance, the idea behind the Identification in the Limit Learning Paradigm is the construction of a model «able to identify and speak a language" in an artificially manner. The model uses an algorithm that is fed with an infinite sequence of data (a finite of set A that contain the alphabet of each language and a finite set SA of strings of elements from A). When a new element is given to the algorithm it may output a hypothesis (At each time *t* the learner is to make a guess *g* of a name of *L* based on the information it has received through time *t*.) The algorithm makes a hypotheses and identifies the language in the limit of time t while there exists a period of time from when the algorithm does not change its hypothesis, and this hypothesis is a correct representation of the target language. When a whole class of languages is considered, the algorithm identifies the class in the limit of time t if it can identify all languages of the class.

In brief, machine learning gives to computers and machines through its algorithms and methodologies the ability to automatically learn and improve from experience without human intervention and without being explicitly programmed.

However mathematical models like the one that has been described above can be used to make predictions, but often lack an explanatory component. This problem is more transparent in cases that involve complicated emotions and reasoning. For example, committing traffic violations is a good predictor for a machine to "explain" whether a person is a bad risk for car insurance, but it is rather difficult for a machine to provide solid explanations and suggestions to a human for improvements in a workplace. Furthermore the human brain seems to handle many operations and to process information from many sources simultaneously—in parallel. CI seems to "solve" these problems by adapting computational models that simulate somehow the human brain. A case like this is neural networks which is a simulation of biological neural networks. A network is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. Neural networks learn by example. They cannot be programmed to perform a specific task (Michael Nielsen, 2019: 1-10).

Now, we have a transition from machine learning to what of we call deep learning. The word "Deep" indicates the ability to model many more layers of virtual neurons than ever before since we have the computational power to succeed that. This evolution creates remarkable results in speech and image recognition. For example, a Google's deep-learning system that had been shown 10 million images from YouTube videos proved almost twice as good as any previous image recognition effort at identifying objects such as cats (Markoff John 2012, Hof Robert 2015).

Can we go beyond deep learning? The answer seems to be the fourth critical term: Cognitive Computing (CC).

2.4 Cognitive Computing

Cognitive Computing (CC) is an approach which tries to mimic human cognitive functions, in other words, it tries to create models of how the human brain/mind senses, reasons, and responds to stimulus.

The significance of CC in the IBM's figure (figure 1) which presents three eras of computing: Tabulating Era, Programmable Era and Cognitive Computing Era.

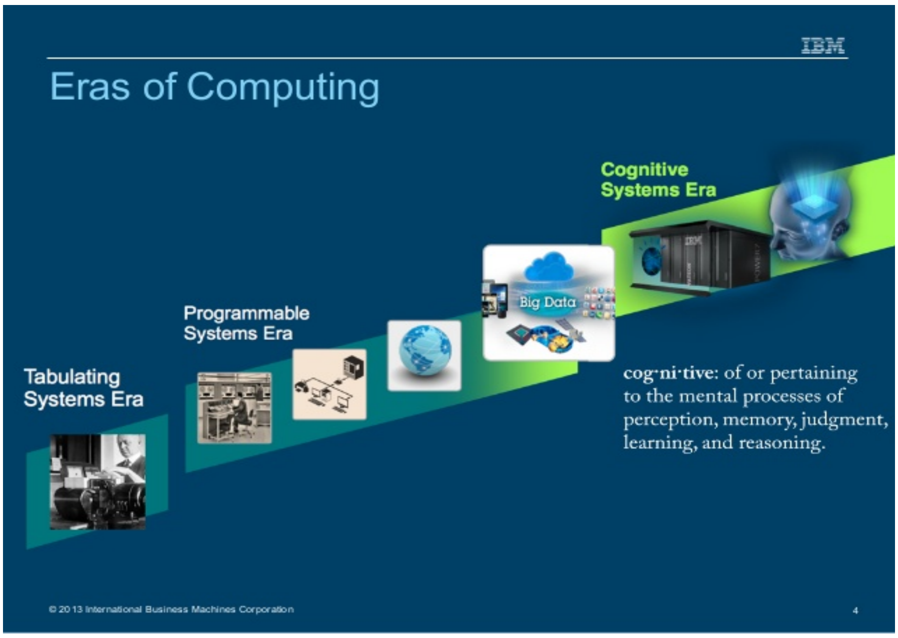


Figure 1: The eras of Computing

More specific, the figure 1 displays the history of cognitive computing as IBM's Senior Vice President John E Kelly sees it from his perspective. Kelly argues that in the Tabulating Era we had primitive machines in which we had stored data as he describes (1)*"In the first era of data we basically fed data in on punch cards."* ("A Brief History of Cognitive Computing - DATAVERSITY", 2014) while in the Programmable Era we had substitute the primitive machines with our current micro-processing computers . As he states (2) *"It was about taking processes and putting them into the machine. It's completely controlled by the programming we inflict on the system."* ("A Brief History of Cognitive Computing - DATAVERSITY", 2014).

The third era is the Cognitive Computing Era in which computers work directly with humans in a more synergetic association. Machines are equipped with cognitive abilities in such a way that the computers will help the humans to manage unravel vast stores of information and to provide solutions.

Core target is the development of models and algorithms constructed by the theories in cognitive science. In that way the machine will exhibit a kind of human-like cognitive intelligence while it would be able to understand and cognize the objective world from the perspective of human thinking. This approach needs the existence of cognitive abilities in order to have improvement in machine's intelligence and decision-making ability.

Various researchers try to develop smart systems, assisted by cognitive computing and cloud computing. First, we provide a comprehensive investigation of cognitive computing, including aspects from knowledge discovery, cognitive science, and big data. An idea of how these effort is developed is displayed in figure 3 (Chen M., Herrera F. and Hwang K. 2018).

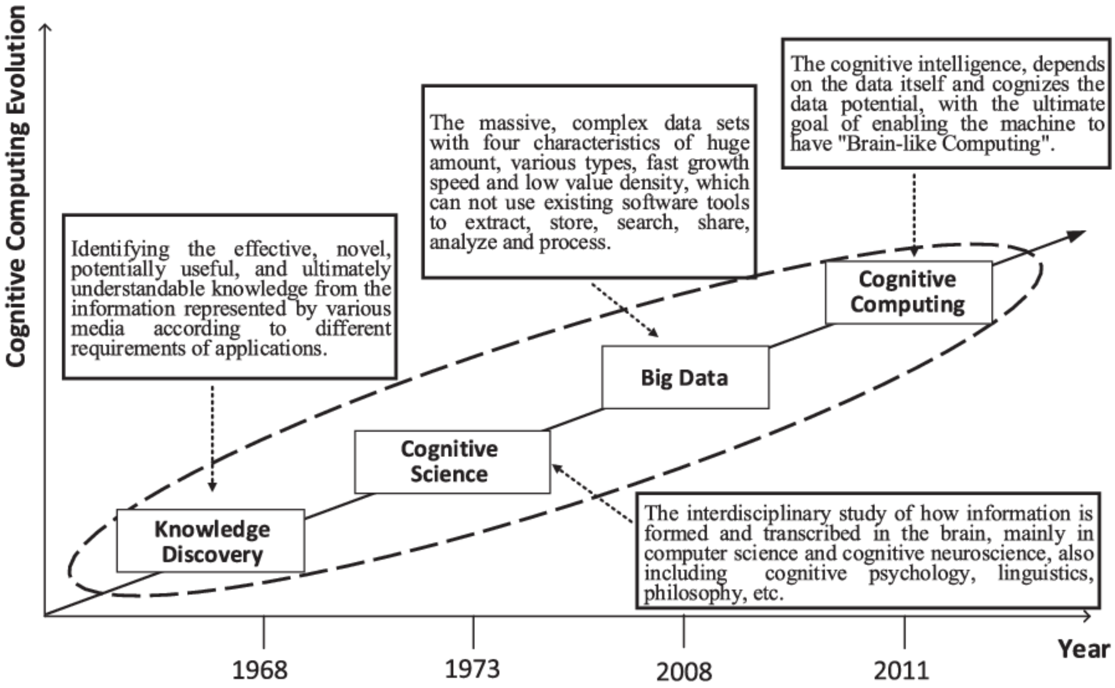


Figure 3: evolution process of cognitive computing (Chen M., Herrera F. and Hwang K. 2018)

However this venture looks hard to be implemented, at least for the time being. But what make this task so difficult? A first answer is that the machines we built show a lack of cognitive functions. Additionally, there are other issues that we need to overcome as the next chapter describes.

2.5 Cognitive Architectures

2.5.1 What is a Cognitive Architecture

Cognitive architecture is characterized by three aspects, first is the memories (short and long memory) that hold beliefs, goals and knowledge. The second is the mental structures which represent the elements which memories store and their organization. The third aspect is the processes (including performance and learning mechanisms) that operate on these structures. The design of these architectures has taken into account that different knowledge bases and beliefs can be interpreted by the same architecture yet, each design can make different assumptions about how to represent, use or acquire these aspects. The main goal is either to model the invariant aspects of the human cognition or to construct intelligent agents. Various authors (Vernon D., Metta G. and Sandini G,2007: 151-180) had clarified that this design is different from the experts' system design which, the latter, provided skill behavior in narrowly defined contexts. The idea is to have an "intelligent" system rather than a component designed for specialized task.

Regarding the evaluation criteria the authors (Vernon D., Metta G. and Sandini G,2007: 151-180) argue that they should demonstrate the same quality comparing those that evaluate aspects of the human behavior since this architecture is based on humans' psychological phenomena. This argument is also supported by the claim that in the field of software engineering we have independent modules with minimal interaction while in this case, the modules support synergistic effect by providing a unified process of cognition. The authors had proposed the following criteria for the evaluation of these architectures and these are: generality, versatility, taskability, rationality, optimality, efficiency, scalability, reactivity, persistence, improvability, autonomy and extended operation. More specific, generality measures the framework's ability to perform intelligent behavior in more environments with a broader domain and a variety of task. In other words, the system should demonstrate intelligence by dealing various aspects from more complex and broad domain(s). This ability, as the authors had stated, needs the implementation of new systems which will "manage" every domain respectively. This effort is measured by the versatility. Less effort to create intelligent behavior leads to more versatility. The number of tasks that the system can perform and their diversity is measured by the taskability. So we need a system as general as possible, capable to perform a large number of diverse tasks with minimum effort on the developer's part.

The relationship among framework's knowledge, goals and actions is measured by rationality. This means that the system should know which of its actions will lead to the goal. The percentage of times that this behavior satisfies the criterion gives the rationality of the system. The framework should produce optimum or as best as possible outcomes in every case. This ability is measured by optimality which describes the degree to which the system produces optimal results. Other researchers had used the term bounded rationality that measures the resources which the system has available for each decision.

The system should be efficient in terms that it satisfies all its constraints on time and space, as in work on real time system (efficiency). Additionally, the system should be scalable meaning that its efficiency is intact by complicating factors such as environmental uncertainty, task difficulty etc.

The framework's ability to react with speed in order to respond to unpredictable changes or unexpected situations which occur in the environment is measured by reactivity. This is crucial when the environment is more dynamic and we might have the frame problem that might cause the "breaking" of the system. In any case the system must be persistent and it must always continue to pursue its goals despite any possible changes in the environment. In other words despite the changes the system's reactions to these it should always continue to fulfill its long term objectives.

The system should have the capacity to add knowledge or to learn from its experiences with environment and therefor to perform new tasks that it could not operate before. This means that the system should have the ability to improve itself (improvability) by adding itself more knowledge and reusing this gained knowledge in a new broader range of tasks. Finally, the system should support a level of autonomy and extended operation by making autonomous decisions or to create its own tasks and goals so that will help it to achieve its goals.

2.5.2 Approaches and Paradigms

We have three different paradigms of cognition, the cognitivist approach, the emergent approach and the hybrid approach which combines aspects of the emergent systems and cognitivist systems. The cognitivist approach is focused on our visible behavior without understanding the internal processes that create it. It is based on the principle that our behavior is generated by a series of stimuli and responses to these by thought processes. The Information processing in cognitivist approach can be viewed in terms of three different kinds of components. First, we have the perception of the external world and the use of symbolic representations of the world's states and behavior of the external world. Second, we have components used for implementing the commands of the symbolic representations. And third, we have the reasoning in which we learn how to solve the problems in the first place. The emergent approach indicates that internal representations have central role and the perception is used for recognition and not for action. Knowledge appears to be not only reconstructive—a reproduction of what was learned, based on recalled data and on inferences from only those data. It is also constructive—influenced by attitudes, subsequently acquired information, and schemas based on past knowledge. An example of this kind of approach is the use of neural networks. These networks simulate parallel processing and they can model cognitive behavior without recourse to the kinds of explicit rules found for example in the production systems (cognitivist approach). They do this by storing patterns of activation in the network that associate various inputs with certain outputs. The hybrid approach does not use explicit programmer-based knowledge in the creation of artificially intelligent systems and uses perception-action behaviors rather than the perceptual abstraction of representations. They make a use of symbolic knowledge to represent the agent's world and logical rule-based systems to reason about this knowledge in order to achieve goals and select actions, while at the same time using emergent models of perception and action to explore the world and construct this knowledge. A cognitive architecture defines how memories are stored and the processes that operate on those memories. In terms of cognitive models it defines the formalisms for knowledge representations and the learning mechanisms that acquire it. In the following there is a brief description of some of the most important cognitive architectures of all three types: cognitivist, emergent and hybrid.

Cognitivist

Soar : The Soar (State, Operator and Result) system operates in a cyclic manner, with a production cycle and a decision cycle. In the first cycle, all productions that match the contents of declarative (working) memory, fire. A production rule can also be seen as a current state, e.g. the current position in amaze. A production that fire (movement in the maze) may alter the current declarative memory (new position in maze) and cause other productions to fire. This loop is repeated until no more productions fire. A decision cycle starts in which a single action from several possible actions is selected, based on the action's preferences. If the cycle reaches an impasse, i.e., no action is available, a new state in a new problem state is setup. This process is called subgoaling, where the new goal resolves the impasse. Additionally a new production rule is created which summarizes the processing that occurred in solving the sub goal. Soar is suitable for reasoning and planning but is weakly addressed for anticipation and adaption. More than one productions can be fired in one cycle. Soar only learns new production rules so it is not capable to perform a large amount of numerical calculations, perform low-level control of motors, or solve optimization problems

EPIC : Like Soar, EPIC is a production system in which the productions have a much larger grain size than Soar productions. The productions which implement executive knowledge do so in parallel with productions for task knowledge. It links high-fidelity models of perception and motor mechanisms with a production system. It comprises a 'Cognitive Processor' (comprising 'Working Memory' and 'Production Rule Interpreter, a auditory processor, a visual processor, an oculomotor processor, a vocal motor processor, a tactile processor, and a manual motor processor. All processors run in parallel. Yet, it does not have any learning mechanism. It is weakly addressed for perception, action and anticipation.

ACT-R : The ACT-R cognitive architecture focuses on the modular decomposition of cognition and offers a theory of how these modules are integrated to produce coherent cognition. Each module processes a different kind of information for example the vision module determines objects, the declarative module is for retrieving information from long-term memory etc.. The production system coordinates the operation of the other four modules by using the module buffers to exchange information. ACT-R operates in a cyclic manner where on each cycle the production system requests information from the modules by supplying constraints to it. The module places then a chunk which satisfies the given constraints in its buffer. However, the content of any buffer is limited. The information of the buffers is then read, interpreted and new information may be requested or stored in those buffers. Declarative knowledge effectively encodes things in the environment, while procedural knowledge en-codes observed transformations. A central feature of the ACT-R cognitive architecture is that these two types of knowledge are tuned in specific application by encoding the patterns of knowledge. It uses two sub modules, one for object localization and associated with the dorsal pathway, and the other for object recognition and associated with the ventral pathway. Only one production is selected to fire in any one cycle. It is weakly addressed for perception, action, anticipation and adaption.

ICARUS : ICARUS takes the physical symbol system hypothesis to its logical conclusion by working only with 'symbolic operators' residing in an abstract 'problem space'. It includes separate modules and it differs from predecessors by positing that cognition is grounded in perception and action, conceptual knowledge is distinct from skills, both are organized in a hierarchy, and short-term elements are instances of long-term structures. It is weakly addressed for perception, action, anticipation and adaption.

ADAPT : ADAPT is somehow a 'cognitive architecture specifically designed for robotics'. It is a cognitive architecture, which is based on Soar but also adopts features from ACT-R (such as long-term declarative memory) and EPIC (all the perceptual processes fire in parallel) but the low-level sensory data is placed in short-term working memory where it is processed by the cognitive mechanism. ADAPT has two types of goals: task goals (such as "find the blue object") and architecture goals (such as "start a schema to scan the scene"). It also has two types of actions: task actions (such as "pick up the blue object") and architectural actions (such as "initiate a grasp schema"). So it is strongly addressed for perception and action although it is weakly addressed for anticipation and adaption.

<u>Emergent</u>

AAR : The main idea behind AARs and behavior-based systems is to avoid a decomposition of the system into functional components by using subsumption. Subsumption means that at the bottom are simple whole systems that can act effectively in simple circumstances; layers of more sophisticated systems are added incrementally, each layer subsuming the layers beneath it. AAR is strongly addressed for perception and action it has autonomy yet it is weakly addressed for motivation while anticipation and adaption are not addressed at all.

Global Workspace (GW) : The GW architecture is a biologically plausible brain-inspired neural-level cognitive architecture in which cognitive functions such as anticipation and planning are predicated on two key observations: (i) The 'simulation hypothesis', that a person's thoughts comprise internal simulations of his interactions with his environment; (ii) The 'global workspace model', whereby the brain's massive parallelism is simulated using a 'global workspace', equivalent to a blackboard architecture.

I-C SDAL : It is a interactivist-constructivist (I-C) approach to modeling intelligence and learning: self-directed anticipative learning (SDAL) Intelligence is consider as a continuous management process that has to support the need to achieve autonomy in a living agent, distributed dynamical organization, and the need to produce functionally coherent activity complexes that match the constraints of autonomy with the appropriate organization of the environment across space and time through interaction. This architecture uses the term "explicit norm signals" for the signals that a system uses to differentiate an appropriate context performing an action. These norm signals reflect conditions for the (maintenance) of the system's autonomy.

SASE: SASE conducts the processing as a result of the real-time interaction of the system with the environment including humans. it should be able to develop its own detailed structure according to the task in hand and nothing is not specified (or programmed) a priori and thus, the architecture is not specific to tasks, which are unknown when the architecture is created or programmed, but is capable of adapting and developing to learn both the tasks required of it and the manner in which to achieve the tasks.

DARWIN : There are robot platforms aimed to experiment with developmental agents. These systems are "brain-based devices" (BBDs) which exploit a simulated nervous system that can develop spatial and episodic memory, as well as recognition capabilities through autonomous experiential learning. These control systems have been designed following a combined connectionist and structuralist approach, whereby arrays of neural units are grouped into a number of different modules, each of which is dedicated to a specific operational task.

Hybrid

HUMANOID : the Humanoid cognitive architecture is a control system for a robot has three levels of perception and action. Humanoid's low, mid, and top levels deal with behavioural sensing/responses, coordination, and planning, respectively. Where appropriate, these modules have access to a global knowledge database and an 'Active models' system, comprising long-term memory and working memory, respectively. The Global knowledge database can be updated by a 'Learning module', while the Active models unit is controlled by an 'Execution supervisor'. Perception and action are coordinated at the top level by means of a 'Dialogue manager'

Cerebus : Cerebus combines the tenets of behavior-based architectures with some features of symbolic AI (forward- and backward-chaining inference using predicate logic). It represents an attempt to scale behavior-based robots without resorting to a traditional central planning system.

Cog: theory of mind : Cog is an upper-torso humanoid robot platform for research on developmental robotics. The driving force behind this architecture is the theory of mind which decomposes the problem into sets of precursor skills and developmental modules, albeit in a different manner. The Theory of mind emphasizes independent domain-specific modules to distinguish: 1) mechanical agency; 2) actional agency; and 3) attitudinal agency; the behavior of inanimate objects, the behavior of animate objects, and the beliefs and intentions of animate objects.

Kismet : Kismet is a robotic head which can express a lot of different human like emotions. It has 21 degrees of freedom for controlling the head orientation, the gaze and its facial features . It additionally has a wide-angle binocular vision system and two microphones. It was designed to engage people in natural expressive face-to-face interaction. Kismet has two types of motivations: drives and emotions. The drive subsystem regulates Kismet's social, stimulation and fatigue related needs. Like in an animal that has a level of hunger, each drive becomes more intense until it is satiated. These drives affect Kismet's emotion system, which contains anger, disgust, fear, joy, sorrow, surprise, boredom, interest, and calm. These emotional states can activate behaviors. For example, the fear emotion can induce the escape behavior. Kismet's cognitive architecture consists of five modules: a perceptual system, an emotion system, a behavior system, a drive system and a motor system. It is strongly addressed for motivation, perception and action although it is not addressed for anticipation and adaption.

2.5.3 Cognitive Agents

The purpose of these systems is to assist and to "improve" humans' decision making. It is far more obvious that cognitive agents should have components that will be "equipped" with cognitive capabilities like those of the human minds in order to comprehend information and be "adaptive" and "autonomous". Key elements are the natural language which humans use for their communication as well as the ways they store, retrieve, and use common sense knowledge.

Several cognitive psychologists have tried to describe the way we comprehend the information and how our cognitive functions operate and interacting together while several other researchers had tried to examine how humans make inferences and how these inferences are produced under a certain contexts. The results of this research are used for the construction of cognitive agents that should be able to learn, and be able to improve from their past interaction with the user by offering personalized solutions. Some characteristics - capabilities that the cognitive systems should have are the following:

Sensing: to perceive the information from the environment or in other words to recognize, organize, and make sense of the sensations that receives from environmental stimuli in way similar to humans.

Learning: The primary goal of machine learning is to derive general patterns from a limited amount of data. It has the ability to modify itself when exposed to more data is dynamic and does not require human intervention to make certain changes (autonomous). The learning task is to predict some additional aspect of an input object (adaptive) but also to remember previous interactions (iterative).

Knowledge: this is the structure of formal knowledge base that represents and manipulates knowledge base as a dynamic concept network simulating human knowledge processing

Comprehension: is the existence of explainable models and the ability for the systems to answer how-why questions. This is a simulation of how humans explain decisions and behavior to each other and helps machines to "understand" the context and environment in which they operate. This component allows them to characterize real world phenomena making them more autonomous and adaptive but also "keen" to reflect and learn.

Decision Policy and Decision Making: Humans, when they are triggered, generate arguments to accept or decline a conclusion, this procedure is native to human reasoning.

But it is also essential for these systems because they perform in open and dynamic environments mostly. This means that some of the properties of our problem domain can change when new information is acquired. The states of the environment are affected by actions that have been performed and/ or by other actions which are blocked from materializing at a particular stage or time in the flow of change. Combined with learning we can plan or modify actions.

Interactive: Cognitive systems should interact bi-directionally. It should understand human input and provide optimum results (communication and action).

Under this frame we can now examine the extent of which an agent can be consider as Cognitive system. The agents are embedded with a knowledge base and probably a goal named KB::KR (Goal::KR, Plan::KR) and represents the targets and the steps needed to be taken to reach those targets. In other words, for each agent's state there is a full "description" of the environment for that state, a plan with a goal and a set of actions that are "linked" together while there are specific knowledge types supporting various others functionalities like reactive responses , temporal reasoning etc. . In that way the problem is represented in a "language" that the agent can reason and it knows the criterion(s) of success the acceptable solutions, goals, possible preferences, tradeoffs, and possible time responses.

For example if the agent was a robot that picks boxes from a warehouse then it would know all the possible states of the warehouse (empty, full, half-empty etc.) where the parts are positioned or stacked in the bin in an organized, predictable pattern, so they can be easily imaged and picked or the parts are positioned in the bin with some organization and predictability to help aid imaging and picking or even the parts are in totally random positions in a bin, including different orientations, overlapping, and even entangled, further complicating the imaging and picking functions. For all possible cases, the robot would have the information to sense, select plan, goal(s), action(s) and even behavior, through the component Profile, in order to reach the target. Regarding the behavior, each agent is "equipped" with a set of profiles that allow the agent to follow a series of rules something like a decision policy, and to make use of the component Capabilities providing reasoning capabilities in order to select the next plan. Furthermore there is a transition component that increment the agent's state while contains in memory the previous state. So the agent, apart from a decision policy and reasoning capabilities has memory that records its previous states.

The design of the agent follows an architecture design that promotes mind-body separation, knowledge representation independence, reasoning system independence, cognitive independence and control independence. The mind-body separation separates the domain reasoning from the physical interface (in humans terms is like separating the brain from the body) so that domain reasoning could be used in a different application with minimum reconfiguration. The knowledge representation independence handles different types of information which have different types of representations and reasoning mechanisms which are supported by the reasoning system independence. The cognitive independence separates the internal cognitive functions and defines a common interface with knowledge representation and reasoning. Finally, the control independence allows the agent to reason of what action or actions should be executed.

In brief, these agents can interact with their environment in order to perform a task or tasks optimally. The agent can plan or react by changing plans in case the environment shows changes and can decide new goals (adaption). It has a kind of memory and it can make predictions by using temporal reasoning while it is flexible by using profiles. Finally, it demonstrates a simulation of the use of some cognitive functions like perception, learning (by using knowledge), reasoning, memory and working memory (mental model as some cognitive psychologists mention like to call). Under this frame these agents can be considered in a way as "complete" cognitive systems.

Lin and Carley (Lin & Carley 1993) argue for two types of agent behavior: reactive and proactive A proactive agent uses an approach similar to introspection thus it perceives the information then makes a decision which decision has taken into account the perceived information. On the other hand, a reactive agent perceives but it will not make a decision, this process will occur when it is requested by the user. Thus, what makes agents so powerful is their proactive behavior. Proactive behavior is also seen as an essential characteristic of autonomous and semi-autonomous agents (Norman, 1994: 68-71). Other researchers (Yorke et al 2012: 1250004) split proactive behavior into two types. The first type, called *task-focused proactivity*, involves providing assistance for a task that the user either is already performing or is committed to performing;. And the second type of proactive behavior, called *utility-focused proactivit* , involves assistance related to helping the user generally with her set of tasks, rather than contributing directly to a specific current task.

2.6 Issues to overcome

Artificial intelligence is not the only field of science that tries to model the human intelligence and behavior. For example, the science of Economics tries to establish methods for modeling human behavior and interaction in markets and other economic settings. This procedure is rather complex since humans' decisions are guided by conflict concepts like rational and un-rational or self-interest and fairness and equity. Additionally, humans' cognitive abilities are limited and they can vary significantly across different individuals. For this reason, the methodology of modeling uses the assumption that all decisions are made on perfectly rational manner. The research on individual decision making was enriched by using the research in the psychology field, so a new field of research has emerged which Thaler, an American economist, named behavioral decision research (BDR) (Committee, Nobel Prize, 2017). This kind of research is strongly influenced by the psychological approach to the study of perception, since during the decision processes we have illusions like the optical illusions that occur in visual processes. Human choices often "replace" the optimum solutions with acceptable solutions that satisfy a set of self-imposed constraints. This is an element that shows the need of alternative models and additional specific predictions. The use of psychology in economics had "created" the field of behavioral economics, an important tool for economics to understand and predict the human behavior. According to Thaler, there are three aspects of human psychology that influence economic decisions. These are the cognitive limitations, self-control problems and social preferences. In other words, Thaler emphasizes, among other things, the role of cognition and the mental states and how these affect the humans' decisions.

But even the concept of human intelligence itself is rather difficult to be defined which seems that is "connected" with the cognitive functions. The answer of what is human intelligence is not easy. This term has been an important and controversial topic and despite the substantial interest in the subject, there is no "clear" definition about what components make up intelligence. In addition to questions of exactly how to define intelligence, the debate continues today about whether accurate measurements are even possible. A rough and a rather plain definition of the concept is given by Wikipedia which states (3) *"**Intelligence** has been defined in many ways, including: the capacity for logic, understanding, self-awareness, learning, emotional knowledge, reasoning, planning, creativity, and problem solving. More generally, it can be described as the ability to perceive or infer information, and to retain it as knowledge to be applied towards adaptive behaviors within an environment or context."* , even this and rather plain definition just enforce the argument about the complexity of the concept. There have been many models of intelligence. Three of these models are:

The Carol's three-stratum model (Caroll 1993) in which the intelligence comprises a hierarchy of cognitive abilities comprising three strata, the Gardner's theory (Patanella , Ebanks 2011) of multiple in which intelligence comprises multiple independent constructs, not just a single, unitary construct and the Sternberg's triarchic theory of intelligence (Sternberg 1985) in which intelligence comprises three aspects: creative, analytical, and practical.

All the above theories indicate that there is a strong relationship between cognition and intelligence and it is rather safe to assume that intelligence is depended on cognitive functions. Without cognition abilities, one cannot show his intelligence, cognition seems to be a requisite element for an intelligent acting and behavior.

The above examples demonstrate the difficulty of the definition of what we call "intelligence" but in any case intelligence and cognitive functions go together. For instance, the common spreadsheets we use in our personal computers can apply mathematic formulas that require a kind of human intelligence, yet we do not consider these spreadsheets as intelligent. Even more sophisticated machines like IBM's Deep Blue may demonstrate superiority in a certain field like chess, but we definitely exclude them on the real concept of intelligence. Because playing chess is not enough, the definition of intelligence has a large number of characteristics that form a very broad domain. In other words, these machines are capable of doing a limited number of functions and operations. For the machines the need for cognition abilities or intuitive physics and psychology, the terms that are used by some researchers is far more than critical, it is essential.

On the other hand, IBM's Watson seems to take the discussion a step ahead. The idea behind Watson is structuralism (simulation of the structural characteristics of the biological neural networks in the human brain by breaking down processes into the most basic components) and the use of natural computation by modelling somehow the human brain. The system can play Jeopardy! in real-time contests against human contestants, adhering to the very same rules. It uses a range of "natural" intelligent skills, including natural language interpretation, command of many facts about our world, and the ability to match sometimes tricky or puzzling questions to correct answers, under these terms it behaves "intelligent". But is this the truth? At this point we have the well-known question "which came first: the chicken or the egg ?" Is the Watson intelligent or the team that built Watson? Watson uses a set of mathematical techniques, an intelligent software architecture that enables massive parallelization, and it has received years of testing and improving by human experts. Additionally, we know that Watson was re-tested, re-trained, and re-developed to perform on a different, yet related, QA tasks in other fields apart Jeopardy!. It cannot do it by itself that would be true intelligence; instead Watson's team once again re-developed a system using the architecture in order to perform computational tasks on another problem. Under this frame we cannot call Watson "intelligent". In other words the idea behind natural inspiration has two issues to overcome. The first is brittleness and the second is transparency. The machines are brittle because they do not have the cognitive procedures to form the context of the various situations that occur in reality. Therefore machines break very often when they have to deal with a different condition comparing with the training examples that they have seen. Additionally, the machines are not transparent which means that they can make decisions yet they cannot provide explanations about these decisions, they function as black boxes. Furthermore, their cognitive functions are limited and work for specific tasks or environments. Besides, how can we model something efficiently although we do not know exactly how it works? We need more information about the human brain and its processes or how human cognitive functions operate. Other researchers have been tried to understand how our cognitive system works or how we argue and others how we behave under some conditions and circumstances. We just speculate and we make some assumptions based on research and experiments which very often provide conflict results, for example is memory a series of processes or a location? How perception, memory and learning work together? What we see first, objects or space? Can humans argue?

A crucial point for producing more autonomous machines besides intuitive physics and psychology is the existence of explainable models. If we want to design, and implement intelligent agents that are truly capable of providing explanations to people, then it is fair to say that models of how humans explain decisions and behavior to each other are a good way to start analyzing the problem. This is the functionalism approach that simulates the functional processes of logical human thinking. This approach will help machines to "understand" the context and environment in which they operate, and over time build underlying explanatory models that allow them to characterize real world phenomena. The key point for the machines to show some intelligence is creation of the ability for them to answer how-why questions. This creation can be helped by the mental models and schemas that are used in cognitive psychology. Mental models are knowledge structures that individuals construct to understand and explain their experiences. The models are constrained by the individuals' implicit theories about these experiences, which can be more or less accurate. For instance, Walter Kintsch provided the Construction-Integration (CI) model (Kintsch, W. 1998), which is a psychological model of text comprehension, a requisite element for a cognitive system. As machine reading and comprehension technology continue to develop, computers will be able to study and process large amounts of text quickly and most important efficiently. The machines could then provide people with very specific details from the information in an easy, understandable way.

The schemas are organized plans that create a meaningful structure of related concepts, guide attention and behavior, and influence the reconstruction of memories. Schemas have several characteristics for example schemas can include other schemas, concepts, attributes etc. Humans use two types of learning, the inductive and the deductive learning. Through inductive learning we can generalize knowledge from a few specific examples. The idea is that we look for patterns that explain the common characteristics of the examples. On the other hand deductive learning allows us to make statements that are entailed by facts that we know. We apply deduction in order to obtain generalizations from a domain theory, a solved example and its explanation. The research (Ahn, Brewer, Mooney 1992) provides several models, two of them are the Similarity-based learning model (SBL) which is based on the inductive learning and the assumption that concepts are formed by extracting similarity across multiple examples and explanation-based learning (EBL) model is based on deduction learning and the role of prior knowledge in learning new concepts.

As Ann etc.all 1992 state in their research the type of stimuli, the learner's domain knowledge, the distinction between explanatory and no explanatory information and the fictitious correspondence among examples are the crucial variables for the SBL model. For humans is rather easy, in knowledge rich domains, to create a schema even from a single instance yet. This theory does not provide a clear account of how a schema could be learned from a single, specific instance. This is a fundamental problem for approaches that assume that generalization occurs only by selecting common information across multiple examples. The EBL approach uses deductive learning and the existing prior knowledge, which provides explanations of why an example belongs to a concept, in order to create a new schema and therefore its possible generalization. The method by using the domain knowledge explains why an instance belongs to a concept and provides justified generalizations avoiding fictitious correspondence by removing the irrelevant features.

The authors have presented the GENESIS program as an example of an EBL system that improves its abilities to explain observed behavior. The GENESIS improves its performance by learning plan schemata from specific observed schemata. The example showed that the schema will be constructed from three types of knowledge: a) from the information that would be acquired during the trial b) from knowledge about schemas similar in content to those that are held in the task and c) from the information related with the task's goals.

This approach has also some withdraw backs. It needs sufficient knowledge about the domain and to construct an explanation while there is no mechanism for making use of similarities across multiple examples. The authors' experiments have shown that the participants, in knowledge-rich domains, by using their previous knowledge through EBL mechanism could acquire a schema even from a single instance. In particular in cases that this knowledge was experimentally provided there was little use of EBL while in cases that the task had demanded the use of the new gained knowledge the EBL had been applied. In cases we had optimum repetitions the SBL had provided schema learning, yet the SBL approach had not provided correct justifications for explanatory constraints. It seems that humans use both SBL and EBL mechanisms.

The need for intuitive physics, has been mentioned previously, is more obvious when we speak for visual perception. Basically, the role of the human eye is to convert light into electrical signals called nerve impulses that the brain converts into images of our surroundings. Researchers at the University Of Pennsylvania School of Medicine (University of Pennsylvania School of Medicine. 2006) estimate that the human retina can transmit visual input at about the same rate as a common local area network system like Ethernet . However we do not perceive the world exactly as our eyes see it. Instead, our brain actively tries to make sense of the many stimuli that enter our eyes and fall on our retina. Then it processes the visual stimuli, giving the stimuli meaning and it interprets them. In other words perception occurs when a perceptual object is created in the brain and reflects the properties of the external world (for example a car that moves away).

It is more than clear that cognitive computing needs the existence of multiple cognitive functions that will allow the applications of intuitive physics and psychology, the explainable models that will provide answers to how and why and the reading comprehension that provides ability to "read", understand, process, and recall text. There is a fifth element, the prediction. Humans use the brain to create predictive models explaining away expected changes, including those made by planned eye-movements(Edwards, G., Vetter, P., McGruer, F. et all 2017) . To recreate an optical illusion, the researchers(Edwards, G., Vetter, P., McGruer, F. et all 2017) had volunteers look at two stationary flashing squares. To the observer, it appeared as though one square was moving between two locations where, in reality, it was just two separately flashing squares. The volunteers were asked to move their eyes during the flashing and fMRIs took scans of the visual cortexes of their brains as this happened. The results revealed that during these flashes, the visual cortex feedback updated to a new predicted coordinate. The fMRI scans also showed that the brain rapidly adjusts predictions each time the eyes move. For the humans, their ability for predictions are based on some attributes like knowledge, experiences, hunch, perception, reasoning and other cognitive processes which for every have somehow unique characteristics. For example, in our society when someone introduces two people, we assume (or otherwise we predict) from them to follow some particular protocols of behavior (state names, give some details about them, allow handshake, helping in initiating a conversation, etc.) Some of the predictions we make cannot affect the future while others can define it.

Additionally, when we predict something we are also taking account the outcome of our prediction. This is not always true, but for most of the times it stands. On the other hand, the machines do not “predict” in that way at least the “old school” does not. The idea behind the “old school” is that the machines, through their perception, receive the data and reasoning about them making a prediction without taking an account the outcome of this prediction. This is of what we call extrospective prediction. However, the new models of prediction use a method, which called introspection (observation of states, a tool that supports functionalism) and it is contrasted with external observation that we saw before. Introspection is familiar concept in cognitive psychology area that deals with human self-reflection and takes into account the announced predictions. Which is the best model? Although, science tries to create and define models whose aim is to be as generic as possible the answer is not simple and it is depended on the situation.

An example of how these two models work is the following scenario. Suppose there is a country in the Mediterranean Sea which has established an agency that has two goals. The first goal is to record the number of tourist flows from abroad and if it is possible to increase them while the second goal is to record the unemployment rate and if it is possible to decrease it by promoting the hotels and other tourist corporations to recruit staff. The agency in order to be prepared for following year tries to predict a) the number of people that is expected to visit the country from abroad and b) the number of unemployed that will find a job in tourists operations.

Using the extrospective prediction model the agency (predictor) chooses for the period from September to March to know the number of reservations made by people from each foreign country (the agency chooses a distribution D to a sample of states S in order to describe the current status of reservations) because it believes that this number of reservation in this time frame is a good indicator that will "reveal" the total number of tourists (final status). Using this information the agency expresses a hypothesis H (1-ε) accurate and estimates the final number of people that will visit the country. The same does with the unemployment, for the same period retrieves the number of recruitments that tourists operations have made or the recruitment ads that they have posted, express a hypothesis H (1-ε) accurate and estimates the final number of people that will be hired by the tourist operations. Let's assume that in this case that the news is good and the results show an increase in the tourist flows and an increase in recruitment in tourist operations. The predictions are not announced and the results are known only inside the agency.

Instead of using the extrospective prediction model the agency uses the introspective prediction model and for the same period retrieves the corresponding numbers of reservations in order to produce the current status of reservations and the same does with recruitments in tourists operations. Like in previous model the agency establishes a hypothesis and creates a "first" result which as we assumed before has positive sign. The model "needs" the agency to announce these first results, so they are announced during this period of time in the country and the foreign countries. But their announcement helps the agency to create a "trend" make more people want to visit the country and therefore to make more reservations. Additionally the tourists operations knowing the results and in order to be prepared need more staff, so they post recruitment ads or hire. The announcement produces an outcome which the model takes into account. Now the numbers have change and the model re –estimate them including the new numbers (old numbers + new numbers) in order to produce the final estimation.

But what if the results were not so good for the agency? Probably, in this case, it will follow the extrospective model to predict the numbers without announcing any prediction since the announcement would form a bad trend for the tourism and unemployment.

The evolution in the Cognitive Computing field would be the creation of an "intelligent agent" that with some initial knowledge as well as an ability to learn and after sufficient experience of its environment, it could become effectively independent in a vast variety of environments.

Chapter 3
Related Research

3.1 Introduction

As it has been mentioned Cognitive computing aims to “connect” humans and machines by using cognitive technologies which provide computational power combining with embedded “human intelligence”. This power leads to a better processing reaction and respond while it provides insight from huge amounts of data. Machine coaching is one of the cognitive systems’ tools so they can “learn” from incoming data and from their interactions with humans. This collaboration of cognitive systems and humans opens new possibilities to produce better products taking advantage of the combination analytic capability and encyclopedic knowledge of computers and the intuition, creativity, moral and expertise of humans.

Cognitive computing has grown in the last few years, increasing the research and commercial interest in the topic. One of the reasons is the use of Natural Language Processing that has expanded the methods for humans to engage with technology. This in turn, reduces the effort to complete a task using reasoning capabilities and by exploiting context, or allow voice interaction when traditional methods are not available or inconvenient.

This expansion has arrived to a number of commercial products based on these technologies, such as Amazon’s Alexa, Google’s DialogFlow, Microsoft’s Luis, IBM’s Watson, Facebook’s Wit and Apple’s SiriKit. Conversational agents have evolved from rather simple systems that do their best effort to maintain a conversation, to personal assistants that understand users’ requests and perform tasks on their behalf. Once again Siri, Google Assistant or Microsoft’s Cortana are a few enlightening examples. In this chapter we will review some of these technologies, starting from the simplest to more complicated, in order to understand the approach, which it is required for designing a system capable of understanding a fairly representative set of competency questions in the application domain, and at the same time being able to handle small talk –as Siri or Alexa do.

3.2 Fuzzy Cognitive Maps

Fuzzy Cognitive Maps (FCM) is symbolic representation for the description and modeling of a system. They consist of concepts, that illustrate different aspects in the behavior of the system and these concepts interact with each other showing the dynamics of the system. In other words, FCM models a system as a collection of concepts and causal relations among concepts. The concepts are represented by nodes and their relations by directed links. The nodes represent descriptive behavioral concepts of the system and the links represent cause-effect relations between the concepts. By describing the behavior of a collection of concepts, FCM provides a more flexible and natural mechanism for knowledge representation and reasoning which are essential to intelligent systems. The concepts might be facts, actions, trends, restrictions, measurements, etc. depending on the system's utility, goals and nature while the nodes are characterized by an activation state value, *Ai* that represents the membership degree of the concept, at a specific time instance. (figure 4)

Fig. 2 Fuzzy cognitive map

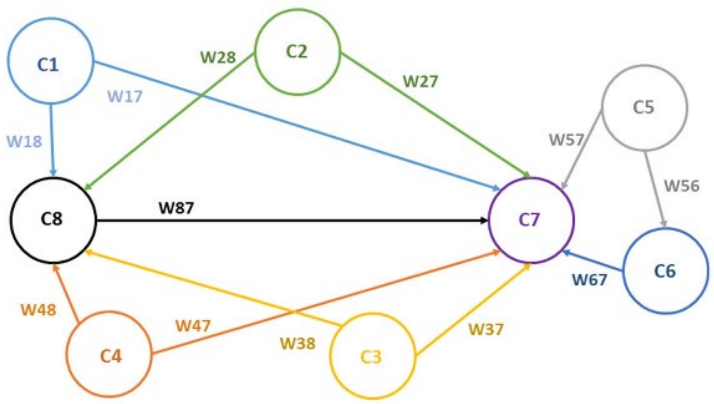


Figure 4 (source Mpelogianni V.·Groumpos P 2018:180)

The human experience and knowledge of the operation of the system is used to develop the Fuzzy Cognitive Map, as a result of the method by which it is constructed, i.e., using human experts that know the operation of system and its behavior in different circumstances. Therefore experts with proper expertise in the scientific or industrial area of the system to be modelled are required for building an FCM. This group is responsible to define the number and the type of the concepts which will be represented by the FCM model.

The relationships between concepts are described using a degree of causality and not the usual binary logic. Experts describe this degree of influence using linguistic variables for every weight; so Weight Wij for any interconnection can range from -1 to +1. A positive weight expresses a direct influence relation whereas a negative one defines an inverse relation between two concepts. Anything between *zero* and *-1* or *zero* and *+1* correspond to various fuzzy degrees of causality.

There are three possible types of interaction. Interaction can express:

- either positive causality between two concepts (Wij>0) when the increase on the value of the *i* concept causes an increase of the value of the *j* concept
- negative causality between two concepts (Wij<0) when the increase on the value of the *i* concept causes a decrease of the value of the *j* concept
- no relationship between two concepts (Wij=0)

The lack of knowledge of the system, the dependence on experts, the disability of self-learning, the definition of the causality, the calculation equation, the ignorance of time factor, and the use of the sigmoid function-interpretation of the results are some of the reasons that this kind approach cannot be considered as cognitive system (Mpelogianni & Groumpos 2018:176)

3.3 Expert systems

These systems are designed to provide answers and solution in a limited domain like a human expert in that domain. They are embedded with a knowledge base and they are aware about the targets and the steps needed to be taken to reach those targets. They can interact with their environment in order to perform a task or tasks optimally. These kinds of system or agents can plan or react by changing plans in case the environment shows changes and can decide new goals (adaption). It has a kind of memory and it can make predictions by using temporal reasoning while it is flexible by using profiles. Finally, it demonstrates a simulation of the use of some cognitive functions like perception, learning (by using knowledge), reasoning, memory and working memory (mental model as some cognitive psychologists mention like to call).

Summarizing, for each system's state there is a full "description" of the environment for that state, a plan with a goal and a set of actions that are "linked" together while there are specific knowledge types supporting various others functionalities like reactive responses , temporal reasoning etc. . In that way the problem is represented in a "language" that the agent can reason and it knows the criterion(s) of success the acceptable solutions, goals, possible preferences, tradeoffs, and time responses.

A key element of the expert systems is an explanatory component that should be able to provide a high level explanation of its results making the system able to explain actions or/and results in a manner similar to the human expert. An idea how these agents operate is given by Kakas, A., Moraitis, P., & Spanoudakis, N. I. 2019 where there is a review of a general methodological approach for developing "decision making" applications of argumentation and the brief report about two real life applications in two domains: an eye-clinic assistant and data access and sharing assistant.

More specific, the authors argue for the importance of argumentation over a wide range of problems such as the debates on online social interaction settings or in the automating legislation. Following, they state some argumentation systems like CaSAPI, DeLP, TOAST and Gorgias that support the study for this kind of problems and their applications while they indicate that decision problems occur in dynamic and incomplete or uncertain environments by giving several examples of problem which support their claim (legal problems, medical problems etc). The researchers' approach for developing applications based on argumentation has three, as they state, challenges. First is the acquisition and elicitation of the requirements that need to be carried out at a high level akin to the natural cognitive level of those whom the application is built. Second, it is the system's development that must be executed incrementally in order to adapt new or changed requirements succeeding a continuous learning process. This process is linked with this method of development. Third, the system must explainable to people, something which, now, is required by law in European Union. The researchers support that their approach, which follows the preference –based structured argumentation frame of Logic Programming with Priorities and its Gorgias implementation, remedy these kinds of issues. The Georgias argumentation framework uses the Modus Ponens argument scheme. The authors talk about beliefs (conditions that are defeasible) object-level arguments (when the claim of an argument is a literal on an option or belief predicate) attacks (when an argument attack each other supporting contradictory claims) priority arguments (give relative strength between arguments) and their role that is to tighten the attack relations. Furthermore they describe the terms acceptability, composite arguments, attack relations admissible arguments and what consists a "good" solution and a "best or optimal' solution. They use the on line assistant example to provide the argumentation framework of the Gorgias and they give the central algorithm for argument generation. The algorithm "translates" the arguments in an automatic way and generates the rules using a hierarchy manner make them invisible to the user. Finally, they introduce a new tool, the Gorgias-B, a novel human-machine interface that supports their proposed approach.

These agents, through their design and architecture, show a level of skills and abilities that could characterized them, in some level, as a cognitive systems. However as it has been stated , the idea is that cognitive systems should be able to learn, and be able to improve from their past interaction with the user offering personalized solutions and giving the impression of “self –reflection”. These agents seem to be designed to achieve goals and plans for specific domains and not to provide personalized solutions. The knowledge seems to be something like a set of skills (knowledge to do something), although it has the ability for new skills that would applied in the domain it cannot learn something new by itself. The agent must re-design in order to obtain these new skills. It can adapt but in a different way, it has to reorganize the states of the domain.

Furthermore, cognitive systems are designed in a way that simulates somehow the human’s perception. Human perceive using among other things past experiences, knowledge, emotions and reasoning. Cognitive systems try to approach the same protocol, so they have components (learning, knowledge, comprehension, decision policy, decision making) that are combined and work together for an optimum “perception”. On the other hand the agent perceives states of the domain that had already been “described” as closest as possible.

Another difference is how these two models act. The cognitive system by using analysis and evaluation acts in a “casual” or not “predefined” manner.. Through training they can learn how to apply “cognitive abilities” producing in that way a sort of human intelligent behavior. On the other hand these kind of agents act according the states of the domain following a predefined certain plan for that state of the domain.

For cognitive systems, cognition plays the central role for the representation of the environment and its behavior, provides adaption, interaction, knowledge and experience, tools that make cognitive systems to increase during time their performance. This does not work for these systems since that they are specified by their environment that determines what is meaningful for them and whose development seem to be depended on skills construction rather than knowledge acquisition.

3.4 Question Answering Systems (QA)

Question answering systems (QA) are set in a research area that combines topics and approaches from different, but related, fields which are Information Retrieval (IR), Information Extraction (IE) and Natural Language Processing (NLP). Their main purpose and function is to provide accurate answers from a vast variety of data available on corresponding databases or on the web. Additionally, QA systems should have the components which allow the users to ask the questions-this implies that the system must understand the questions- and obtain the answers in their native language.

The question answering system can work either in closed domains or open domains. Closed domain question answering deals with questions under a specific domain and can be seen as an easier task because NLP systems can exploit domain-specific knowledge frequently formalized in ontologies. An example of this case is the Task-oriented dialog agents. These are designed for a particular task and set up to have short conversations (from as little as a single interaction to more extended) to get information from the user to help complete the task.

The open-domain question answering deals with questions on wide area of topics and can rely on general ontologies and world knowledge. An example of this case is the chatbots. They are systems designed for mimicking written or spoken human speech for the purposes of simulating a conversation or interaction with a real person. They can offer extended conversations, rather than focused on a particular task like hotel booking.

The first instance of a conversational agent based on question answering system was born in 1966: ELIZA was an agent that simulated a psychiatrist and rephrased user input using a natural language processing technique. It was a Rule-based system which was trained on a predefined hierarchy of rules that govern how to transform user input into output dialogue or actions. Eliza first scanned the input text for keywords, assigned each keyword a programmer-designated rank, decomposed and reassembled the input sentence based on the highest-ranking keyword, and if it encountered remarks that didn't match any known keyword, prompted the user to provide more input (Sanjay D. , Vaishali S. 2013:418) .

Some years later we had the publishing of LUNAR for answering questions about the geological analysis of rocks returned by the Apollo moon missions. In 1976 we had the introduction of TRIPSYS(HWIM). It was called HWIM (for "Hear What I Mean) and it could understand speech questions. Another contribution came from the Text Retrieval Conference (TREC), it is a yearly held conference which provides large scale infrastructure and resources to aid research in the field of information retrieval. It has been researching the management and the queries of large volume of data in open domain question

answering from unstructured data sources (Sanjay D. , Vaishali S. 2013:418).
The first TREC evaluation campaign provides a list of 200 questions and a document collection. The answers were known to be present in the collections. The maximum lengths of answers were allowed to be 50 or 250 characters. In TREC-10 in 2001, a new complexity with respect to answers. The lengths of answers were reduced to 50 words.
QA systems have developed over the past few decades until they reached the structure that we have nowadays, some well-known systems are the Watson (IBM), Alexa (Amazon), Siri (Apple) and Cortana (Microsoft).
A QA system usually has a research domain which can vary between small sets of documents locally stored, to a vast size of documents, that can be set in networks managed by companies, enterprises or organizations. Even more, the research can be conducted in the Internet, in structured databases or in sets of documents in Natural Language. This effort aims to the construction of an accurate answer to the question posed by users, by consulting the QA's knowledge base. This research deals with a wide range of question types, including: facts, lists, definitions, hypothetical, semantically limited, language-independent questions (cross-lingual questions). Prior knowledge of the type of expected answer helps QA systems to extract accurate and correct answers from the collections of documents that make up their knowledge domain. Apart from that, there are several challenges that QA systems are facing. Some of them are described in the rest of the section.
In a natural language, the same meaning can be expressed in different ways. QA systems, in order to respond and provide accurate answer, use a model-vocabulary that rests in their knowledge base and contains all the words and synonyms that can be modeled. Yet, this model cannot contain all the different terms that can refer to a certain entity. Therefore, if a user states a question using words from a vocabulary different from the one which the QA uses in its knowledge base, then we have gap between these vocabularies which is described by the term "lexical gap".
The identification of each word in the question is a critical point that minimizes this gap and increases the proportion of questions that can be answered by a system. Some techniques to remedy this problem are the normalization by using a stemming procedure or even a similarity function (Höffner K. , Walter S. , Marx, E., Usbeck R. , Lehmann J. Ngonga Ngomo A. 2016 :7) .
A second challenge is ambiguity (Höffner K. , Walter S. , Marx, E., Usbeck R. , Lehmann J. Ngonga Ngomo A. 2016 :9) which is the phenomenon of the same phrase having different

meanings; this can be structural and syntactic or lexical and semantic. But there are also some other issues such as homonymy, where the same string can refer to different concepts (bank- the company and bank- the river) or polysemy, where the same string refers to different but related concepts (as in bank as a company vs. bank as a building). We distinguish between synonymy and taxonomic relations such as metonymy and hypernymy. In contrast to the lexical gap, which impedes the recall of a QA system, ambiguity negatively effects its precision. In other words ambiguity is the flipside of the lexical gap.

Named Entity Disambiguation (NED) is the process of selecting one of multiple candidate concepts for an ambiguous phrase. We differentiate between two types of disambiguation based on the source and type of information used to solve this challenge. Common context features used are word co-occurrences, such as synonyms, hyponyms, POS-tags and the parse tree structure. More elaborate approaches also take advantage of the context outside of the question, such as past queries of the user.

Another challenge is the distribution of knowledge (Höffner K. , Walter S. , Marx, E., Usbeck R. , Lehmann J. Ngonga Ngomo A. 2016 :12). All the available knowledge is stored in various storage places while it is stored by using different techniques and formats. We have three distinct types :documents , data and hybrid (data and documents). So If a concept information–which is referred to in a query–is represented by distributed resources, then if the dataset search is not efficient it is possible that the information needed for answering the query may be missing because one or more responses of the knowledge bases are not found.

The fourth challenge is the complexity of questions (Höffner K. , Walter S. , Marx, E., Usbeck R. , Lehmann J. Ngonga Ngomo A. 2016 :11) . Simple questions can most often be answered correctly. Factual, list and yes-no questions are easiest to answer. Problems arise when several facts have to be found out, connected and then combined. Queries may also request a specific result order or results that are aggregated or filtered. Furthermore the same question can be expressed in various ways. The QA model should have a model of understanding and processing in order to recognize equivalent questions, regardless of how they are presented. This model would enable the translation of complex questions into a series of simpler questions, would identify ambiguities and treat them in context or by interactive clarification. Another solution could be the following, if the question is one of a series of related questions, then the previous questions and their answers might guide the system on the intentions of the user.

Finally we have the cross-lingual problem. The system should have the ability to answer a question posed in one language using an answer corpus in another language (or even several).This is important because knowledge on the Web is expressed in various languages making the questions and the resources-responses available in multiple languages. Furthermore, there is not a single language that is always used in Web documents. This is a challenge for a QA system because it must handle multiple input languages, which may even differ from the language used to encode the knowledge.

As it already stated there are two types of QA systems, these for functioning in open domain and these for closed domain. An open domain system provides answers to any question and it has a large repository of queries that can be asked. QA systems leverages on general structured texts and world knowledge in their approaches to produce answers while casual users are responsible for posing questions A closed domain system offers answers on some fixed topics. This domain requires a linguistic provision to comprehend the natural language text to provide solution to queries precisely. The difference between open and closed domain QA systems is the presence of domain-dependent information that can be used to better the accuracy of the system.

General we can distinguish three types of systems: the question answering systems, the dialogue systems and the spoken dialogue systems. The question answering systems combines research from different, but related, fields which are Information Retrieval (IR), Information Extraction (IE) and Natural Language Processing (NLP), Artificial Intelligence and machine learning.

The dialogue systems can be chatbots or dialogue agents aiming to serve a purpose or a goal. These systems use a dialogue manager that manages the flow of the conversations. They are restricted to a specific domain so they function on a closed domain. This means that if we apply the dialogue system in a different domain from then the system should be adapted by using new grammar suitable for the new domain. These kinds of systems should have “habitability” and high precision, these are utilized by resolving disambiguation and human clarification only when it is unsure.

The spoken dialogue systems take an input and return a voice answer. They have a speech recognizer and text to speech module making the interaction procedure free of constraints and limitations as regards the expressiveness.

Besides the classification using the domain or the type we have the system’s classification based on the type of questions. Generating answers to users’ queries is directly related to the question type. The classes types are: confirmation questions (yes/no), factoid questions(when/who/where), definition, causal questions (how/why/what), procedural, comparative, with examples and opinionated questions.

On the basis of knowledge source type, a classification of the QA system is done in terms of three categories: documents, data and hybrid (data and documents). In any case the Question Answering systems generally follow a pipeline structure with three major modules namely: Question Analysis, Passage Retrieval, and Answer processing. The questions posed to QA systems need to be parsed and understood before answers can be found. Hence, all necessary question processing is carried out in the Question Analysis module. The input for this stage is the user query and the output is consisted by the representations of the query; this is useful for analysis in other modules. At this level, the semantic information contained in the query, constraints and needed keywords are extracted. The activities in this module includes: parsing, tokenization, disambiguation, internationalization, logical forms, semantic role labels, reformulation of questions, co-reference resolution, relation extraction and named entity recognition.

Passage retrieval is naturally based on a search procedure to retrieve a set of significant candidate responses from a knowledge base. This stage makes use of the queries formulated from the question analysis module, and looks up information sources for suitable answers to the posed questions. Candidate answers from dynamic sources such as the Web or online databases can also be incorporated here. Text retrieval structures split retrieval process in three stages: retrieval, processing and ranking. The processing step involves the use of query analyzers to identify texts in a database. Then, retrieval is done by matching documents with resemblance of the query patterns.

Developing the Information retrieval phase of a QA system is complex because systems deploy different mix of Natural Language techniques with conventional Information Retrieval systems

Answer extraction is a major part of a Question Answering system. It produces the exact answer from the passages that are generated. It does this by firstly producing a set of candidate answers from the generated passages and then ranking the answers using some scoring functions. The answer extraction utilizes various techniques for answer extraction.

Different approaches have been applied. For instance, QA systems relying on linguistic approach were basically built upon a knowledge base for specific domain, which provides an efficient and reliable response for short answers. Answer extraction mechanism from the knowledge base is supported by deep linguistic analysis to identify the relevant answer. In other linguistic approaches, web is used as the knowledge resource for local knowledge base, which not only led to enhanced knowledge within the domain but paved the way for thought of question answering with the integration of the local knowledge base in an open domain too. Some of these earlier systems also relied on heuristic rules to identify question class and applied some NLP techniques. This approach has limitation though, because it has success to the systems having only text documents as their knowledge resource. An additional drawback of this approach is that the construction of proper rules required sufficient amount of training data and time along with skillful human effort.

Statistical approach is most likely to be useful for large quantity of data having enough word for statistical comparisons to be considered significant. The obvious choice of large data set for this approach is made to provide the sufficient amount of learning data while training statistical models. However, once statistical models have been properly trained, these systems could successfully provide the response of even complex questions. Pattern based approach is applied in texts finding text patterns making this approach efficient exploiting the Web as a data source. It lacks in semantic understanding and reasoning yet, the pattern matching not only reduces linguistic computations but also aid in automatic wrapper generation for handling heterogeneous web data.

Research in QA has been developed from two different scientific perspectives, artificial intelligence (AI) and information retrieval (IR). An example of IR we see in 1999 in which the Text REtrieval Conference (TREC) had invested in QA systems the producing of answers to factual questions using a set of documents from the TREC corpus. A significant number of systems presented in this effort we had the successful combination of IR and NLP techniques. Various trends has been developed connecting the structured knowledge-based and the free textbased QA systems.

The sources of the answers can be completely unstructured like web pages are, semi-structured like the comment fields in databases or completely structured like database entries. The more structured the application, the easier it is to build a QA system upon it but at the same time the less universal the system can be (Hirschman L., Gaizauskas R. 2001:292-293), as it can deal only with this structured data and not with sources of whatever nature. Concerning the range of the search documents a QA system utilizes, they can be anything from fixed collections, as the TREC corpus mentioned before, to widely open collections, like a search over the Web. Finally, there are QA systems that can be applied on only specific domains, in the sense that they can only answer questions coming from a specific field.

QA systems can be used in many disciplines and areas of human activity which have their own specific knowledge sources. An example is the medical domain, that contains a vast size of technical information and resources that can be used for a QA system targeting this kind of information. Besides medical domain here are many applications in variety of other domains ranging from the field of computer science, geology, sports, and tourism among others.

But QA systems can be used as help desks in large organizations and companies. For these structures is very critical to satisfy the customer's need for information as quickly and efficient as possible. Although they might provide information by other means (web site, FAQs etc.) the existence of a QA system increases the company's value because the user receives the information instantly and by a more efficient and user friendly way.

Another use of these systems is in expert systems. These are simple computer programs which try to emulate decision making capabilities of a human expert. The main component of expert systems is the knowledge base which consists of documents produced or gathered by an expert in the domain.

Advanced reasoning techniques that are used in QA systems raise new challenges for advanced QA systems. The answers are not just extracted directly from the text or from structured databases, instead the procedure of building an answer can evolve several reasoning forms with the goal of generate explained and justified answers. The integrated knowledge representation and reasoning mechanisms enable the systems to anticipate an answer to questions that may raise and to solve cases where the answer cannot be found in the knowledge base. These systems should identify and explain false assumptions and others conflict types that might be found in a question.

3.5 Deep Learning Systems

Deep learning is really just a term to describe certain types of neural networks. Neural Networks are inspired by our understanding of the biology of our brain that has been described in the previous section. These networks have gained widespread recognition as an effective machine learning algorithm in various relevant applications. A neural network is an architecture that comprises of neurons, similar with those in the human brain. These networks have architectures that usually are consisted of three different types of layers: the input layer which contains the input feature vector; the output layer that consists of the neural network response; and one or more hidden layer(s) in between that contains the neurons that connect to both the input and output. An example of a neural network is called a Feed-forward neural network, because only allows signals to travel from input to output. There are also the Feedback networks which can have signals travelling in both directions by introducing loops in the network.

When one talks about an N-layer neural network, what is generally referred to N is the number of hidden layers plus the output layer (figure 5 shows the difference between a neural network and a deep learning neural Network).

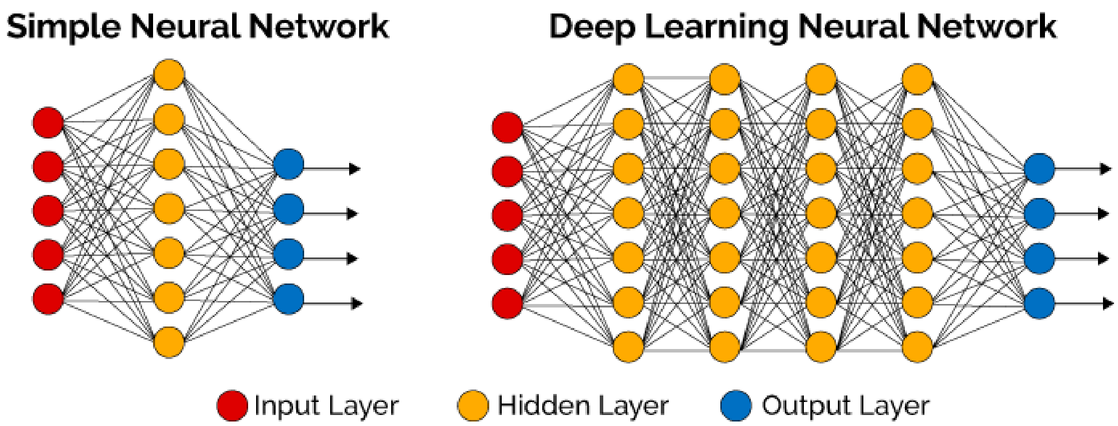


Figure 5 (Kampakis, S. 2020)

The Artificial neural networks (ANN) are consisted of four fundamental characteristics:

The network architecture:

- Input and activation functions
- The weight that each input connection has.
- The bias (denoted as *b*), which adds constant value to the input (a bias can be considered to be a measure of how easy is to get a neuron to "fire")

There are four major network architectures:

- Unsupervised Pretrained Networks (UPNs)
- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks
- Recursive Neural Networks

The network architecture and functions are chosen at the initial stage and remain the same during training. The performance of the neural network is reliant on the value of the weights. The weights are tuned during training so that a certain output is achieved.

Regarding the neurons, each one has three main parameters: the input data, output data and activation function. The input layer neurons take in information, in the form which can be numerically expressed. The following layers receive data from previous layers. The activation function converts the input data to the output. In the process of learning a neural network, the main variable parameters are the weights of the synapses, but sometimes also some parameters of the activation function.

Schematically, the work of the neuron is shown in the Figure 6 where synapses are represented as weights.

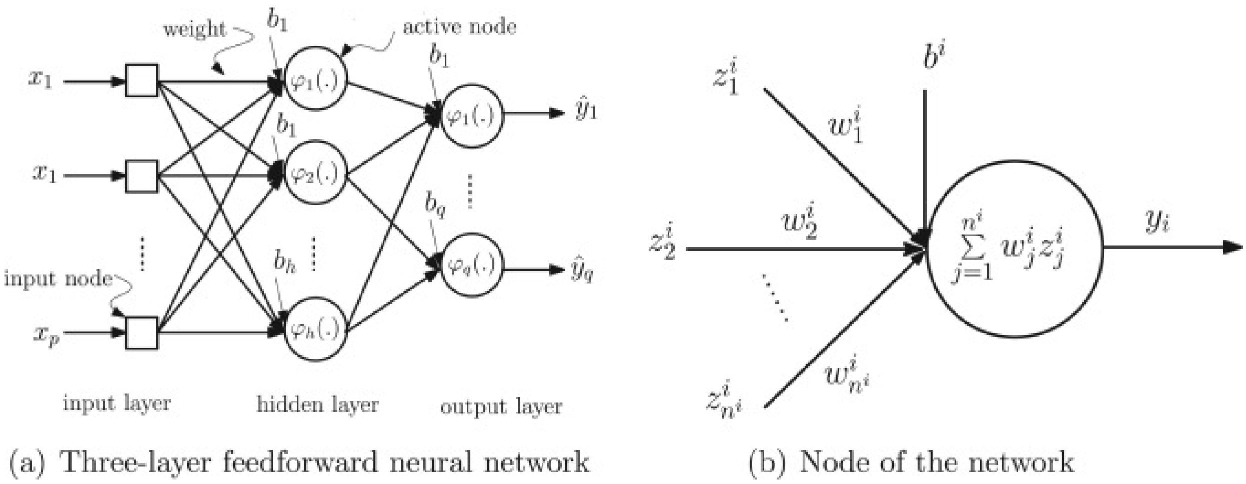


Figure 6 (Ojha, V. K., Abraham, A., & Snášel, V. 2017:99)

Therefore, the whole function is composed by gathering activation values from all neurons plus the bias (Figure 6).

Another interesting interpretation would be that, contrary on other methods; once the network is trained it does not provide any insight on what it does. As such, ANNs are sometimes referred as black boxes, as it is not impossible to understand their functioning.

Deep learning has been used successfully in many applications, and is considered to be one of the most cutting-edge machine learning and AI techniques. The associated algorithms are often used for supervised, unsupervised, and semi-supervised learning problems. For neural network-based deep learning models, the number of layers is greater than in so-called shallow learning algorithms. Shallow algorithms tend to be less complex and require more up-front knowledge of optimal features to use, which typically involves feature selection and engineering.

In contrast, deep learning algorithms rely more on optimal model selection and optimization through model tuning. They are better suited to solve problems where prior knowledge of features is less desired or necessary, and where labeled data is unavailable or not required for the primary use case.

More generally, deep learning falls under the group of techniques known as feature learning *or* representation learning. As discussed so far, feature extraction is used to 'learn' which features to focus on and use in machine learning solutions. The machine learning algorithms 'learn' by themselves, the optimal parameters to create the best performing model.

3.6 Paradigms of Cognitive Assistants

The following sections describe four successful models of cognitive assistants: Watson, Alexa, Cortana and Siri.

3.6.1 Watson

The history of IBM Watson had started when IBM had took the challenge to build a computer system that could compete at the human champion level in real time on the American TV Quiz show, "Jeopardy!". The "Jeopardy!" is a Quiz Show in USA that has been on the air since 1984 . It features rich natural language questions covering a broad range of general knowledge. It is a game that requires smart, knowledgeable and quick players who have very limited time to answer a question. Three contestants compete against one another in a competition that requires answering rich natural language questions over a very broad domain of topics, with penalties for wrong answers. The nature of the competition demands the use of many cognitive functions at their best performance. The researchers build the system not for performing in a laboratory for experiment reasons but to perform in real, hard, demanding situations and conditions like those which the Quiz creates.

In 2011, IBM's Watson computer beat two of the most successful human contestants of the show (figure 7).

Figure 7 (CBSNews.com 2011, February 17)

The computer's opponents were the two all-time best Jeopardy! champions, ever. IBM Watson beat them both in a live, real-time competition. Watson analyzed the clues in a real game of Jeopardy, in real-time game competition, and providing the answers according to the rules of the game. The event was held at Yorktown Heights in New York, at IBM Research. Watson won with an impressive lead of $77,147 after two exhibition matches and it marked a breakthrough in artificial intelligence with its understanding of natural language and ability to make sense of vast amounts of written human knowledge (CBSNews.com 2011) .

Although, in 1997, IBM had made the news with its Deep Blue supercomputer, which beat then World Chess Champion Gary Kasparov in a live, televised match. But, Watson seems to succeed a breakthrough since it excels a task that is not so easily reduced to logical rules, like chess. The machine had used the natural language in way similar we used when we talk to each other or read and go about our daily lives.

Watson was based on a QA system named DeepQA. It is a sophisticated software whose architecture's development and integration had conducted by the use of many different algorithms and artificial intelligence technologies making the Watson able to look the data from different perspectives. This approach had succeeded to develop a flexible question-answering system able for deep content analysis and evidence-based reasoning by using advanced natural language processing (NLP), information retrieval, reasoning, and machine learning and other artificial intelligence methods (Ferrucci, D., etc all 2010).

As the DeepQA can anticipate the available data from different perspectives it has the capacity to make, for each question, a number of possible assumptions, but also the potential to gather and evaluate all the data that support or fail a hypothesis. Additionally, it can provide many plausible answers. Its efficiency is also based on the fact that each component provides cognitive functions in such way that they can define the meaning of the questions and the correctness and the relevance of an answer.

The idea of using multiple components that can perform multiple functions had come by the assumption that each component in the system does or cannot deliver a perfect job. Watson's framework is called Unstructured Information Management Architecture (UIMA), and it is designed to support interoperability and scale-out of deep analytics (figure 8).

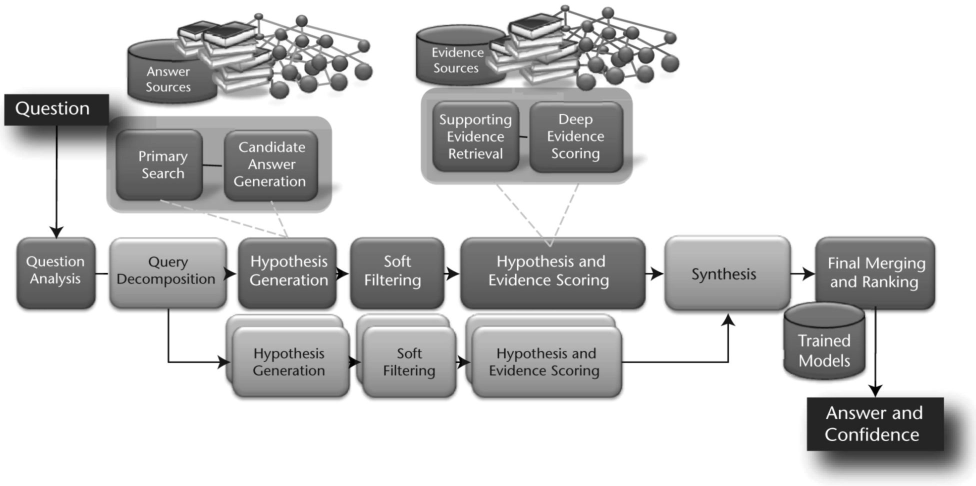


Figure 8 (Ferrucci, D., etc all 2010:69)

Watson receives a question as input and returns an answer and an associated confidence score as output. Therefore, except the massive parallelism of multiple interpretations and hypotheses and the number of components (experts) that do the analysis and the evaluation the system uses a pervasive confidence estimation which means that all components produce features and associated confidences, scoring different question and content interpretations which are then combined to produce a score. The system in order to determine the final answer collects the possible answers by evaluating the evidence from by structure and unstructured data. Structured data included databases, taxonomies, and ontologies while unstructured data that included a wide range of encyclopedias, dictionaries, thesauri, newswire articles, literary works, text corpora derived from the web.

When the system receives a question conducts the following steps ((Ferrucci, D., etc all 2010 :69-74)

1. <u>Content Acquisition</u>

The system defines the problem domain and produces a description of the questions that must be answered. Then applies a process that involves the identification of the necessary data (documents and nuggets) the scoring of data based on whether they are informative with respect to the original seed document; and the merging of the most informative nuggets in order to construct a reasonable baseline corpus used when applied (at apply-time) to answer new questions

2. Question Analysis

This is the step where the system attempts to understand what the question is asking and performs the initial analyses that determine how the question will be processed by the rest of the system. This is made by examining the question into its parts of speech and identifying the various roles that the words and phrases in the sentence are playing. During this procedure, Watson uses a mix of natural language processing (NLP) analytics, including keyphrase extraction keywords to be used to compose a query for primary search), information extraction (identifying entities and relations), lexical answer type identification (LAT) that represents the type of the answer, and question classification (each question is classified by a set of categories). Question classification may identify the type of a question (puzzle question, a math question, a definition question, etc) as well as the the constraints, definitions components, or entire sub-clues within questions

3. Primary Search

After establishing the question's analysis the system searches to find the sources that come from the data that contain the candidate answers. The primary search is performed in structured or unstructured data. More specific in case of the unstructured data the system uses a search engine which is in fact is a combination of different search engines that is applied in the data. The searching is performed by combining the keywords extracted by question analysis. The result is a list of text passages. In case of structure data the search is a SQL or SPARQL query to structure knowledge bases (KBs), returning a list of entities and their names. The question is converted from natural language to a structured query, matching the results of question analysis to the schema of the KBs that are used.

4. Hypothesis generation

The documents collection creates the question which triggers the search. The results of this search are a set of answers named as candidate answers or hypotheses. A key element at this point is the quantity of the possible answers since its large number increases the possibility that the set of these hypotheses includes the correct answer. Following, the results are analyzed by using information extraction (IE) algorithms that are able to identify entities and other relevant terms. A list of these entities is generated and it contains the entity names in the form of strings while it is also contain the links which they have been extracted. All those entities are regarded as competing hypotheses or candidate answers that must be evaluated during the step named Soft Filtering.. For their evaluation are used different strategies like articles' titles or Named Entity Recognition and key phrase extraction algorithms. The candidate answers are represented as strings and are linked to their sources when possible.

5. Soft Filtering

This is the step where the candidates answers are evaluate for their precision. The aim is to cut them down to a smaller set of candidates before the more intensive scoring components see them. This is made by applying a soft filtering score which all the candidate answers have to pass in order to proceed to the next step. The soft filtering scoring model and the filtering threshold are determined based on machine learning over training data.

6. Hypothesis and evidence scoring

Watson treats each candidate answer as a competing hypothesis. Therefore, it identifies evidence in a KB or in textual passages of candidate answers. For this step are used various techniques that are referred to as passage scorers. One particularly effective technique is passage search where the candidate answer is added as a required term to the primary search query derived from the question. This will retrieve passages that contain the candidate answer used in the context of the original question terms. Another technique applies the check regarding the compatibility between each candidate's type and the lexical answer type that is required by the clue. Different answer scoring algorithms(scorers) rate the quality of answers from different points of view while there is a set of possible answer scores that together provide a feature space for machine learning algorithms to assess the overall confidence of the answer. A very basic scorer is the lexical overlap, showing how many keywords are in common between the question and the supporting evidence for each candidate answer. Another scorer looks for taxonomic relations between the candidate answer and the lexical answer type in a large taxonomy of types.

The procedure of scoring is very demanding since Watson has to execute a very large number of analytics in parallel in a limited time frame. This execution is based on UIMA AS (Asynchronous Scaleout), a semantic integration platform that enables scale out on thousands of cores in a massively parallel architecture. These components can analyze text and produce annotations or assertions about the text. Watson has evolved over time and the number of components in the system has reached into the hundreds. UIMA facilitated rapid component integration, testing, and evaluation.

7. Final merging and ranking

The goal of final ranking and merging is to evaluate the hypotheses and their scores to identify the single best-supported hypothesis given the evidence and to estimate its confidence. Watson through training learns how to weigh, apply, and combine its own algorithms producing in that way the final ranking for all the possible answers. This is done by training a logistic regression model on the task of providing a confidence score close to 1 (one) for the positive answers, and close to 0 (zero) for all remaining candidate answers. The produced answers are finally ranked accordingly, and the top one is selected if the confidence is above a game-specific threshold.

All the equivalent answers are grouped together and their supporting evidence is combined. This is called final merging and it is *d*one by exploiting information in the knowledge bases, such as synonymy or selected semantic relations.

A regression algorithm takes each feature vector and assigns a single confidence value to the candidate answer. Training is a key element for this step and it is performed by using historical data that is provided by past Jeopardy! games. The resulting models are stored and used when that is necessary. So this algorithm is trained on thousands of candidate answers and question pairs, each labeled for whether the answer is correct or incorrect with respect to the question, together with their feature vectors, and learning to predict a probability of being a correct answer.

The following Watson APIs, that show of what Watson is capable to perform, are currently available (IBM Watson products)

Language:

- Conversation
- Document Conversion
- Language Translator
- Natural Language Classifier
- Natural Language Understanding
- Personality Insights
- Retrieve and Rank
- Tone Analyzer

Speech:

- Speech to Text
- Text to Speech

Vision:

- Visual Recognition
- Data Insights:
- Discovery
- Discovery News

In 2014, IBM formed the Watson Group to commercialize Watson technology(IT Infrastructure). This group had launched the IBM Watson unit, a business dedicated to developing and commercializing cloud-delivered cognitive computing technologies. The move signified a strategic shift by IBM to deliver a new class of software, services and apps that think, improve by learning, and discover insights from massive amounts of Big Data. IBM is investing a significant amount of money into the Watson unit, focusing on development and research, and bringing cloud-delivered cognitive applications and services to market. This includes investments that support IBM's ecosystem of start-ups and businesses building cognitive apps powered by Watson.

3.6.2 Alexa

Alexa is Amazon's cloud-based voice service available on Alexa devices like the Echo and Echo Dot, as well as Alexa companion devices like the Fire tablet and Fire TV. In other words, Alexa is the cognitive system (software) which needs, besides, the Echo -the physical device- a WiFi network so it can function and allow the stakeholder to interact with it. Alexa skills are apps that enable voice-activated capabilities for connected smart devices and online services. It is a system able to play music, provide information, deliver news and sports scores, tell you the weather, control your smart home and even allow ordering products from Amazon.

Regarding the physical device, the Echo, is a cylindrical construction that measures around 9.25 inches (23.5 centimeters) in height and 3.27 inches (8.3 centimeters) in diameter. A microphone off/on button and an action button at the top of the device provide some control options. The main control is the seven-microphone array built into the top, which uses beamforming technology and noise cancellation to "hear" our voice. A light ring at the top outer edge provides status information, such as the volume level and whether the device is streaming audio or the microphone is turned off, via various light colors and motions. An LED that lets you know the status of the device's WiFi connection sits near the base just above the power cord. Echo comes with a 21-watt power adapter, which is its only power source. It is far more than obvious that Alexa, besides the above requirements (the physical device (Echo) and the WiFi network) needs space for its own and is not portable.

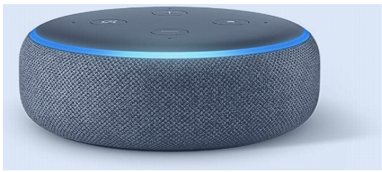


Figure 10 Echo Dot (3rd Gen) - Smart speaker with Alexa - Charcoal

Skills can be enabled on Alexa devices for a variety of voice-enabled services. Users interact with Alexa by saying a wake word, either “Alexa” or something else according to their preferences, to wake the device and then speaking an invocation phrase that consists of an utterance, the invocation name of the skill, and supported connecting words. For example, by using the voice command “ What's the weather in [city, state or city, country]?” we can ask “Alexa, What's the weather in Athens, Greece?” As amazon states, in app’s description, this command wakes the device, opens the weather skill, and a card opens in the Alexa app with a seven-day forecast for the requested location. Alexa, according to Amazon uses AccuWeather for the latest weather information and uses the device location set in the Alexa App settings.
The system works as follows: parses the spoken words, interprets the commands and routes them to the appropriate web service to get the right response. Alexa then converts the response (whether from an Alexa service or a third-party web app) and sends it back via audio to our Echo, and in many cases via text and graphical cards to the Alexa app home screen (figure18). As its manufacture states, the Echo device can be connected with other devices (like smartphone) via Bluetooth. It supports audio streaming from smartphones and tablets via Advanced Audio Distribution Profile (A2DP) and voice control via Audio/Video Remote Control Profile (AVRCP).

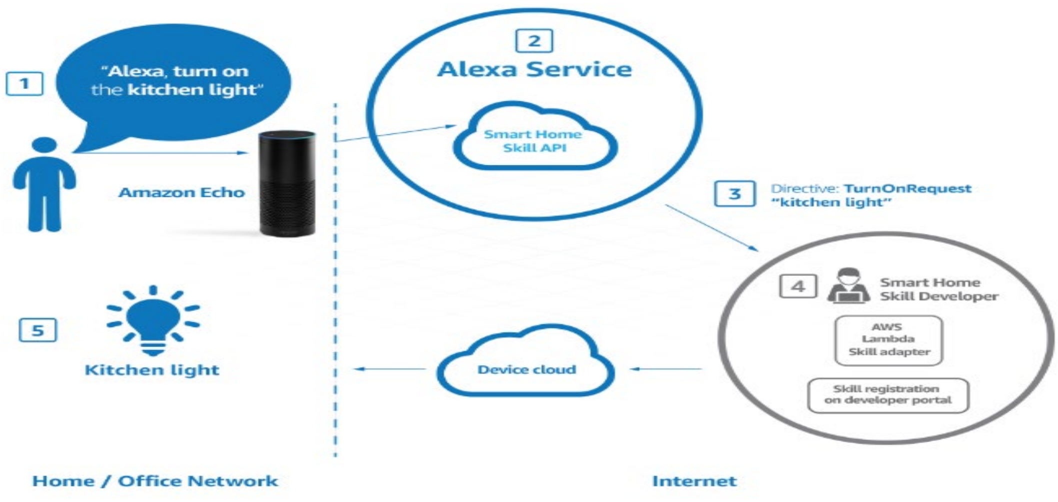


Figure 11(Gonfalonieri, A. (2018)

The system uses the Alexa Skills Kit that contains the Alexa's version of apps. Some are created by Amazon and affiliated partners, but the vast majority have been created by third parties. Independent developers and large companies alike have published skills for Alexa to help for doing daily things like playing, ordering or booking.

However, Alexa can only comply with and function within the range of skills that have been created and enabled for its use. If a skill does not exist or has not been enabled, Alexa will not be able to process the request submitted by the user. In other words, the system cannot "learn" a new skill by itself; instead it has to be re-built in order to develop new skills. For this reason the system uses the Alexa Skills Kit that contains the Alexa's version of apps. Some are created by Amazon and affiliated partners, but the vast majority have been created by third parties. Independent developers and large companies alike have published skills for Alexa to help for doing daily things like playing, ordering or booking.

Even though, there is a limit regarding the type of skills which Alexa can deploy. In fact there are only three distinct types of skills that can be created by the developers, these are: custom skills, smart home skills, and flash briefing skills. The level of effort regarding the construction of each skill is different since there are cases in which the developers work from the scratch while others use Alexa's Application Program Interface (API).

An important aspect regarding Alexa is that during its function collects quite a bit of data. The type of data varies and includes the following: skills enabled by a user, questions or requests made by the user, requests and changes made with the Alexa app, all responses given by the Echo, basic Amazon subscriber information, including payment options and shipping and billing address, information provided by third-party services, such as from a linked account.

Amazon states that collecting such data helps them to analyze user trends, enabling them to provide users with more customized services and features such as recommendations and advertisements. But the contra-argument for that storing is that Alexa works as a 'collector' of data while the real process of the collected data is made by Amazon. All collected data is stored on Amazon servers which users can access and view their data and their interactions with the system. The log history display what the device heard and how it responded. Also, users will be able to listen to the snippet of speech that was recorded by the device, giving users the opportunity to hear exactly what Alexa heard and recorded. The log history can be deleted by users.

But, can Alexa be considered as a cognitive system? It uses natural language for the interaction with the user it can interact with its environment (users and cloud devices) in order to perform a task or tasks optimally. Even more it has the ability for adaptation since it can react in cases the environment shows changes. This adaptation is exhibited through the introduction of new skills. Additionally it has a kind of "memory" since it records all the transactions and it can make predictions (by using these previous transactions) while it is flexible by using profiles (custom skills, home skills etc.). Finally, it demonstrates a simulation of the use of some cognitive functions (besides language and memory) like perception ("perceives" the user's commands) and learning (learns user's habits). Under this frame we can call it as a cognitive system.

But there are several issues which prevent us to characterize it as cognitive system. The first one is the procedure of learning. Although, cognitive systems are able to learn based on their data inputs (like Alexa), yet they also learn from their outputs and more important from their experience; furthermore, cognitive systems receive training, constructing, in that way, a learning ability able to make generalizations based on their exposition to a number of cases and then be able to perform actions after new or unforeseen events.

As we saw, Alexa collects data which are processed by Amazon's staff. All the knowledge and experience which Alexa gathers is compiled and processed by humans witch then re-design and re-built Alexa by embedding it new skills. There is no training between user and machine. In cognitive systems, learning can be done by applying specific learning strategies, such as the supervised strategy to map the data inputs and model them against desired outputs, and the unsupervised strategy, to map the inputs and model them to find new trends. Leaning in cognitive systems understand patterns while their experience from

their past interaction gives the impression of “self –reflection”.
On the other hand, Alexa during its construction is embedded with the necessary knowledge which looks like a set of skills (knowledge to do something) for specific domains and not to provide personalized solutions. It cannot learn something new by itself since the learning procedure is defined by the human factor(Amazon’s staff & third parties) who provide the new skills.
Cognitive systems can perceive and reason. Cognitive systems try to “mimic” the human perception follow somehow the same protocol, they have components (knowledge, comprehension, decision policy, decision making) that are combined and work together for an optimum “perception. Alexa perceives requests that had already been “described” as closest as possible. It cannot “self-organized”, its development is not autonomous and it cannot learn new facts from its own.

3.6.3 Cortana

Cortana, like Alexa is the voice-activated digital assistant released by Microsoft. It was first introduced in 2014 for Windows Phone and Microsoft hoped it would compete with Siri and Alexa. The system named after Cortana which is a synthetic intelligence character in video game franchise of Microsoft’s Halo. Cortana can set reminders, recognize natural language without the requirement for keyboard input, and answer. It is helpful, among other things in getting weather forecasts, setting up reminders, telling jokes, sending emails, finding files and searching the internet.
A significant difference from Alexa is its ability to conduct casual conversations with users; what Microsoft calls "chitchat" while it supports seven languages; English, French, German, Italian, Spanish, Chinese, and Japanese. Additionally, since it is cross-platform, it has an open SDK for third party developers who can create their own applications and to demonstrate new actions. Actions are however restricted to Windows 10 Desktop and Mobile, and Android. The developers need to register their actions, which can be done without any cost, yet these actions are reviewed by the Cortana Team at Microsoft.
Cortana for its function uses the Microsoft’s speech API (Application Program Interface) which is a cloud-based automatic service that provides translation. It also uses Bing as web search engine and Santori which is semantic search database.

Figure 12 shows how users interact with Cortana and your skill. Regardless the platform when the user requests a skill then this skill is executed in the cloud, not on the actual device.

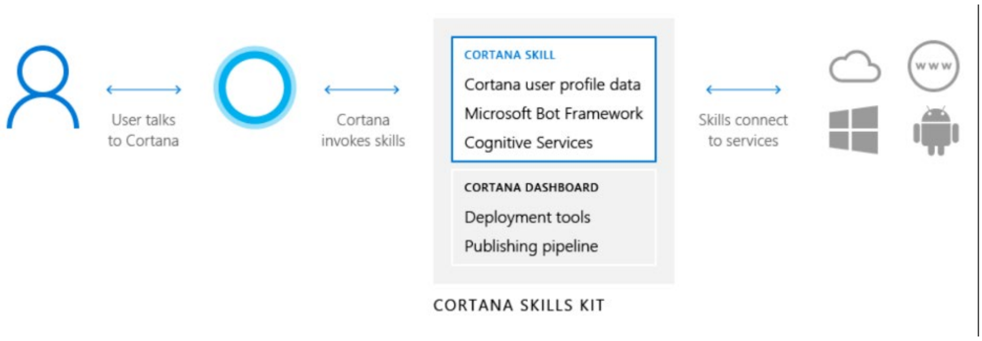


Figure 12

In fact Cortana is a part of a completed intelligence suite. This is obvious when we see the architecture behinds Cortana's function (figure 13).

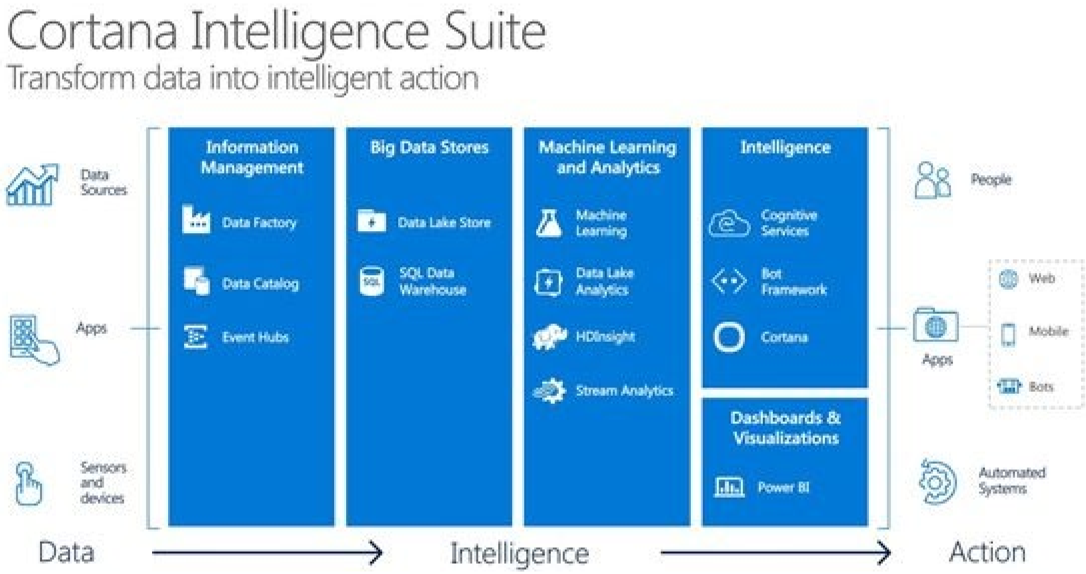


Figure 13 (Vignesh, M. 2017)

The Azure Data Factory collects the data from the users, manages the data sources from the users' application while it builds the pipelines between the users and these applications. The Data Catalog manages the data and it simplifies data source discovery via search. The Event Hub is a cloud-scale telemetry ingestion service that collects, transforms and stores events. Main purposes are to allow events to be ingested into Azure from many data sources, apps, platforms and devices.

The Data Lake Store is a cloud that works as a storage place for unstructured, semi-structured and structured data. It has no restrictions on data size making it ideal for storing big data. The SQL Data Warehouse works with a massive parallel processing power and supports relational and non-relational data. The data itself is stored in blog storage (not SQL DataBase).

Machine Learning is a service for building predictive models using statistical techniques. It can learn from existing data to forecast future behaviors, outcomes and trends. Main purposes are anomaly detection, clustering scenarios, multiclass classification and regression analysis. Data Lake Analytics is an on-demand cloud analytics service where parallel data transformation and processing programs can be run. Main purpose is to process any data, regardless of size or structure. HDInsight is the only fully-managed cloud working as a cluster-as-a-service offering for distributed big data processing, scaling and querying capabilities. Stream Analytics is an analytic processing engine, and provides real-time stream processing in the cloud. Main purposes are to gain real-time insights into data coming from IoT devices and other applications, and perform real-time analytics on data from Event Hubs.

Cognitive Services are a set of APIs, SDKs and cloud services to build intelligent systems. Main purposes are to make applications more personalized, intelligent and engaging by incorporating emotion recognition, facial detection, video intelligence, recommendations etc. Bot Framework allows humans to interact with computer systems in a more humanly fashion. The main purpose is to build and connect intelligent bots with end users, wherever they are and whatever platform they use. Supported platforms include custom websites and apps, Facebook Messenger, SMS, Skype and Slack. Cortana is the virtual personal assistant for asking questions, finding and managing things, and monitoring and alerts. The main purpose is to connect users to systems and services across platforms and devices. This is typically performed using natural language. Cortana interacts with Bot Framework and application APIs to provide answers and results for the end user. Dashboards collect data from various sources, extract relevant information, and present it using powerful dashboards and graphics.

A key element is the use of scenarios. These help Cortana to function in a proactive manner to make predictions and to establish a decision automation and support. This technology has a range of practical applications, and is already being used to solve complex analytics challenges around the world as figure 14 displays.

Industry	Sales & marketing	Finance & risk	Customer & channel	Operations & workforce
Retail	• Demand forecasting • Loyalty programs • Cross-sell & upsell • Customer acquisition	• Fraud detection • Pricing strategy	• Personalization • Lifetime customer value • Product segmentation	• Store location demographics • Supply chain management • Inventory management
Financial services	• Customer churn • Loyalty programs • Cross-sell & upsell • Customer acquisition	• Fraud detection • Risk & compliance • Loan defaults	• Personalization • Lifetime customer value	• Call center optimization • Pay for performance
Healthcare	• Marketing mix optimization • Patient acquisition	• Fraud detection • Bill collection	• Population health • Patient demographics	• Operational efficiency • Pay for performance
Manufacturing	• Demand forecasting • Marketing mix optimization	• Pricing strategy • Performance risk management	• Supply chain optimization • Personalization	• Remote monitoring • Predictive maintenance • Asset management

Figure 14 (Vignesh, M. 2017)

Microsoft is trying to create a new version of Cortana. In fact Windows 10 will include access to a new Cortana experience with an emphasis on productivity, helping the user to find information across Microsoft 365. Cortana would include new capabilities in better reading and summarizing emails, text messages and other types of communications for users on the go.

In 2019, Microsoft bought Semantic Machines, a conversational AI startup in order to expand Cortana's functionality to look like more as a cognitive assistant. For instance, it will have the ability to pull important points out of long messages and summarize them. In that way Cortana would offer an assistance since the human brain struggles to digest long, complex messages.

However Cortana cannot be characterized as a "real" cognitive assistant. Cognitive assistant should be adaptive and self-teaching. The Knowledge should be dynamic. Cortana like Alexa interact with users by providing an answer to the questions they formulate working more as a Question Answering systems with some extra skill but Cognitive systems interact with users and improve their decision making process. Cognitive System should go beyond answering a series of questions and should exhibit a proactive behavior. They should have interactivity in natural language and the capability to solve ambiguities. Moreover, they can *adapt* their reasoning and learn dynamically from information sources.

3.6.4 Siri

Siri is a built-in, voice-controlled personal assistant that uses sequential inference and contextual awareness to help perform personal tasks for iOS users. Siri is designed to offer interaction with an iPhone, iPad, iPod Touch, Apple Watch, HomePod or Mac by speaking to her using natural language. It has access to every in application on usre’s Apple device - Mail, Contacts, Messages, Maps, Safari etc. – and it will call upon those apps to present data or search through their databases whenever she is asked to do so .

The user can initiate Siri by two ways, either by voice by just saying "Hey Siri" or by using buttons. Some of the use cases supported by SIRI are the following: call someone from the contact list, launch an application on iPhone, send a text message, set up a meeting on a calendar, play a song in iTunes library, create a note, check weather, online booking, get directions etc..

Siri can speak and understand English, Spanish, French, German, Italian, Japanese, Korean, Mandarin, Norwegian, Cantonese, Swedish, Danish, Dutch, Russian, Turkish, Thai and Portuguese while it is possible to change Siri to a man or the opposite. Once the Siri microphone button is touched, whatever is said is recorded, compressed and sent to Apple's data centers.

When a voice file arrives at Apple‘s data center, the Nuance speech-to-text engine translates the request into text. The system breaks down the message to identify particular patterns, phrases, and keywords. Additionally, this input is combined with other information such as location, time, other task context etc.. This data gets input into an algorithm that sifts through thousands of combinations of sentences to determine what the inputted phrase means (figure 15).

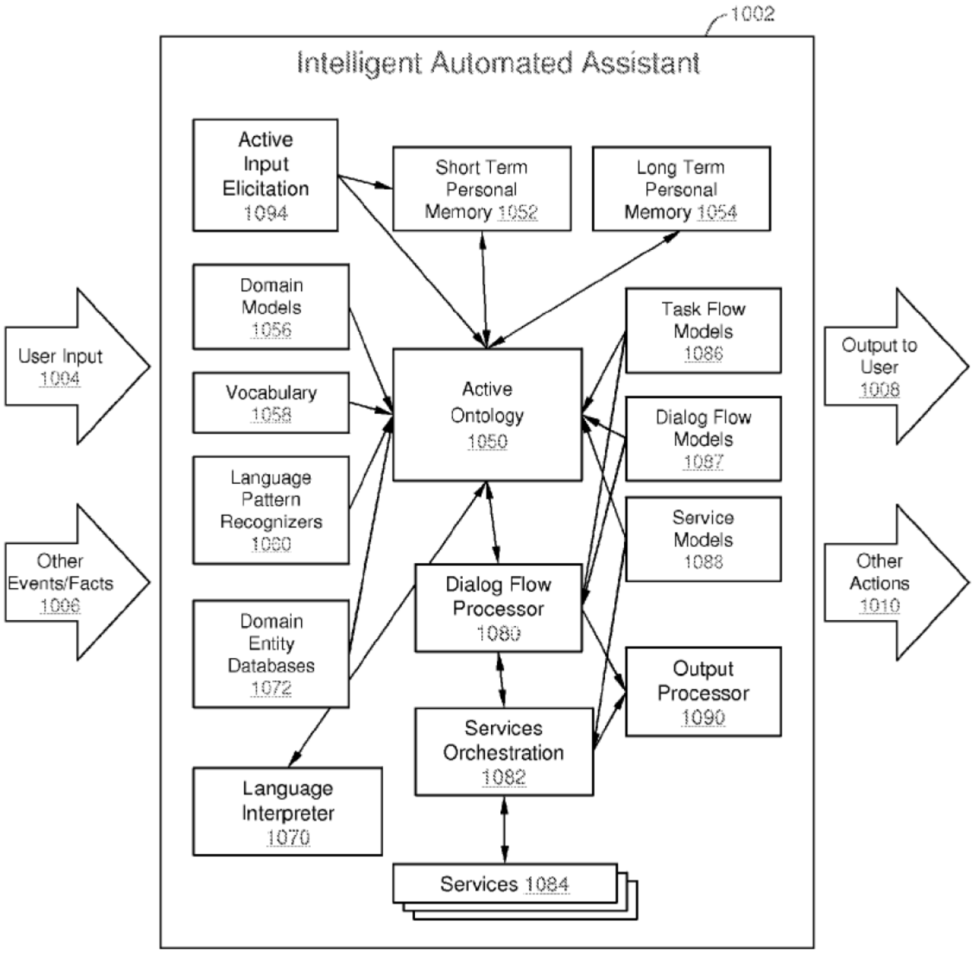


Figure 15

Once Siri determines its request, it begins to assess what tasks needs to be carried out. When Siri needs more information to fulfill a request it asks the user for more information without forgetting what was originally asked. Siri is then able to display the answer or to provide a service.

In any case Siri operates in a closed ecosystem. It doesn't work with other applications or a service other than those Apple has connected on the backend. Furthermore Siri, like the other cognitive assistants, works as an example of Artificial Intelligence that provides in some way intelligence yet it lacks from several cognitive abilities. For instance, an advanced version of Siri enriched with several cognitive abilities would have perception (hearing the phone), comprehension and decision making (answering or not), language skills (talking and understanding language), social skills (interpreting tone of voice and interacting properly with another human being), while there also other cognitive functions besides those in the example.

Reasoning, learning, planning, flexibility and working memory are some examples. Some of these systems demonstrate a number of these functions but in fact these systems exhibit the ability of humans to construct machines that produce some sort of intelligence.

Chapter 4
System's Architecture Design and Analysis

4.1 Why the Call Assistant is Necessary

The history of mobile phones starts back to 1908 when a US Patent was issued in Kentucky for a wireless telephone. Mobile phones were invented as early as the 1940s when we had cells for mobile phone base stations. However, the devices that had made use of the cells were not really mobile phones at all. They were two-way radios that allowed people to communicate.

Motorola, on 3 April 1973 were first company to mass produce the first handheld mobile phone that weighed 1.1Kg (Jackson, K. 2018). In 1992 the world's first ever SMS message was sent in the UK. Neil Papworth, a developer for a telecom contractor tasked with developing a messaging service for Vodafone sent to Richard Jarvis, a director at Vodafone a text message by saying "Merry Christmas" (Vodafone 2017).

Since then, the mobile phone industry grew enough to create powerful and more sophisticated phones available for every person on Earth. In fact, the World Advertising Research Center (WARC), estimates that around 2 billion people currently access the internet via only their smartphone, which equates to 51 percent of the global base of 3.9 mobile users. They also forecast that almost three quarters (72.6 percent) of internet users will access the web solely via their smartphones by 2025, equivalent to nearly 3.7 billion people (Handley, L. 2019).

According to research from RescueTime, one of several apps for iOS and Android created to monitor phone use, people generally spend an average of three hours and 15 minutes on their phones every day, with the top 20% of smartphone users spending upwards of four and a half hours. Pickups are also an important metric in determining how our

devices affect us. On average, we pick up our phones 58 times a day while we check 30 times during working hours (9am–5pm) (Matei, A. 2019).
Assurion (2019), an industry in mobile insurance, technology and support had conducted a research according to which Americans check their phones 96 times a day – that's once every 10 minutes, that’s a 20 percent daily increase from a similar survey conducted by the company two years ago. Eighteen- to 24- year-olds check their phones twice as much as the national average (King, J. 2019).
The above facts display why a potential use of a cognitive assistant responsible of handling call messages could be very helpful. The use of a cognitive assistant will save time and it will increase productivity, not only during the times we spend in the office, but in the rest of the day, since there also tasks and events that we have to cope. The cognitive assistant can work around the clock, seven days a week, with no breaks and without tiring, a fact that save resources and time (and some cases money), while it increases the stakeholder’s flexibility on scheduling or performing an event.

4.2 System’s Architecture

The assistant should have a cognitive architecture having the characteristics of the cognitive architectures which have been described in the previous chapter. The assistant’s architecture should be capable of holding memories beliefs, goals and knowledge. It should also create mental structures (contexts) using its memory and knowledge base and it should be equipped with processes (including performance and learning mechanisms) that operate on these structures.
An architecture that fulfills these criteria is proposed by Michael, L., Kakas, A. C., Miller, R., & Turán, G. (2015 :4). (figure 16) which is refined by Kakkas and Michael (2016) (figure 17) in which the assistant sense stimuli from the user and the environment. These stimuli are processed by the modules of the architecture that produce an action from the Assistant.
A significant characteristic is that the process of the stimulus should run parallel in all modules and parallel with-in the modules.

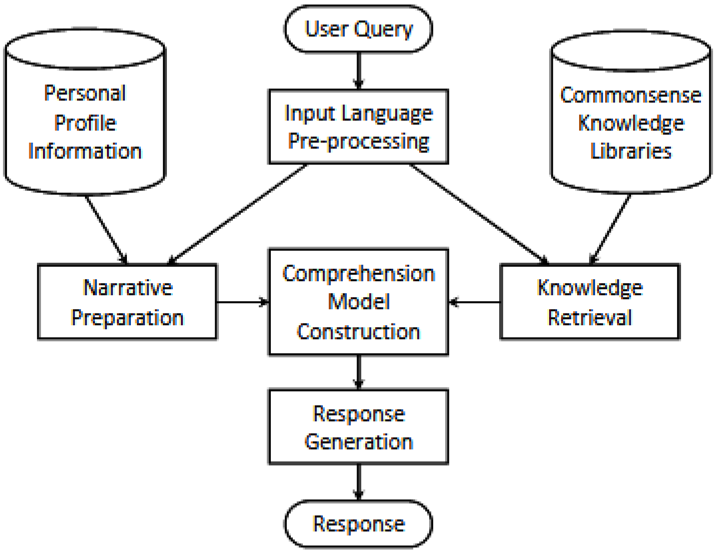


Figure 16

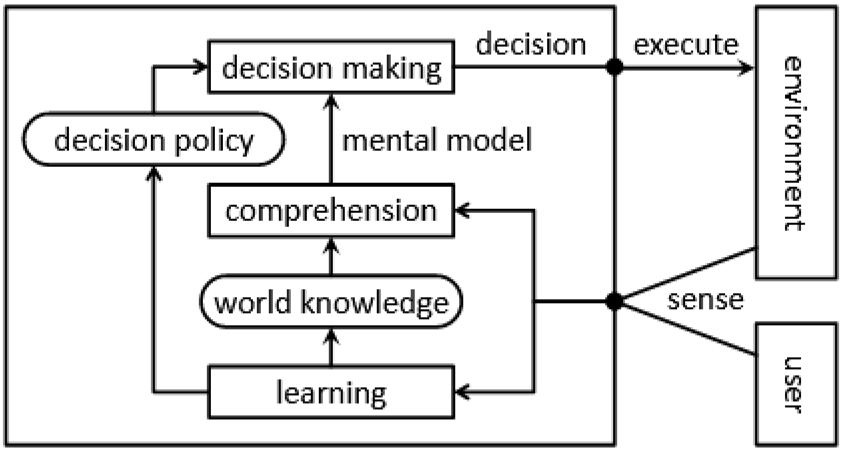


Figure 17

Another critical aspect of the system would be the use of Natural Language Processing tools that would establish the communication between the two parts. Mitsikas, T., Spanoudakis, N.I., Stefaneas, P., & Kakas, A. (2017) proposed a model that uses Natural Language Processing tools and the SoDA Methodology to develop and generate an argumentation theory that captures the guidelines of operation given in the natural language description of the user (figure 18). This aspect can minimized the need for detailed operational instructions since there is a level of interaction between the user and the system where the two understand and can anticipate the behavior of each other (Michael etc. all, 2015.)

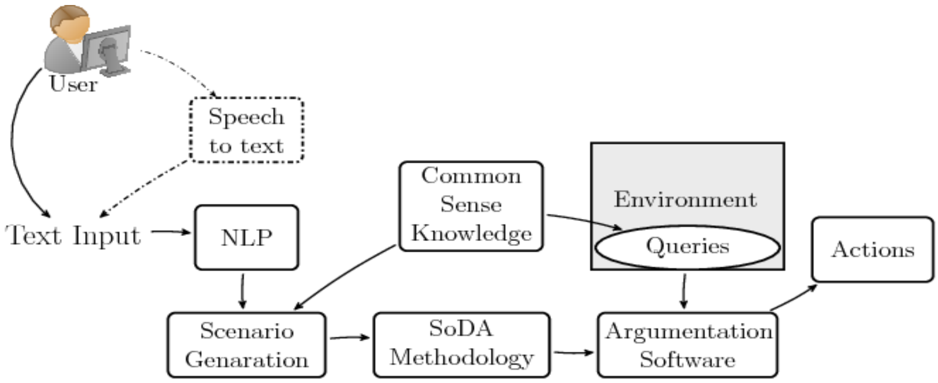


Figure 18 From Natural Language to Argumentation and Cognitive Systems Theodoros Mitsikas, Nikolaos I. Spanoudakis, P. Stefaneas, A. Kakas

Using the above ideas and architectures as a reference point, the Call Assistant makes a use of a slight altered architecture which is displayed in figure 19.

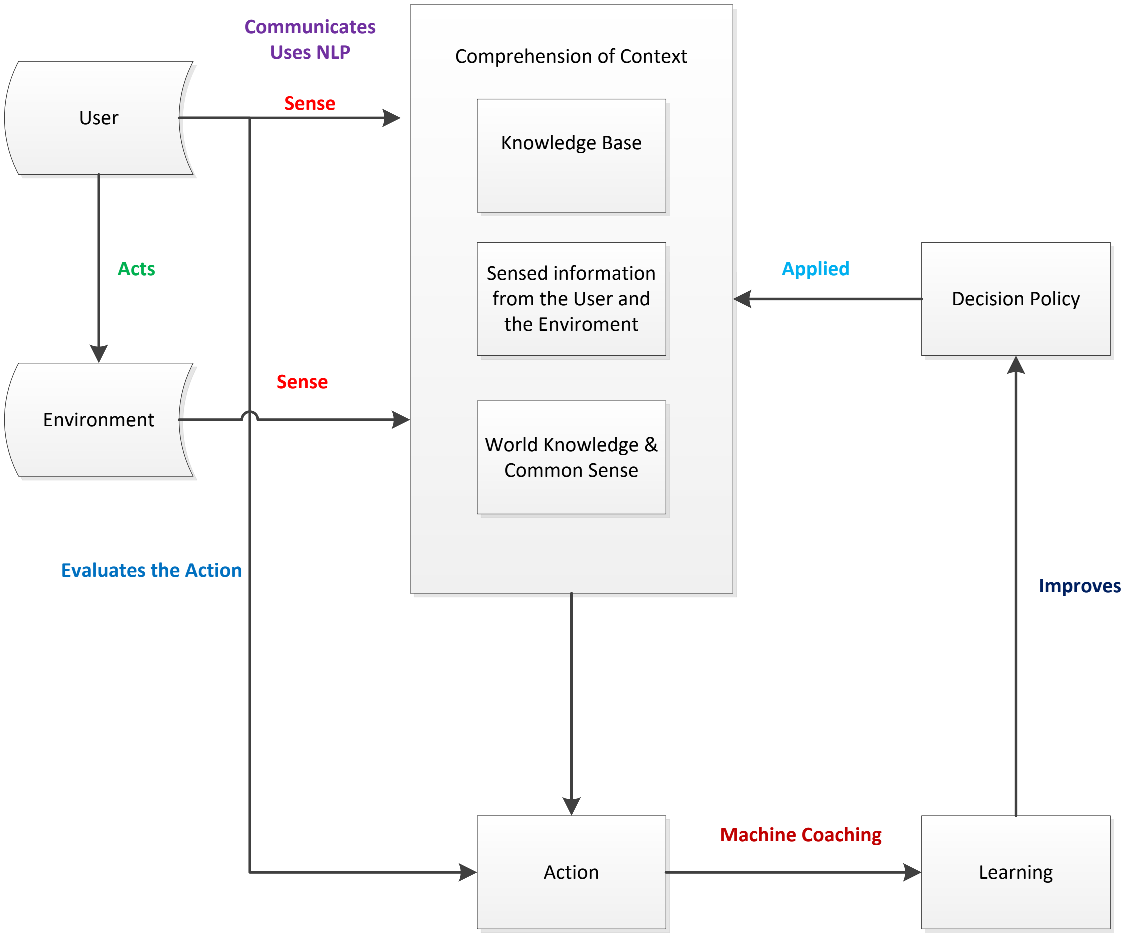


Figure 19 Architecture of the Call Assistant

This architecture succeeds a series of characteristics that fulfill the criteria for an efficient and an ideal cognitive architecture. First, it establishes a direct and immediate behavior, making decisions and acting in an effective and timely manner.

Second, the system will learn the routine behaviors gradually, by using, besides some means of knowledge acquisition; the user's coaching during its function.

Third, through the interactions between system and user and/or environment, the Call Assistant should be able to recognize places, actions, and persons as instances of known patterns but also it should also be able to learn new patterns and categories, modify existing ones, either by direct instruction or by experience.

Fourth it can identify and represent alternative choices and then decide which are the most appropriate and select an action for execution. These decisions can, through learning, be improved while it has the mechanism to predict situations and events, and therefore can anticipate some future conditions, in other words, it can adapt.

Fifth, the module of knowledge defines the Assistant's beliefs about user's word and environment while the comprehension and learning modules allows the cognitive system to draw inferences from these beliefs, either to maintain the beliefs or to modify them.

Finally, the system has a mechanism, by using the mobile sensors, to represent and store motor skills (like movement from place to place) that can be used in the execution of the assistant's actions.

4.2.1 User and Environment

The Call Assistant will operate in a difficult and demanding environment while there is no "typical" user profile. It will work in an open and dynamic environment because there is no limit to the novel combinations of circumstances that can arise. Furthermore, there is no a single user's profile; each user is unique, as each person is unique. For instance, different people, different environments, different cultures, different habits, different schedules, different levels of education, different levels of familiarity with technology, all these factors make the project unique.

Even more, the performance measures are high because the idea is to "compare" the Call Assistant and its capabilities with a human assistant. But the description of the environment and the obstacles do no stop here. The environment is also partially observable and stochastic. The system has sensors to identify who is calling or it can know time and date of calling, or to use a satellite global positioning system (GPS) in order to has accurate position information with respect to an

electronic map, it movement, but it cannot perceive the environment fully.
The following example defines the problem, suppose the user is at work which the Call Assistant can observe it, yet the system cannot observe that the user has just met his manager and they have a discussion at hoc. We have a new situation which defines a different decision policy about calls. Now, the environment has changed, the user knows it, a human assistant would know it, yet the call assistant does not. In other words, the assistant can observe a part of the environment while it cannot predict, in a sufficient level, the behavior and actions of other people.
The previous example, displays another aspect of the environment. It is continues because it has continues states that change over time while it is a multi-agent due to the presence of other people that participate in it.
The assistant will use the mobile's sensors and resources like calendar appointments and contacts, current date and time, in order to construct, as close as possible, the states of the environment.
Regarding the actions must be more or less the same as those available to a human assistant: answer or decline calls, send messages, set notifications, control in some level the management of the phone.
The communication between the user and the assistant is established through the use of NLP tools avoiding the sterile and monotonous communication which most of the software has, for example through pre-established forms. We want the communication to be more dynamic and user independent.

4.2.2 Comprehension of Context

The system's success relies, first, on how it would interact with the user, second on how it comprehend the information, third, on how it would act and finally on how the user will coach the machine. The idea is that the assistant will be cognitively-compatible with, human abilities as much as possible.
For example, how a human assistant would manage the situation? Which information a human would use and how he/she would process them in order to act? Let assume, for simplicity reasons, which the human assistant knows already, as much as possible, about the stakeholder's profile (job, schedule, daily routine, social life, habits, hobbies etc.) and he/she receives a call. By using our experience we assume that the human assistant, before acting, would ask the following questions:

1. Who is calling? (The person who calls carries a “status”, a “weight”, which affects our decision to answer or not, for instance if the person is a family member, a mother or a child with health issues.)
2. What time we receive the call? (The time of calling combing with other facts can characterize the call either important because we expected it, or insignificant and even annoying because on the other line is someone who acts on antisocial behavior.)
3. Where is the stakeholder when we receive the call? (There are times, during the day in which the user cannot answer the phone, like being in an important job meeting or being in a doctor’s office for medical examination.)
4. What the stakeholder does when we receive the call? (For instance, during driving a car, according to law, the user cannot answer the phone unless he/she uses Bluetooth hands-free, on the other hand, if the user seats next to the driver this restriction is not applied.)
5. Are there any specific directions: for the person who calls? (For instance every time Iraklis calls pass me the line), for the time? (example: today between 08.pm and 10.pm deny all the calls, because I want to have dinner with my wife), for the place? (example: when I am at the library for studying deny all the calls), for the stakeholder’s actions? (example: when I run deny all calls), or other general directions? (example: when I receive calls from private number, deny the call)

The answers of these questions combined with the human’s assistant knowledge about the stakeholder, the human’s assistant experience from previous calls and the human’s assistant common sense will define the response-action of each calling. After the call, the assistant is able to provide explanations about his/her actions, to justifying them and to receive guidance for the future calls making adjustments to the decision policy that he/she applies. By this example, we see that our system needs autonomy, ability to perceive the environment, learning, anticipation, action, and adaptation.

But how humans’ mental model works? It seems that memory plays a critical role. In fact, there are a relative large number of theories (the number varies from 15 to 45) about the structure and functioning of the human memory system that was introduced by various researchers, as Kent (2016:165) estimated in his research. Terms like “sensory memory”, “primary memory”, secondary memory”, “short – term memory “or “long –term memory” were introduced in order to define our memory system and our mental model as a whole.

More or less, all these theories define a memory system that combines components which coordinate together in order to “construct” a complete mental model. However, as Kent (2016:164) states clinicians and researchers have not yet agreed to a universally accepted theory about our memory system. Questions like how attention and working memory inter-related is not yet clarified clearly.

For the time being, the most dominant theory about our mental model comes from Alan Baddeley (2011) who has suggested the working memory model in which working memory is like a multi part system, and each system is responsible for a different function (figure 20). According to the model the working memory contains four elements: the visuospatial sketchpad, the phonological loop, the central executive, and the episodic buffer.

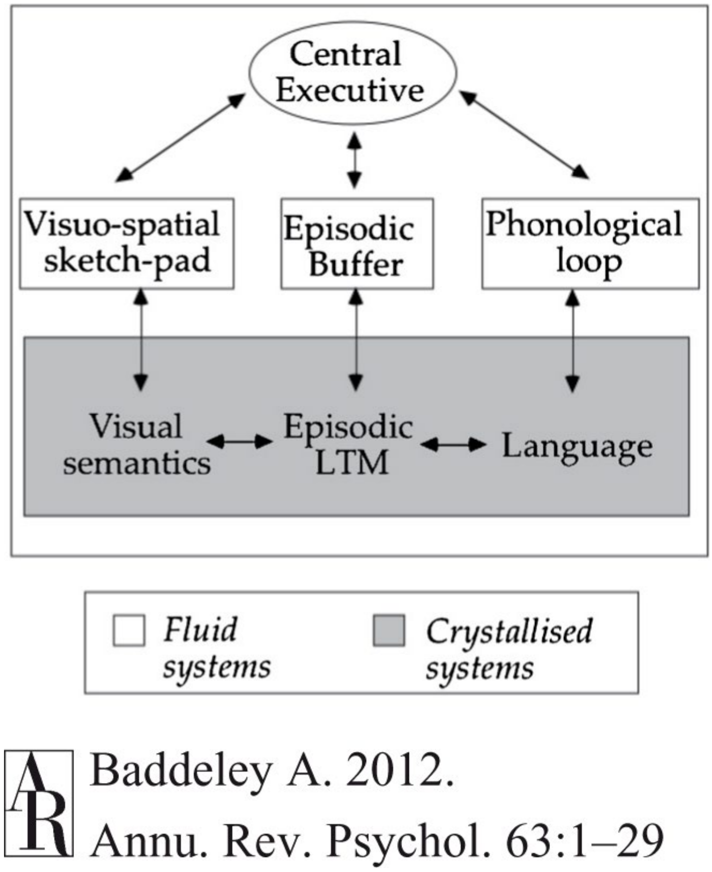


Figure 20

The visuospatial sketchpad holds visual images; the phonological loop holds information about speech for verbal comprehension and for acoustic rehearsal. There are two critical components of this loop. One is phonological storage, which holds information in memory. The other is sub vocal rehearsal, which is used to put the information into memory in the first place. The third element is a central executive, which both coordinates attentional activities and governs responses.

The central executive is critical to working memory because it is the gating mechanism that decides what information to process further and how to process this information. It decides what resources to allocate to memory and related tasks, and how to allocate them. It is also involved in higher-order reasoning and comprehension and is central to human intelligence.

The fifth component is the episodic buffer. The episodic buffer is a limited-capacity system that is capable of binding information from the visuospatial sketchpad and the phonological loop as well as from long-term memory into a unitary episodic representation. This component integrates information from different parts of working memory—that is, visual-spatial and phonological—so that they make sense to us.

Chai et. all (2018) provide an overview of several working memory-relevant studies in order to harmonize the findings of working memory from the neurosciences and psychological standpoints, especially after citing evidence from past studies of healthy, aging, diseased, and/or lesioned brains. The phonological loop appears to involve activation in the left hemisphere of the lateral frontal and inferior parietal lobes as well as the temporal lobe while the visuospatial sketchpad appears to activate slightly different areas like occipital, right and left frontal lobes and the parietal.

The central executive functions appear to involve activation mostly in the frontal lobes. Finally, the episodic buffer operations seem to involve the bilateral activation of the frontal lobes and portions of the temporal lobes, including the left hippocampus. The theory of Baddeley seems to have a strong basis.

The Call Assistant is equipped with an architecture that would follow somehow the same patterns enabling the interaction between the user and the system in a natural way, by using natural language. The system should also include common sense knowledge that people have and use, and a learning mechanism to enrich the structure of its knowledge towards drawing appropriate inferences (Kakkas and Michael 2016).

4.2.3 Thoughts in How People Create a Context and Reason on it (psychology perspective)

In our everyday life we are able to perform more "complex" and in a different way than these machines operate. We are able, based on our cognitive functions, knowledge and intuitions born out of a lifetime of experiences, to learn and to reason for events or situations that occur or will occur in the future. Learning, reasoning and argumentation play an important role in our daily routine and in fact affect our future. If we understand how these three functions are associated to each other then we can apply them into the machines in a more efficient way.

Reasoning and argumentation in humans is not a simple process as it seems. Byrne (1988) had tried to research how humans make inferences and how these inferences are produced under a certain contexts. In theory, as Byrne describes, when conditional sentences with a corresponding categorical premise are presented, then we have a production of corresponding mental rules that can be applied in order to have interferences. Two of these rules are the Modus Ponens inference and the Modus Tollens inference. These are rules of classical logic that can be used in human reasoning.

In brief, Modus Ponens inferences can be described as follows:

If X is true then Y is true.

X is true.

Therefore Y is true.

While Modus Tollens can be described as:

If X is true then Y is true.

Y is false.

Therefore X is false.

Byrne had researched a theory, the Semantic Model theory, regarding the question of how people make inferences of these two forms. According to the Semantic Model Theory there is a network of concepts with their relationships in which these inference rules can be applied.

However since our world is more complex and other conditionals could be appeared we have very often the phenomenon in which we do not have the application of these inference rules and therefore the appearance of unexpected inferences. For example when the researcher had conducted an experiment by presenting a second conditional containing an alternative condition for the same consequent, the results had showed that the expected inferences are often not made.

Byrne provides an alternative theory, the Mental Model Theory which assumes that people do not innately rely on formal rules of inference, but instead relies on their mental models which are constructed based on the understanding of the premises and their general knowledge. A conclusion is made if there is no other mental model which contradicts it. It seems that the suppression effect by introducing alternative or additional conditions results in subjects is constructing different sets of models, which provide different kinds of counterexamples.

Summarizing, according to the research, the production of inference is not a simple application of some logical rules like Modus Pones, but it is something more complicated, like our world and its states. This production is depended on variables that somehow are linked on a) how we understand the premises b) the quantity and the quality of knowledge we have and c) how this knowledge is applied in order to reason having in mind that each individual has their own perspective on the world.

But there also other researchers who tried to examine how we reason and how we argue. An earlier theory, named expected – utility theory (Prokop, 2021), had explore the effects of limited cognition and it was set as a benchmark theory which later, other researchers use for their research. However, a more recent research had shown violation of the predictions of this theory. For example, Zhou, S. (2016 :228) describes that Allais(1953) had shown that in some situations the actual behavior differs systematically from the predictions of the expected-utility theory, while Simon (1955) had argued that consumers "replace" the optimum solutions with acceptable solutions that satisfying a set of self-imposed constraints. The same violation indicated by Kahneman and Tversky (1979) and the prospect theory that had displayed that decisions which are made under risk are not necessarily optimal or rational. The prospect theory uses four elements, individuals derive utility from gains and losses relative to some reference point, individuals are more sensitive to losses that to gains (loss aversion), individuals exhibit diminishing sensitivity to gains and losses and individuals weigh outcomes subjectively, overweight low probabilities and underweight high probabilities(probability weighting).

Thaler (Thaler, 1985) had introduced the theory of mental accounting in order to understand the cognitive operations that individuals use for their economic activities. This theory explains how humans simplifying their economic environment in order to overcome their cognitive limitations and how the simplification can lead to suboptimal decisions. Key elements for this theory are the grouping of human expenditures in categories, the context, the framing and the situation in which a transaction will occur. Using the diminishing –sensitivity property the theory make predictions when outcomes are added together or separated before being evaluated. The evidence display that humans behave by the hedonic-editing hypothesis, thus during editing outcomes for maximizing their utility they segregate gains and integrate losses , to cancel small losses against larger gains and to segregate small gains from large losses. Thaler uses the acquisition (value of the item as a gift minus its price) and transaction(reference price or the difference between the actual price and the “fair” price that the item should have) utilities in order to establish the terms “good” and “bad” deal. The Thaler’s work on this field was the set point for other researches like Hastings and Shapiro (Hastings, Shapiro 2013) who showed the lack of fungibility of money or for Prelec and Loewenstein (Prelec, D., & Loewenstein, G. (1998) who through their prospective-accounting model had introduce the term “coupling” (degree which consumption calls to mind thoughts of payment and vice versa). Furthermore, Shefrin and Statman(SHEFRIN & STATMAN, 1985) had labeled the disposition effect (holding an account during loss in order to avoid closing the account and experience the total loss) while Read, Loewenstein and Rabin had (1999) introduced the term “choice bracketing” to describe the extent to which choices are separated or grouped together in mental accounting.

Humans, as the philosophers and social sciences, have indicated may fail to apply common sense. Furthermore, they have two “selves” to manage plans. The “present self” tries to save more in the future while the “future self” will prefer not fulfill the “present self” plans. Thaler proposed the planner-doer theory in order to model these selves. The planner is far-sighted while the doer is short-sighted. The planner applies rules that constraint doer causing a physic cost. This willpower to apply these constraints which has individual characteristics describes the different degrees of self-control. The planer-doer model looks similar with the brain’s function that neuroscience describes and it can be compared with the dual-process theories in psychology that claim the existence of two systems, system 1 for fast automatic decisions and system 2 for slower controlled and effortful decisions. Regarding the individuals, who do not act in their own best interest due to limited cognitive abilities and willpower the Thaler’s theory, displays answers by providing the design of the default options through nudging. These options are specified in advance by the agent who designs the decision problem. The individuals have just to apply these options accordingly. Nudging has two views, the paternalism” part that these options influence human’s behavior and the “libertarian” part that these options are not restrict people’s choices. Nudging has been applied in various areas like pension, education, health although it has faced some critique. A critical point is that these policies should be tested and evaluated thoroughly before their implementation.

All the above research indicates the measurement of the problem and the reason of why cognitive computing is so important. Probably because the cognitive computing is meant to extend the boundaries of the human cognition and the capabilities of the human brain

by using machines. Humans can reason and solve complex problems. But the human ability to read, analyze, and process huge volumes of data is quite poor. However, this is the strength of the computing intelligence and machine learning. It is about the creation of systems capable to reason and thinking similar to the ones that humans use, combining the strengths of human and machine into a collaborative situation.
Cognitive systems use techniques, such as machine learning, data mining, natural language processing, and pattern matching to mimic how a human brain works. Such systems are ideal to interact with an increasingly complex world (figure 21).

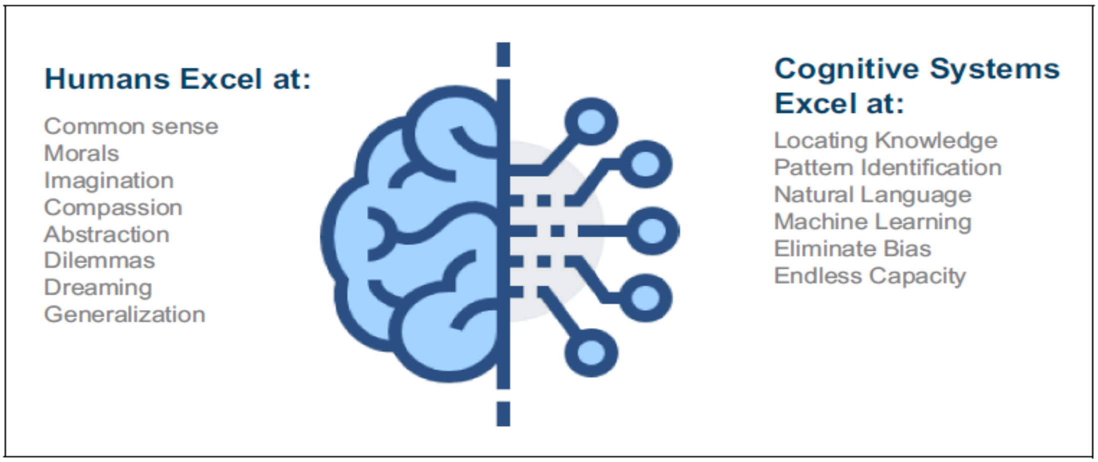


Figure 21 (IBM Watson Doc)

4.2.4 Decision Policy and Machine Learning-Why it is Based on Argumentation

The relationship of human reasoning and machine learning and coaching and how these processes are set are more transparent in the term cognitive programming that was introduced by Michael et al, (2015). Cognitive programming sets the level of interaction between the user and the system where the two understand and can anticipate the behavior of each other· The idea is that Cognitive systems should be able to learn, and be able to improve from their past interaction with the user offering personalized solutions. Key elements are the natural language which humans use for their communication as well as the ways they store, retrieve, and use common sense knowledge. The authors in their research argue among other things for the construction of a comprehension model, similar somehow to Byrne's mental models, which its task is the elaboration of the input narrative with new information, or inferences in order to capture the (or a possible) implicit meaning or intention of the narrative. They describe the comprehension model as coherent, that includes only inferences that are important for successful understanding, while it omits cluttering details and speculations.

One year later, Kakas & Michael (2016) explain in their research the term "programming paradigm for the masses" which was an initial solution for the communication between users and smart devices. Yet, the new developed cognitive systems "exceed" the role of an automated system and become more complex characterized by the ability to look like as a human personal assistant. How this can be realized? The answer is provided by the figure 22 : machine learning and decision making depending on "common sense" that humans have.

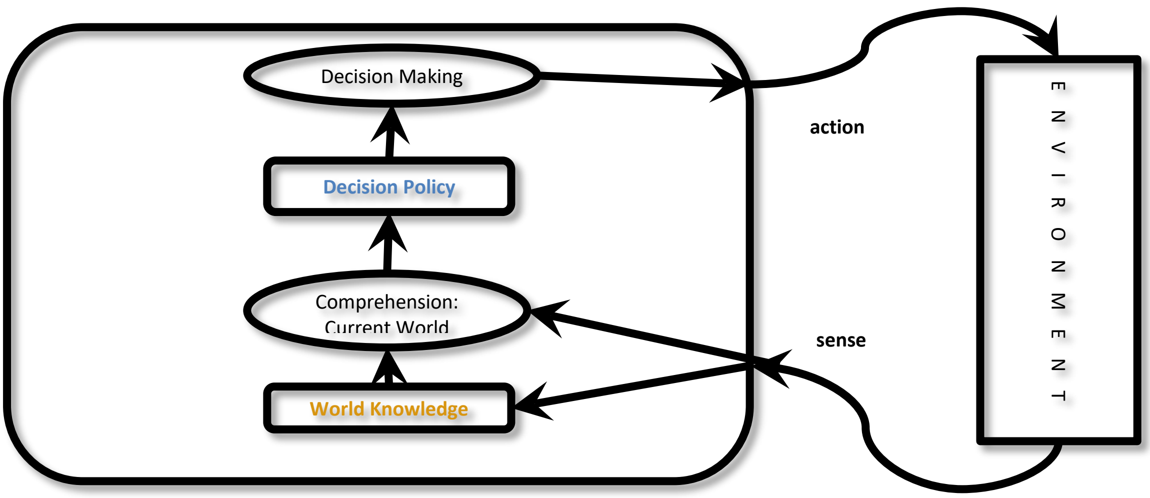


Figure 22

This picture can be read as follows: First, cognitive systems understand (sense) data by extracting information from available sources, like using natural language or image processing techniques. Then, they reason this data based on the application of machine learning and reasoning techniques to form hypotheses or generate suggestions. Finally, they have the ability of continuous learning from data insights and human interactions.

The idea behind of this design is the use of logic and what of we call common sense. Common sense is based on perception understanding and judgment and people with common sense are seen as reasonable, down to earth, reliable, and practical. Although, we see that people often make unreasonable decisions and perform actions that can be consider the opposite of what we call common sense. These two elements are used for the construction of arguments in order to produce representations of the environment with conclusions. In that way the system can use knowledge, verify facts, apply logical rules, change beliefs and accepts conclusions. Kakas & Michael (Dec 2016) had described the computational argumentation in Artificial Intelligence and had explained the various forms that it has as well as the frameworks under which this kind of arguments take place. More specific in case of Kakas & Michael (Dec2016), there is an explanation of abstract argumentation in which abstract arguments represent entities or a relationship. A rough description could be the following; the set of the abstract arguments is called A while the set of arguments' binary relation (conflicts) from A is called R (this is called attack relation). Besides "attacking" the arguments can "defend" other arguments by attacking back. The authors explain the terms preference – based argumentation and argumentation – based logic. In the first case, an argument may be preferred to another one when it is stronger or its beliefs have a higher probability. In the second case, the argumentation deals with inconsistent information and its arguments attack each other while there is an evaluation of arguments. In this paper, the researchers argue how the argumentation and the human decision are "embedded" in cognitive systems. They describe what are the cognitive assistants and some of their potentials and they raised issues such as the decision making procedure in open and dynamic environments. This is very important because cognitive systems perform in open and dynamic environments mostly. Decision making is a key element for cognitive assistants' efficiency but the decision making is depended by argumentation. Therefor argumentation and decision making go together making the argumentation the basis for the cognitive assistants' optimum operation.

The authors also research the topic of argument and narrative comprehension and they had explained what are the belief arguments and the differences between beliefs and preferences. As the authors had described the belief arguments capture aspect of the world while preferences capture the commonsensical reasoning pattern that humans use to form a coherent understanding of the situation. There is also a description of reasoning about actions and change (RAC). This is concerned of how some of the properties of our problem domain change when new information is acquired and how this view of the problem world is affected by actions that have been performed and/ or by other actions which are blocked from materializing at a particular stage or time in the flow of change.

Regarding the formulation of decision problems there is a "protocol" that it must be followed. First we have the options, these consist the set of actions or in some cases the explanations that are available when a condition or a situation is occurred. Second, we have the scenario information or in brief scenarios that describe the possible states of the environment. The system can form the scenarios by "observing" the environment either by observing it directly or by making hypothetical assumptions that describe some possible states. These scenarios must be plausible avoiding in that way the inconsistency between the environment's states. Third, we have scenario-based preferences (S;O) where a scenario S is "linked" with a corresponding subset O of options. These scenarios can be grouped in hierarchies of increasing specificity.

This framework supports generality because it can perform intelligent behavior in a broader environment which can include vast number of domains with a variety of tasks. The system can use sub-systems for each sub-domain covering the broader environment. For example, suppose we have a system regarding the domain of tourism in general. This system can include sub -systems for concepts that are related with this domain like accommodation, museums, sightseeing, transportation, excursions etc. It can be as general as possible while some of the sub systems, with minimal redesigns, can be used in other similar domains (for instance the domain entertainment can include sub systems like museums sightseeing, transportation, and excursions. This example also indicates the high level of versatility because the procedure of re-design would demand relative limited effort, resources and time. Regarding taskability the framework can manage as many tasks as the developer can "describe" through the procedure of the creation of the scenarios with their preferences or priorities. Summarizing, the argumentation –based cognitive architecture can produce a system as general as possible, capable to perform a large number of diverse tasks with a relative minimum effort on the developer's part.

The criterions of rationality and optimality seem to have high value in this case, because in any case (scenario) the system has the knowledge according to each case to select the optimum action(s) which lead to one its goals. Each scenario is constructed in that way so it uses knowledge, arguments, entities and relationships and selects the optimum actions that lead to a goal. The only possible drawback is in case that the scenarios or the preferences or even the actions are not being "described" or stated correctly so in this case the system will not perceive the right dimension of the environment or it might choose to perform a wrong action. Critical part at this point is the acquisition and elicitation of the requirements that might prevent situations like these.

The system is efficient in terms that it satisfies all its constraints on time and space, as in work on real time system but again the scenarios and the arguments must be set in an efficient way. In case of scalability, reactivity and persistence it seems that there is a drawback since the system's development must be executed incrementally in order to adapt new or changed requirements succeeding a continuous learning process that is linked with this development.

A note from the authors is that they had not included in their research the connectionist approaches (this approach rejects the use of explicit rules and symbols and use distributed representations, in which concepts are characterized as patterns of activation in the network) although they had admitted that these approaches served an important role in large-scale architectures like Sun, Merrill etc

Kakkas, Moriatis & Spanoudakis (2019) had presented the development that has occurred in the field of argumentation with the Gorgias preference – based argumentation framework of Logic programming with Priorities. Additionally, apart from the explanation of the Gorgias's framework there is a review of a general methodological approach for developing "decision making" applications of argumentation and the brief report about two real life applications in two domains: an eye-clinic assistant and data access and sharing assistant.

More specific, in the first section (introduction), the authors argue for the importance of argumentation over a wide range of problems such as the debates on online social interaction settings or in the automating legislation. Following, they state some argumentation systems like CaSAPI, DeLP, TOAST and Gorgias that support the study for this kind of problems and their applications while they indicate that decision problems occur in dynamic and incomplete or uncertain environments by giving several examples of problem which support their claim (legal problems, medical problems etc.). The researchers' approach for developing applications based on argumentation has three, as they state, challenges. First is the acquisition and elicitation of the requirements that need to be carried out at a high level akin to the natural cognitive level of those whom the application is built. Second, is the system's development that must be executed incrementally in order to adapt new or changed requirements succeeding a continuous learning process that is linked with this development. Third, the system must explainable to people, something which, now, is required by law in European Union. The researchers support that their approach, which follows the preference –based structured argumentation frame of Logic Programming with Priorities and its Gorgias implementation, remedy these kinds of issues. Finally they introduce a new tool, the Gorgias-B, a novel human-machine interface that supports their proposed approach.

In the second section, the authors introduce the general terms that are used in order to formulate decision problems. First we have the **options**, these consist the set of actions or in some cases the explanations that are available when a condition or a situation is occurred, for example some options for someone who wants to work out in the gym are: to workout, to work out with a friend, to work out with a trainer or not to workout at all, etc. In this case, the trainee according to his/her mood decides to choose and follow one of these available options, in other words, to act accordingly. Second, we have the **scenario information** or in brief **scenarios** that describe the possible states of the environment. For instance, using the previous example, possible scenarios could be that the trainee visits the gym for the first time, the gym offers personal training free of cost for a month or today the gym is closed. The system can form the scenarios by "observing" the environment either by observing it directly or by making hypothetical assumptions that describe some possible states. In another part of the paper, the authors argue that these scenarios must be plausible avoiding in that way the inconsistency between the environment's states. Third, we have **scenario-based preferences (S;O)** where a scenario S is "linked" with a corresponding subset O of options, so again in the previous example, when the scenario states that the gym is closed today then the linked option is not workout at all. The authors' advice is to group these scenarios in hierarchies of increasing specificity.

Following, we have the introduction of the notation that is used for the above terms and the introduction of two other terms, the refinement of a scenario and the combination of scenarios. When a scenario S can be expanded by using some information C then we have a new scenario S' (S' = S U C) which is called refinement of S. Accordingly, when we have two initial scenarios S_a and S_b that can be combined (S_{ab} = S_a U S_b) then we have their combination that consists a new scenario obtained by their set union. The refinement has a result the hierarchy of scenarios which, as the authors had argued, increases the specificity. Finally, we have the scenario based preferences (SP =(S;O)) that can be focused or sharpened by more 'specific" SPs (SP' =(S';O')) so in that way we have the hierarchy of scenario based preferences.

All the above terms are explained by the researchers by using a simple example for an on line shopping assistant.

The authors briefly describe the Georgias argumentation framework and how the problems can be formulate as an argumentation theory in it. The framework uses the Modus Ponens argument scheme. They talk about beliefs (conditions that are defeasible) object-level arguments (when the claim of an argument is a literal on an option or belief predicate) attacks (when an argument attack each other supporting contradictory claims) priority arguments (give relative strength between arguments) and their role that is to tighten the attack relations. Furthermore they describe the terms acceptability, composite arguments, attack relations admissible arguments and what consists a "good" solution and a "best or optimal' solution. They use the on line assistant example to provide the argumentation framework of the Gorgias and they give the central algorithm for argument generation. The algorithm "translates" the arguments in an automatic way and generates the rules using a hierarchy manner make them invisible to the user. In this section there is a short description about Gorgias system while they present a number of real-life application problems (such as Deep Venus Thrombosis , Ambient Assisted Living etc) that have been studied with the Gorgias argumentation framework and they provide solutions in different fields, such as medicine, networking, diplomacy, product pricing image analysis and others.

They also explain the Gorgias argumentation based approach for investment on an asset (add or not an asset in the investment portfolio, the name of the tool is PORTRAIT) They give the options fund or not (select (Fund) and not select (Fund) and the scenario information . The system provides the primary scenarios which are then refined and combined in order to product some scenario based preferences.

There is a presentation of the manner that we have the interaction between the domain expert and Gorgias framework. This interaction uses a table in which the domain expert adds in the table's columns the possible options and in the table's rows the scenarios in which the different options are enable or are preferred. Once there is identification of language of options and the description of the scenarios then, the expert executes two steps. The first step is to name the columns and to "link" each scenario with the corresponding option(s) while the second step is to execute a conflict analysis and to provide resolutions to possible conflicts that would arise from the available options since we might have different scenarios that would valid simultaneously. In case we have more conflicts new row with refined scenarios can be added because the table is built incrementally and its construction follows the two steps every time. The authors give directions about the arguments view that Gorgias-B uses in order to construct the table and to generate the scenario based preferences.

In section six the authors describe two real life applications that were built up by Gorgias-B tool, the Eye Clinic (first level) Support application and Data Access and Data Sharing application. The authors conclude that this approach is sufficient general to allow the development of applications in any of the many argumentation frameworks that support conditional and hierarchical forms of preference. Yet, they state that this methodology can be improved giving some ideas for example of how the preferences can be extract without the presence of the domain expert. At the end the authors give a hint about the new system Gorgias-NL which is under development where the preferences would be extracted directly from dialogues with the user in structured forms of natural language.

Other researchers like John Fox (Fox, J. 2011:2) comment the role of inference in cognitive agents and they use argumentation by using similar ways. More specific Fox examines the term “signature” that specifies how one set of sentences (e.g., propositions) is entailed by another set of sentences (e.g., a database of propositions and rules)

Example of signature : (Database / Conclusion) L Inference

So we can say that Conclusion can be validly inferred from Database under the axioms of inference system L

More complicate tasks like decision making and planning require a more complex signature. For example in case of a cognitive agent in the medicine field we need a reasoning model in which a general medical knowledge is applied to specific patient data by arguing the pros and cons of alternative ways of achieving clinical goals, for example:

((Knowledge U Data) / (Claim, Grounds, Qualifier)) LA Argumentation

This formulation makes the arguments structure explicit where the term Claim represents the conclusion, the term Grounds represents the justification and the term Qualifier represents the confidence or the uncertainty . Like humans, agents may have arguments that “compete each other” increasing or decreasing the Confidence. A qualifier may indicate that an argument “supports” or “opposes” a claim.

So we have:

- the Evidential mode : more arguments that support (or against) a claim create more (or less) confidence.

- The Dialectical argumentation, promotes the "discussion" and "debate" with other agents allowing the arguments' contradiction.

As Fox argues, argumentation theory may therefore offer insights into the kinds of cognitive agents interactions. For instance Logic of Argumantation is used for PROforma, a language for modelling cognitive agents (Fox & Das 2000; Fox et al. 2003) which has been used to develop many practical decision tools, notably in medicine.

According to Fox, argumentation theory clarifies and somehow defines the term "evidence" as it is used in legal, medical, scientific, and other kinds of reasoning and in everyday decision-making and evidence-based discussions.

Other researchers, (M. Mozina, etc all 2009:18-23), had discussed the use of arguments in interaction between machine learning and cognitive agents and the concept of Argument-based machine learning (ABML). Mozina (Mozina, M. 2018:53). had provided other studies such Fails et al. (2003) in which they used the term interactive machine learning to describe an iterative system for correcting errors of an image segmentation system. Besides better performance research reports that users also gain trust and understanding of their systems. This claim is supported by Stumpf et. al. (2009), (Mozina, M. 2018:53) where a user can comment on automatically generated explanations provided by a learned model. These comments are then used as constraints in the system when relearning the model. This in fact, is what we call machine coaching because the users identify "bugs" in a system by inspecting explanations and then explain necessary corrections back to the system. It is a case of learning from feedback, data and prior knowledge, where prior knowledge is represented with arguments.

An example of machine learning and coaching is the following; a machine call assistant might argue that a phone call must be answered in any case because the caller is a close relative. The stakeholder could then counter argue that the call could be answered in any case when it comes from a close relative but it is also characterized as an emergent. Then, the system should induce a new model for call, which would state emergency (among others) as the reason for answering the call in any case.

An advanced aspect of machine learning and coaching is the Argument mining (AM) which includes aspects of natural language processing and understanding, information extraction, feature discovery and discourse analysis. As Lippi & Torroni state the relations between premises and claims, or between different arguments can instead be easily modeled as a task of link prediction within a graph, where nodes represent arguments or argument components (Lippi & Torroni 2016:A18-A19). Furthermore all the argument mining frameworks proposed so far can be described as multi-stage pipeline systems, whose input is natural, free text document, and whose output is a mark-up document, where arguments (or parts of arguments) are annotated. Each stage addresses a sub-task of the whole AM problem, by employing one or more machine learning and natural language processing methodologies and techniques(Lippi & Torroni 2016:A5).

Chapter 5
Implementation

In this chapter, we present the implementation of the Call Assistant. The structure of the chapter is as follows: in section 5.1, we present the implementation of sensing of the user's physical actions and the environment; in section52, we present the implementation of the communication procedure between the user and the assistant; in section 5.3, we present the knowledge base of the Call Assistant in section 5.4, we present the creation of the phone calls' context; in section 5.5, we present the decision policy, how it is applied and how the machine learning through the user's coaching is achieved.

5.1 Sensing the User's Physical Actions and Environment

As we saw in the previous chapter, the system should know the answers in the following questions:

1. Where is the user when he receives the call?
2. What the user does when he receives the call?

The function of sensing tries to answer these two questions whose answers create a part of the context of call. As we saw, based on the context, which is created in every phone call, we have the application of the decision policy making this creation critical for optimum performance of the assistant.

One of the first things, which the assistant needs to know, is the user's physical activity. This is implemented by using the Google's Activity Recognition API. This API automatically detects activities by periodically reading short bursts of sensor data and processing them using machine learning models.

The API can identify the following physical activities:

IN_VEHICLE	The device is in a vehicle, such as a car.
ON_BICYCLE	The device is on a bicycle.
ON_FOOT	The device is on a user who is walking or running.
RUNNING	The device is on a user who is running.
STILL	The device is still (not moving).
TILTING	The device angle relative to gravity changed significantly.

UNKNOWN	Unable to detect the current activity.
WALKING	The device is on a user who is walking.

The assistant displays a confidence property which indicates the likelihood that the user is performing the activity represented in the result.

Opening the application we have the creation of a service connected to play activity recognition. The service manages the connection and requests activity updates using a PendingIntent. The PendingIntent fires an IntentSrevice to handle the activity updates from Android and post broadcasts back to the application's MainActivity to update the activity and confidence level(which in this case is set to 60%).

To set up Activity Recognition we have to edit the manifest by declaring the services we want implement (we implement two services). The app also has to request the user's permission to use ACTIVITY_RECOGNITION as shown below.

```
<uses-permission
android:name="com.google.android.gms.permission.ACTIVITY_RECOGNITION" />
<service
    android:name=".DetectedActivitiesIntentService"
    android:exported="false" />
<service android:name=".BackgroundDetectedActivitiesService" />
```

The assistant tries to identify the user's activities in the "background" (when the user pushes the app into the background and it's not visible) and perform an action when a specific activity is detected -- for example, high confidence of walking.

```
private void startTracking() {
    Intent intent1 = new Intent(MainActivity.this,
BackgroundDetectedActivitiesService.class);
    startService(intent1);
}
```

In the code below we extract the result from the *intent* then pop of each activity and broadcast the result to the main activity.

```
@Override
protected void onHandleIntent(@Nullable Intent intent) {
    ActivityRecognitionResult result =
ActivityRecognitionResult.extractResult(intent);

    // Get the list of the probable activities associated with the current state of
the
    // device. Each activity is associated with a confidence level, which is an int
between
    // 0 and 100.
    ArrayList<DetectedActivity> detectedActivities = (ArrayList)
result.getProbableActivities();

    for (DetectedActivity activity : detectedActivities) {
        Log.e(TAG, "Detected activity: " + activity.getType() + ", " +
```

```
activity.getConfidence());
        broadcastActivity(activity);
    }
}
```

The main activity gets each activity returned in the list (type and confidence interval) in turn displays those values on the UI. In this case, the requested update period is 30 seconds (could be faster or slower) then we see the screen update to one of the activities.

```
private void handleUserActivity(int type, int confidence) {
    String message = "I believe that you are ";

    switch (type) {
        case DetectedActivity.IN_VEHICLE: {
            label = "In vehicle";


            break;
        }
        case DetectedActivity.ON_BICYCLE: {
            label = "On bike";

            break;
        }
        case DetectedActivity.ON_FOOT: {
            label = " On foot";

            break;
        }
        case DetectedActivity.RUNNING: {
            label = " Running";

            break;
        }
        case DetectedActivity.STILL: {
            label = " Still";
            break;
        }
        case DetectedActivity.TILTING: {
            label = " Tilting";

            break;
        }
        case DetectedActivity.WALKING: {
            label = " Walking";

            break;
        }
        case DetectedActivity.UNKNOWN: {
            label = " Uknown activity";
            break;
        }
    }

    Log.e(TAG, "User activity: " + label + ", Confidence: " + confidence);

    if (confidence > Constants.CONFIDENCE) {
        txt_activity.setText(message + label);
        txt_confidence.setText("Confidence: " + confidence);
```

```
        activity = label;
    }
}
```

The next thing which this assistant does is the identification of the current location of the user. In other words, the assistant tries to sense where is the user in order to answer the first question (Where is the user when he receives the call?) so it would be able to create the answer for the second (What the user does when he receives the call?).

The steps of the identification and the display of the current location are presented in the figure 22.

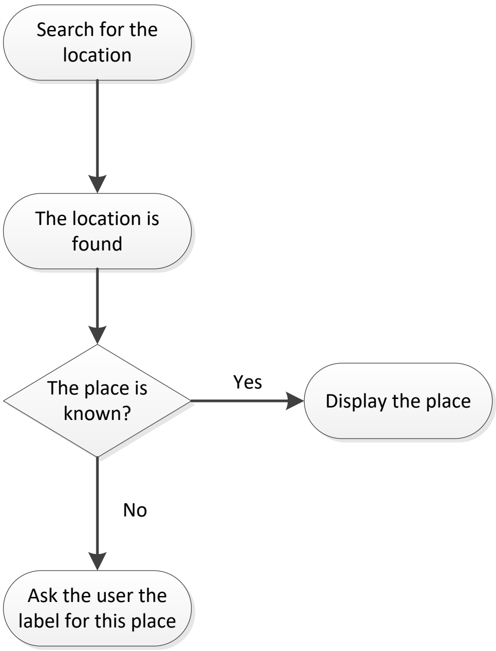


Figure 22

The app has to request the user's permission to use location and we have to edit the manifest. Depending on the LocationManager methods used, either ACCESS_COARSE_LOCATION or ACCESS_FINE_LOCATION permission is needed. For example, we need to declare the ACCESS_COARSE_LOCATION permission if our application uses a network-based location provider only. The more accurate GPS requires the ACCESS_FINE_LOCATION permission. The ACCESS_FINE_LOCATION permission implies ACCESS_COARSE_LOCATION already. Since we use a network-based location provider we have also to declare the internet permission. So the manifest has to include

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

When the application initializes then the method getLocation() is applied

```
private void getLocation() {
    try {
```

```
        locationmanager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);



        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_COARSE_LOCATION)
!= PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode, String[]
permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See the
documentation
            // for ActivityCompat#requestPermissions for more details.
            return;
        }
        locationmanager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER,
500, 5, (LocationListener) this);

    } catch (SecurityException e) {
        e.printStackTrace();
    }
}
```

Every time which the location is change we use the method

```
@Override
public void onLocationChanged(Location location) {
    if(location!=null) {
        try {
            Geocoder geocoder = new Geocoder(getApplicationContext(),
Locale.getDefault());
            List<Address> addresses =
geocoder.getFromLocation(location.getLatitude(), location.getLongitude(), 1);

            locationarea = addresses.get(0).getAddressLine(0);
            lat = addresses.get(0).getLatitude();
            lng = addresses.get(0).getLongitude();
            lati = Double.toString(lat);
            longi = Double.toString(lng);
            txt_location.setText(locationarea);


            if ((!label.equals(" Running")) || (!label.equals("On bike")) ||
(!label.equals(" Walking")) || (!label.equals(" On foot")) || (!label.equals("
Tilting")) || (!label.equals("In vehicle"))) {

                searchPlaceAndSpeak();
            }
            Intent in = new Intent();
            in.setAction("my.action");
            in.putExtra("activity", label);
            in.putExtra("location", locationarea);
            in.putExtra("place", txt_place.getText().toString());
            sendBroadcast(in);

        } catch (IOException e) {
```

```
            e.printStackTrace();
        }

    }

}
```

This method is has in mind the physical activity of the user, so if the user is for example running then it does not apply (because we do not want the assistant to ask the user, during running, for the each location's label, it is not efficient and productive). Both Google Api and Location Manager are cooperate together in order to perform as best as possible When the application identifies a location it checks if it has a label for it, for instance office, home or my gym etc. In case there is a label it displays it otherwise it asks the user to provide a label as it shows in figure 22. After providing the label, the user can add action(s) which are conducted in this place. Figure 23 displays the whole concept of sensing the environment and user's activities.

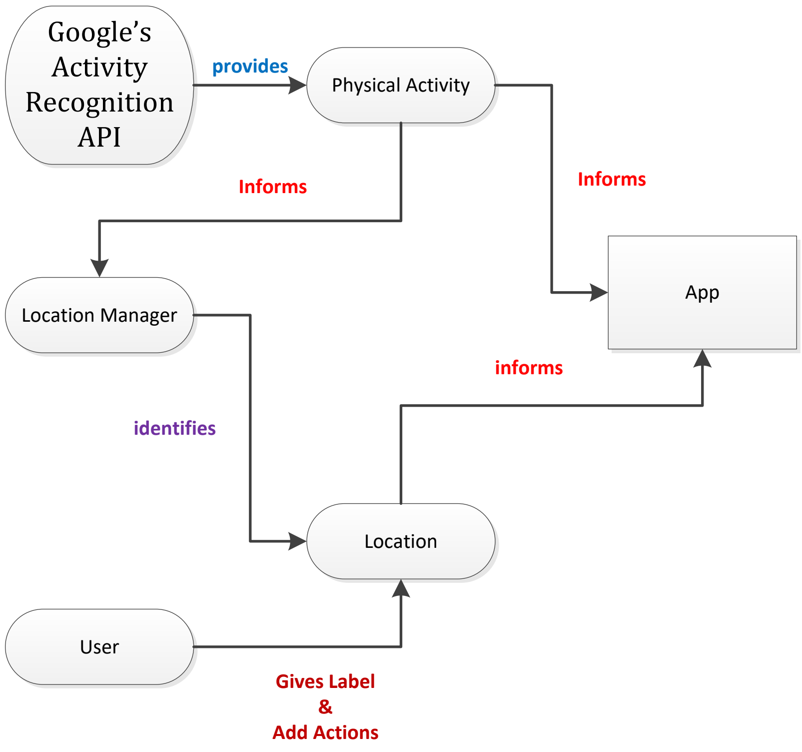


Figure 23

When the user adds an action this action is linked with the place, it has a time period(start and end time) and a schedule which is characterized by selecting the corresponding option among the values :

- Monday to Friday
- Weekend

- Everyday
- Today

The assistant holds all the places, for each place holds a list of action that the user has informed it and for each action there are available the time frame and the schedule (figure 24)

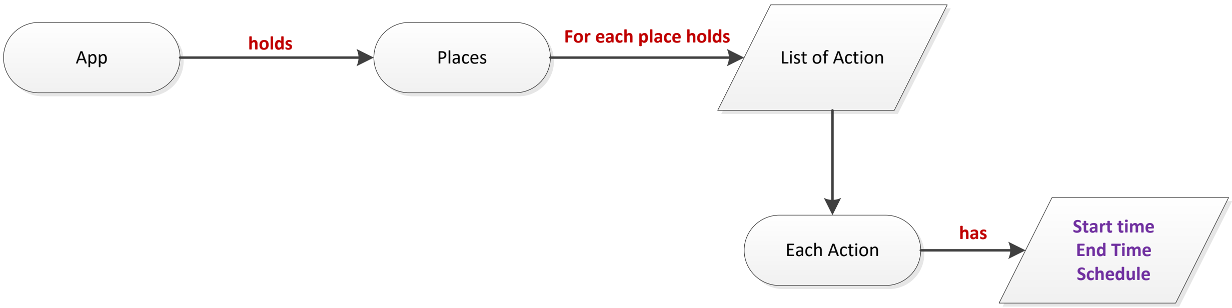


Figure 24

The assistant is capable to choose the action, among the list of actions, which is currently valid. This is done by making a use of the function with signature

```
private void searchActions(String place, String day, String time)
```

when the searchActions receives the place, the current day and time

searchActions(placearea, currentday, currenttime);

returns the action which is executed on this place and period of time

Summarizing through sensing the Call Assistant can identify the physical actions of the user, the location and the list of actions (figure 25).

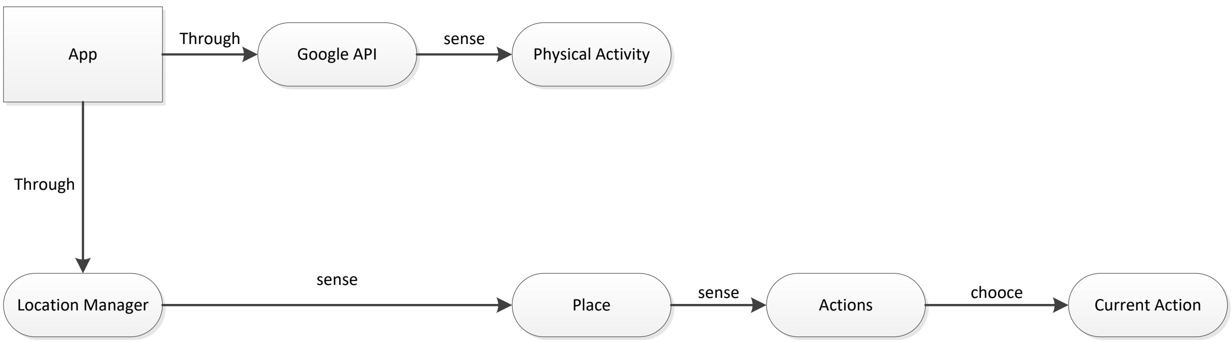


Figure 25

Final, a component that also assists the Call Assistant to sense the environment is a Broadcast Receiver. A broadcast receiver is one of the basic components of Android and it can be used as a messaging system across apps and outside of the normal user flow. For example, the Android system sends broadcasts when various system events occur, such as when the device starts charging or about the device boot. Apps can also send custom broadcasts such as to notify other apps that something that they might be interested which is occurred. In this case, this component is used to sense the incoming phone calls and to create the context in which we have the application of the decision policy. More details about this component we will display in the description of how we create the context of a call.

5.2 Communication Procedure between the User and the Assistant

The communication between the user and the assistant is conducted through the use of Text-to-Speech (TTS) capability based on an API which launches Google's Speech Recognition service and returns back the text result for us. This capability, also known as "speech synthesis", enables our Android device to "speak" text of different languages. In other words, the user uses natural language which the assistant, through the use of the TTS capability, converts to text. All the dialogues between the user and the assistant are conducted by using natural language so when the assistant needs some piece of information asks the user, by using natural language which the user can hear through the mobile's speaker and the user responds using the microphone of the device. All the answers are converted to text and then are store in the knowledge base of the assistant.

First we need to create a RecognizerIntent by setting necessary flags such as *ACTION_RECOGNIZE_SPEECH* – Simply takes user's speech input and returns it to same activity.

LANGUAGE_MODEL_FREE_FORM – Considers input in free form English

EXTRA_PROMPT – Text prompt to show to the user when asking them to speak

The assistant when needs information triggers Android's Speech-to-Text Intent and displays a dialog that indicates that your app is ready to accept speech input. Once the user has finished speaking, the dialog will close automatically and ACTION_RECOGNIZE_SPEECH will send the recorded audio through a speech recognizer and the input will be converted into text, and then displayed as part of a TextView.

The button that acts as a microphone executes the following method

```
public void btnSpeech (View view) {
    Intent intent = new Intent((RecognizerIntent.ACTION_RECOGNIZE_SPEECH));
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.ENGLISH);
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Hello, please advice me");
```

```
    try {
        startActivityForResult(intent, 1);
    } catch (ActivityNotFoundException e) {
        Toast.makeText(this, e.getMessage(), Toast.LENGTH_SHORT).show();

    }
}
```

An advance is that Speech-to-Text doesn't require an active internet connection, so it'll work correctly even when the user is offline.

5.3 Knowledge Base

The knowledge base is designed to hold all the necessary information in order to help the assistant to create the context of each call and to apply the decision policy with the most productive manner. All data are hold in the user's device so there is no issue about transmitting information to a third part for processing or storing needs. This approach fulfills the demands of the GDPR rule about personal information and data.

The application is make a use of Android SQLite database. It is a very lightweight database which comes with Android OS. The reasons for the use this king of database is that combines a SQL interface with a very small memory demands and a decent speed. Once a database is created successfully its located in **data/data//databases/** accessible from Android Device Monitor.

SQLite is a typical **relational database**, containing tables (which consists of rows and columns), indexes etc. in which we can create our schema of tables and structure.

Android has features available to handle changing database schemas, which mostly depend on using the SQLiteOpenHelper class.

Using the SQLiteOpenHelper we can create the tables of data while in case of upgrade to a newer schema we will have option to alter the database schema to match the needs of the rest of the app. The assistant makes a use two of the method of the SQLiteOpenHelper. The **onCreate(SQLiteDatabase db)** . This is executed when there is no database and the app needs one. It passes us a SQLiteDatabase object, pointing to a newly-created database, that we can populate with tables and initial data. The second method is **onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)** : It's called when the schema version we need does not match the schema version of the database, It passes us a **SQLiteDatabase** object and the old and new version numbers.

The two methods of the Call Assistant are displayed here:

```
@Override
public void onCreate(SQLiteDatabase db) { // create the database and the table my_incoming_calls
   String query = "CREATE TABLE "+ TABLE_NAME +
         "(" + COLUMN_PERSON+" TEXT )";

   db.execSQL(query);

   String query1 ="CREATE TABLE " + TABLE_NAME1 +
         "(" + COLUMN_PLACE1+ " TEXT, "+COLUMN_ADDRESS1+" TEXT )";

   db.execSQL(query1);

   String query2 ="CREATE TABLE " + TABLE_NAME2 +
         "(" + COLUMN_PLACE2+ " TEXT, "+COLUMN_ADDRESS2+" TEXT, "+COLUMN_ACTIVITY2+" TEXT,
"+COLUMN_DAY2+ " TEXT, "+COLUMN_START_TIME+" TEXT, "
         +COLUMN_END_TIME+" TEXT, "+COLUMN_SCHEDULE+" TEXT )";

   db.execSQL(query2);

   String query3 ="CREATE TABLE " + TABLE_NAME3 +
         "(" +COLUMN_CONTACTNAME3+ " TEXT, "+COLUMN_PHONENUMBER3+ " TEXT, "+ COLUMN_DAY3+ "
TEXT, "+COLUMN_TIME3+" TEXT, "+COLUMN_ADDRESS3+" TEXT, "+COLUMN_PLACE3+" TEXT,
"+COLUMN_ACTIVITY3+" TEXT )";

   db.execSQL(query3);

   String query4 ="CREATE TABLE " + TABLE_NAME4 +
         "(" +COLUMN_CONTACTNAME4+ " TEXT, "+COLUMN_PHONENUMBER4+ " TEXT,
"+COLUMN_RELATION4+" TEXT )";

   db.execSQL(query4);

   String query5 ="CREATE TABLE " + TABLE_NAME5 +
         "(" + COLUMN_RULE_ID+ " INTEGER PRIMARY KEY AUTOINCREMENT, "+COLUMN_ARGUMENT+"
TEXT )";

   db.execSQL(query5);

   String query6 ="CREATE TABLE " + TABLE_NAME6 +
         "(" +COLUMN_DATE6+ " TEXT, "+COLUMN_DAY6+ " TEXT, "+COLUMN_PERIODOFWEEK6+ " TEXT,
"+COLUMN_TIME6+ " TEXT, "+COLUMN_PERIODOFTIME6+ " TEXT, "+ COLUMN_PLACE6+ " TEXT,
"+COLUMN_ACTIVITY6+" TEXT, "+COLUMN_SCHEDULE6+" TEXT, "+COLUMN_START_TIME6+ " TEXT,
"+COLUMN_END_TIME6+" TEXT, "
         +COLUMN_CONTACTNAME6+" TEXT, "+COLUMN_PHONENUMBER6+" TEXT, "+ COLUMN_RELATION6+
" TEXT," +COLUMN_PHYSICAL_ACTIVITY6+ " TEXT,"+COLUMN_NUMBEROFCALL6+ " TEXT,"
         +COLUMN_TIMEPREVIOUSCALLS6+ " TEXT, "+COLUMN_TIMEDIFFERENCE6+ " TEXT, "
+COLUMN_CALLREVIEW6+" TEXT )";

   db.execSQL(query6);

   String query7 ="CREATE TABLE " + TABLE_NAME7 +
         "(" +COLUMN_CONTACTNAME7+" TEXT, "+COLUMN_PHONENUMBER7+" TEXT, "+
COLUMN_RELATION7+ " TEXT," +COLUMN_CALLREVIEW7+" TEXT, "+COLUMN_REASON7+" TEXT )";

   db.execSQL(query7);

   String query8 ="CREATE TABLE " + TABLE_NAME8 +
         "(" + COLUMN_WORD8+ " TEXT, "+COLUMN_VARIABLE8+" TEXT )";
```

```
    db.execSQL(query8);


    String query9 ="CREATE TABLE " + TABLE_NAME9 +
            "(" +COLUMN_RULE_ID9+ " INTEGER PRIMARY KEY AUTOINCREMENT, "+COLUMN_RULE9+ " TEXT,
"+COLUMN_FEEDBACK9+ " TEXT, "+COLUMN_VARIABLES9+ " TEXT, "+COLUMN_CALL_ACTION9+" TEXT )";

    db.execSQL(query9);

    String query10 ="CREATE TABLE " + TABLE_NAME10 +
            "(" + COLUMN_CONTACTNAME10+ " TEXT, "+COLUMN_PHONENUMBER10+" TEXT, "+
COLUMN_DAY10+ " TEXT, "+COLUMN_TIME10+" TEXT, "+COLUMN_ADDRESS10+" TEXT,
"+COLUMN_PLACE10+" TEXT, "
            +COLUMN_ACTIVITY10+" TEXT, "+COLUMN_RULE_ID10+ " INTEGER, "+COLUMN_RULE10+ " TEXT,
"+COLUMN_FEEDBACK10+ " TEXT, "+COLUMN_CALL_ACTION10+" TEXT )";

    db.execSQL(query10);


}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);// stakeholder
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME1); // place
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME2); // actions
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME3); // calls
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME4); // relationships
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME5); // arguments
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME6); //context
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME7); //call review
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME8); //vocabulary
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME9); //machine learning
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME10); //call_log
    onCreate(db);
}
```

The knowledge base has the following schema (figure 26)

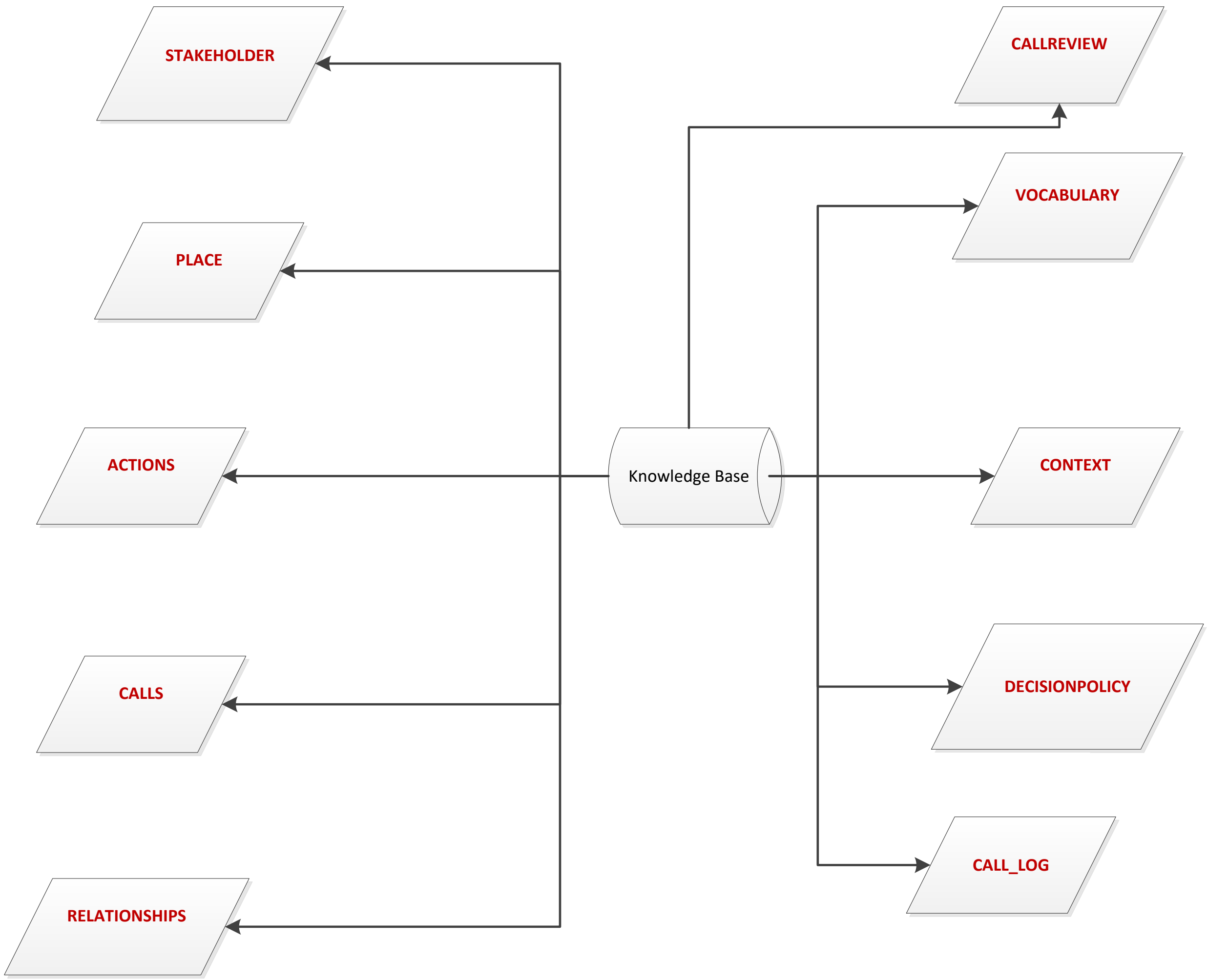


Figure 26

More specific:

In the table Stakeholder the assistant holds the name of the user

In the table Place the assistant holds information about the place such as the address

In the table Actions the assistant holds information about each Action like where this action is take place, day, time, schedule etc.

In the table Calls the assistant holds information for each call like the name of the caller, the phone number, the day , time and the place where the user was during the phone call

In the table Relationships the assistant holds information about the status of the caller (family, friend etc.)

In the table CallReview the assistant holds information about the type of the call (important, annoying etc)

In the table Vocabulary the assistant holds all the worlds that it needs in order to construct the decision policy

In the table Context the assistant holds the context of each call

In the table DecisionPolicy the assistant holds the rules with the advice from the user

In the table Call_Log the assistant keeps a log.

The figure 27 displays how the assistant operates and improves his knowledge base

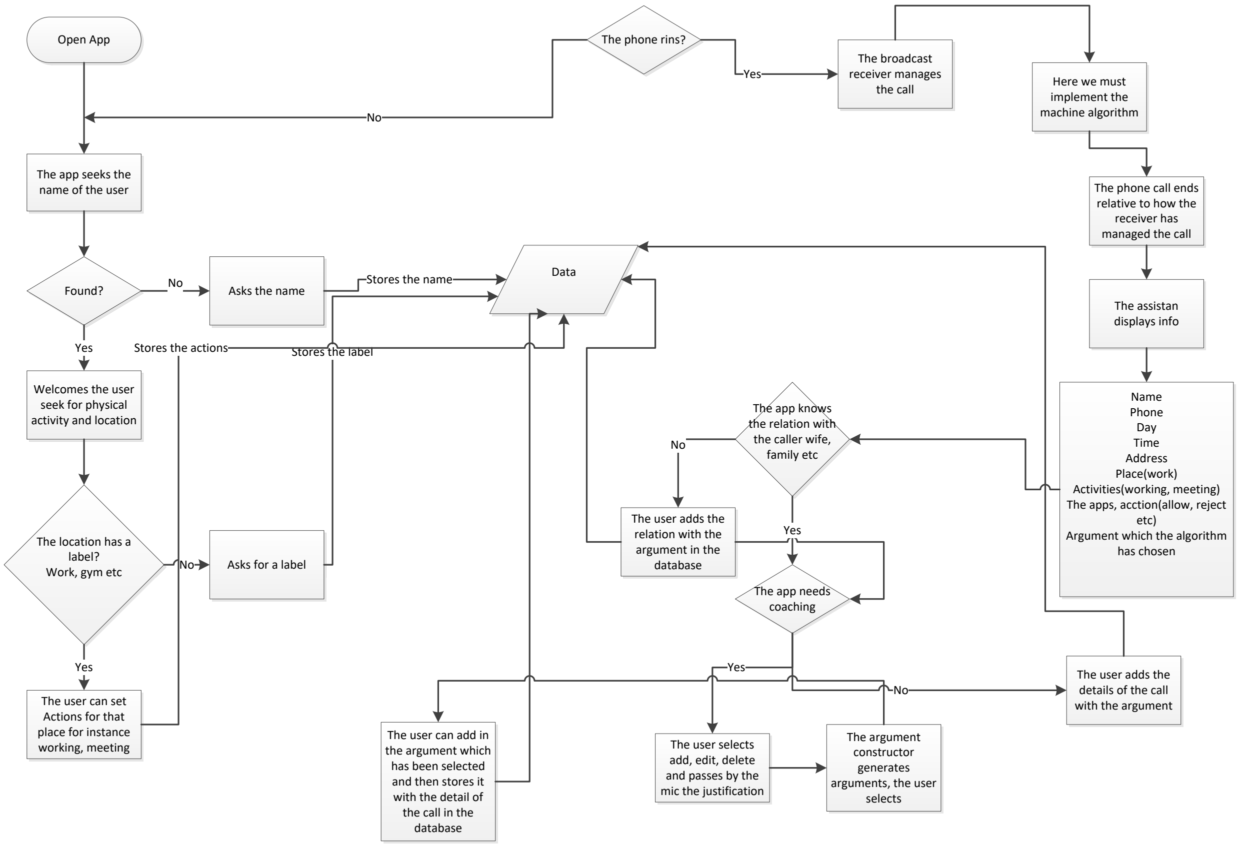


Figure 27

5.4 Context of Calls

The term context includes the perception of the environment, the circumstances and information which are needed for making a decision. As we have described in the previous chapter the following questions create the context:

1. Who is calling? (The person who calls carries a “status”, a “weight”, which affects our decision to answer or not, for instance if the person is a family member, a mother or a child with health issues.)
2. What time we receive the call? (The time of calling combing with other facts can characterize the call either important because we expected it, or insignificant and even annoying because on the other line is someone who acts on antisocial behavior.)

3. Where is the stakeholder when we receive the call? (There are times, during the day in which the user cannot answer the phone, like being in an important job meeting or being in a doctor's office for medical examination.)
4. What the stakeholder does when we receive the call? (For instance, during driving a car, according to law, the user cannot answer the phone unless he/she uses Bluetooth hands-free, on the other hand, if the user seats next to the driver this restriction is not applied.)
5. Are there any specific directions: for the person who calls? (For instance every time Iraklis calls pass me the line), for the time? (example: today between 08.pm and 10.pm deny all the calls, because I want to have dinner with my wife), for the place? (example: when I am at the library for studying deny all the calls), for the stakeholder's actions? (example: when I run deny all calls), or other general directions? (example: when I receive calls from private number, deny the call)

The component which gathers all the information which the assistant either perceive from the user and the environment or from its knowledge base and common sense is a broadcast receiver. As we have already mentioned broadcast in android is the system-wide events that can occur when the device starts, when a message is received on the device or when incoming calls are received, or when a device goes to an airplane mode, etc. Broadcast Receivers are used to respond to these system-wide events. Broadcast Receivers allow us to register for the system and application events, and when that event happens then the register receivers get notified.

The Call Assistant uses a dynamic broadcast receiver which works only if the application is active or minimized. It can receive messages regarding the incoming calls and also information about the place, possible actions which the user can conduct as well as the physical activity when the phone rings.

The registration of the broadcast receiver is made by the following commands

BroadcastReceiver myBroadcastReceiver = new MyBroadcastReceiver();

```
private void registerBroadcast(){
    IntentFilter intentFilter = new
IntentFilter("android.intent.action.PHONE_STATE");
    intentFilter.addAction("my.action");

    registerReceiver(broadcastReceiver, intentFilter);

}
```

The receiver creates the context of the call and applies the decision policy which the user "learns" to the assistant. Finally it applies the rule which is the most suitable regarding based on the created context. Schematically we can see the whole procedure in figure 28.

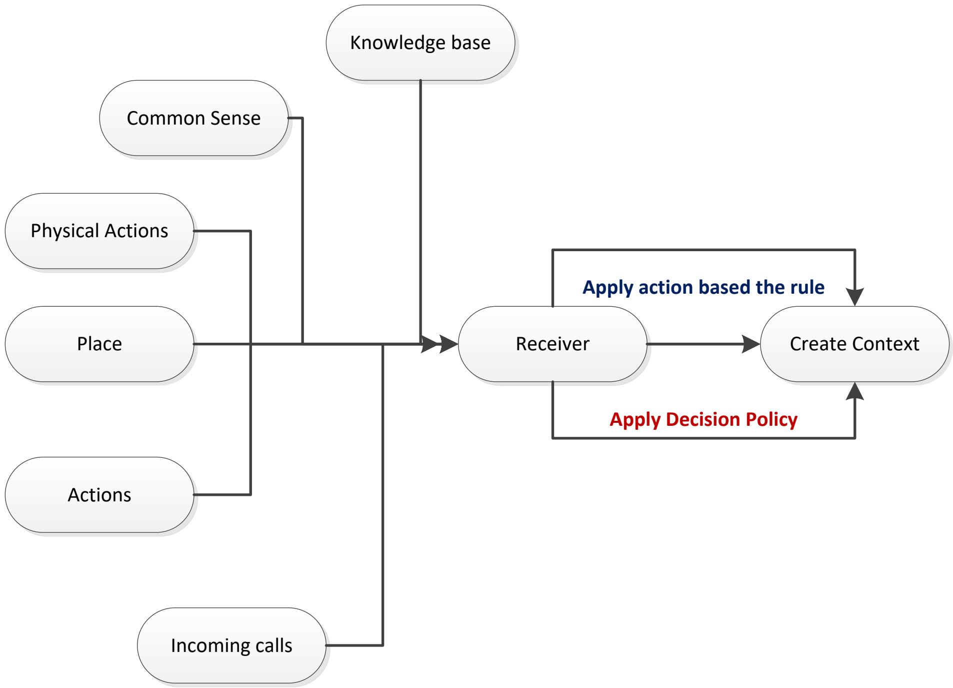


Figure 28

The receiver monitors the state of the phone regarding calls. So when a phone call is occurred the broadcast receiver knows when the phone rings, when the phone called is hooked or when it is answered.

When the receiver senses the phone call and **before** the first ring, it creates the context of the phone call, meaning that it gathers all the data which are essential for the assistant to decide for the action.

The context is consisted from the following facts (literals)(Michael:2019:83) **and their negations**:

date : represents the date of the call, for instance 10 April

day : represents the day of the call, for instance Saturday

periodOfweek :represents the period of the week, the possible values are weekend and workweek

timeOfCall : represents the time of call, for instance 12:31:05

periodOfday : represents the period of the day, the possible values are early morning, Morning, Afternoon, Evening and night

place : represents the place which the user was when the phone rung, for instance office

action : represents the action which the user was doing when the phone rung, for instance meeting

schedule : represents the action's schedule, for instance Monday to Friday

startTime : represents the time which the action starts, for instance 07:00

endTime: represents the time which the action ends, for instance 07:30

contactName: represents the caller's name as it is recorded in the mobile's contacts, for instance Iraklis Papadopoulos

phoneNumber : represents the caller's phone number as it is recorded in the mobile's contacts, for instance 9999999999

callerStatus : represent the relation between the user and the caller as it has been labeled by the user, for instance family

physicalActivity : represents the physical activity of the user when the phone rung, for instance walking

numberOfCallsFromThisNumber : represents the number of calls which the user has received from the caller that day, for instance 3

timeDifferenceOfTwoPreviousCalls : represents the time difference (in minutes) between the last two calls from the same caller that day, for instance 20 minutes

timeDifference: represents the time difference(in minutes) between this call and the last one, for instance 3 minutes

An example of the created context is the following:

```
date(10
Απριλίου)day(Σάββατο)periodOfweek(Weekend)timeOfCall(12:31:05)periodOfday(Afternoon)
place(swimming pool)place(Unknown)action(Unknown action)action(Unknown
action)schedule(Unknown)schedule(Unknown
schedule)startTime(Unknown)startTime(Unknown)endTime(Unknown)endTime(Unknown)contact
Name(Βατικιώτης
Κώστας)phoneNumber(+306974057197)callerStatus(Unknown)callerStatus()physicalActivity
(Uknown
activity)numberOfCallsFromThisNumber(0)timeDifferenceOfTwoPreviousCalls(0)timeDiffer
enceOfTwoPreviousCalls(0)timeDifference(0)timeDifference(0)callReview(Unknown)callRe
view()-date(10 Απριλίου)-day(Σάββατο)-day(Σάββατο)-periodOfweek(Weekend)-
timeOfCall(12:31:05)-periodOfday(Afternoon)-place(swimming pool)-place(Unknown)-
action(Unknown action)-action(Unknown action)-schedule(Unknown)-schedule(Unknown
schedule)-startTime(Unknown)-startTime(Unknown)-endTime(Unknown)-endTime(Unknown)-
contactName(Βατικιώτης Κώστας)-phoneNumber(+306974057197)-callerStatus(Unknown)-
callerStatus()-physicalActivity(Uknown activity)-numberOfCallsFromThisNumber(0)-
timeDifferenceOfTwoPreviousCalls(0)-timeDifferenceOfTwoPreviousCalls(0)-
timeDifference(0)-timeDifference(0)-callReview(Unknown)-callReview()
```

The context answers the following questions:

- What day and time the phone call occurred? => gives date, day and time, if is weekend or workday, if it is holiday, the period of day (early morning, morning, Afternoon, evening, night)
- Where is the user during the phone call? => gives place
- What is he/she doing ? => gives the action which the user does that specific time and also gives the physical activity of the user (running, in vehicle, etc)
- Who is calling? => gives phone number, contact name as it is record in the log of the mobile
- What king of relation has the caller with the user? => gives the relation between them.
- How many times the caller called the user during the day? => gives the number of calls for this day so far.
- What is the time difference between the last two calls? =>gives the time difference between the last two calls.
- What is the time difference this call the last call? =>gives the time difference between this call and the last call.

All data create the context which the assistant comprehends and in this context we have the application of the rules which the user provides.

The receiver is designed to execute (after the application of the decision policy) the following actions: **allow calls, to deny calls, to mute calls, to volume up calls.**

5.5 Decision Policy and Machine Coaching

The decision policy is consisted by a set of rules that follows the idea which had been displayed by Michael (2019:83).

A **rule** is a triplet (name, body, head) were:

- name is any finite alphanumeric sequence (including text underscore), and denotes the rule's name , for instance Rule_1 or Rule_2 ;
- body is a conjunction of facts (literals), that is, given the notation defined above, a comma-separated list of plain of negated predicates, for instance place,

action(Meeting);

- `head` is a single literal, for instance assistantActions(Allow).

The rule has the following structure:

```
name :: body implies head;
```

The :: is a delimiter separating the rule's name from its main part, the implies wires name and body with the head, and ; denotes the rule's end. We say that two rules are conflicting in case their heads are conflicting literals, as defined above

Some examples of rules are presented:

Rule 1

When I am at the Office and Iraklis calls me then I want you to allow calls.

Rule_1:: place(Office), caller(Iraklis) implies assistantAction(Allow)

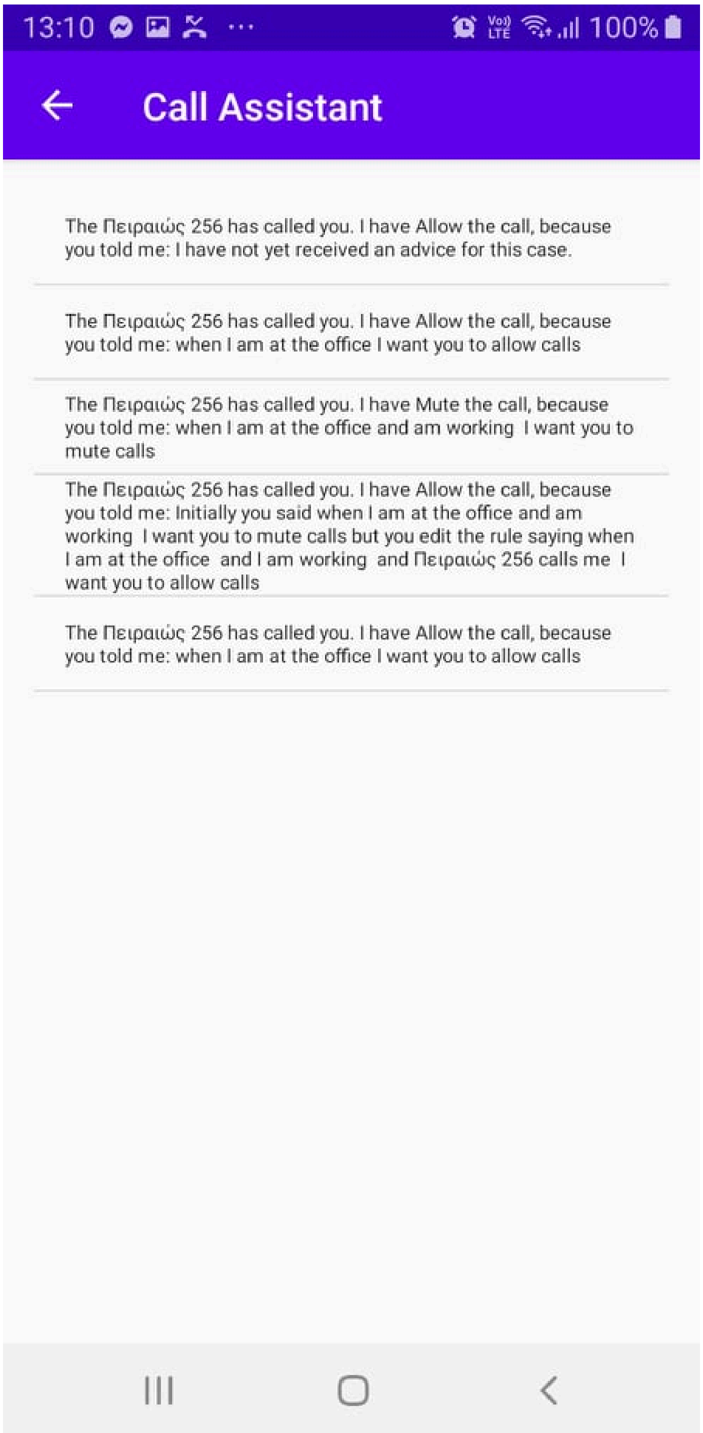
Rule 2

When I am at the Office and I am on a meeting and Iraklis calls me that means I want you to mute calls

Rule_2:: place(Office), action(Meeting), caller(Iraklis) implies assistantAction(Mute)

The assistant "learns" by the feedback which the user provides through a machine coaching procedure. This procedure allows the user to provide the advice and all the facts that are necessary either for creating the context or for the decision policy. The assistant translates the advice to a rule that its structure is complied with the decision's policy rule. In other words, the user coaches the assistant through a learning procedure so it can develop its decision policy. This procedure is explained as follows.

Initially, the assistant has no rules in its knowledge base so it applies a "default" rule. This rule permits the calls in all those cases in which the assistant has no advice from the user. So in this case, it will allow the call. The user can see all the incoming calls and he/she can select the call which he/she wants to apply a new rule.



The user can select the call which he/she likes to provide a rule, in this case there are five calls by clicking on it.

Figure 29

By pressing the microphone, the user expresses the rule by using natural language. The assistant records the advice and it "translates" by creating the corresponding rule automatically. The created rule will have the structure which it has been described above.

The automatic "translation" of the natural language to a structured rule is possible to a mechanism that allow assistant to "learn" the words, in previous steps, that are useful for constructing the rule. More specific, when the user records his name at the begging, the application creates a table named VOCABULARY which will hold all words that will be used for the creation of the rules and they are connected with context of a call. This table is updated every time the user passes a fact.

For instance if the user states that the location: Alexandras 26, Athens 18833, Greece represents the Office then in the VOCABULARY we have the creation of a record indicating: Office=>place(Office). Now the assistant has associated the word office with the fact place(Office) so when it "hears" the word office it will use the appropriate fact.

This is repeated every time the user informs the application for a place, meaning that user also "learns" the assistant new words which the assistant "learns" and "translates" into his language. So if the stakeholder adds three labels like Office, Home and swimming pool for three corresponding places we have three records like:

Office=>place(Office)

Home =>place(Home)

Swimming pool =>place(Swimming pool)

Like the case of Office, during the creation of the argument when the assistant listens the word "swimming pool" it will understand that it has to use place(Swimming pool)

The same idea is applied and in the other cases of actions, relations, contacts names etc. The assistant "learn" new words and connect them with the corresponding facts.

When the user express the advice the assistant parses the words and checks those that they a "meaning" for the assistant (they have a corresponding relation in its vocabulary). The only limitation is that the user should use the word *"and"* in order to have the *comma-separate* between the facts which they consist the rule.

Example

In the vocabulary we have the following records:

Office	place(Office)
Working	action(Working)
Family	relation(Family)
Allow	assistantActions(Allow)
Want	implies

Let assume that the user gives the following advice using the mobile's microphone

"When I am at the Office and I am working and a member of my family calls then I want you to allow calls"

When the assistant receives the advice it parses it word by word. For each word it searches the Vocabulary in order to find a record with a corresponding value. In this case for the first five words “When I am at the” it finds nothing, so it continues with the sixth word which it has found the corresponding value place(Office). The rules has just started to be constructed and initially we have

Rule_1:: place(Office)

The assistant continues and founds another word, this case the “and” so it adds in the rule the comma-separate

Rule_1:: place(Office),

From the next four words “I am working and..” the assistant “knows ” two of them so now the rule is constructed as follows

Rule_1:: place(Office), action(Working),

In the rest part of the advice the assistant clarifies the words “family”, “want” and “allow” so it updates the rule as follows:

Rule_1:: place(Office), action(Working), relation(Family) implies assistantActions(Allow);

When the rule is constructed the user can add the advice in natural language and the corresponding constructed rule.

In the previous lines we explained the rules’ form and how the assistant “understands” and “translates” the advice expressed in natural language to the proper structure of rule. In the rest of the chapter, we display how the assistant selects the desirable rule.

After the creation of the context the assistant checks which of the rules hold in this context. A rule is hold when the rule’s head is hold meaning *that all the facts* which describe the rule’s head are compliant with the facts of the created context.

The figure 30 displays the context which is created by the method which it has been described previously and it is consisted by a number of facts (including their negative values). The rule holds when all the facts of the rule’s body they are also described in the created context. By using modus ponens we can assume when the rule’s body holds then the rule’s head may also holds. In the previous chapter we see that in humans we have the suppression effect (Byrne, R. 1989) but in this case the application of the user coaching remedy this issue since it can provide more accurate rule.

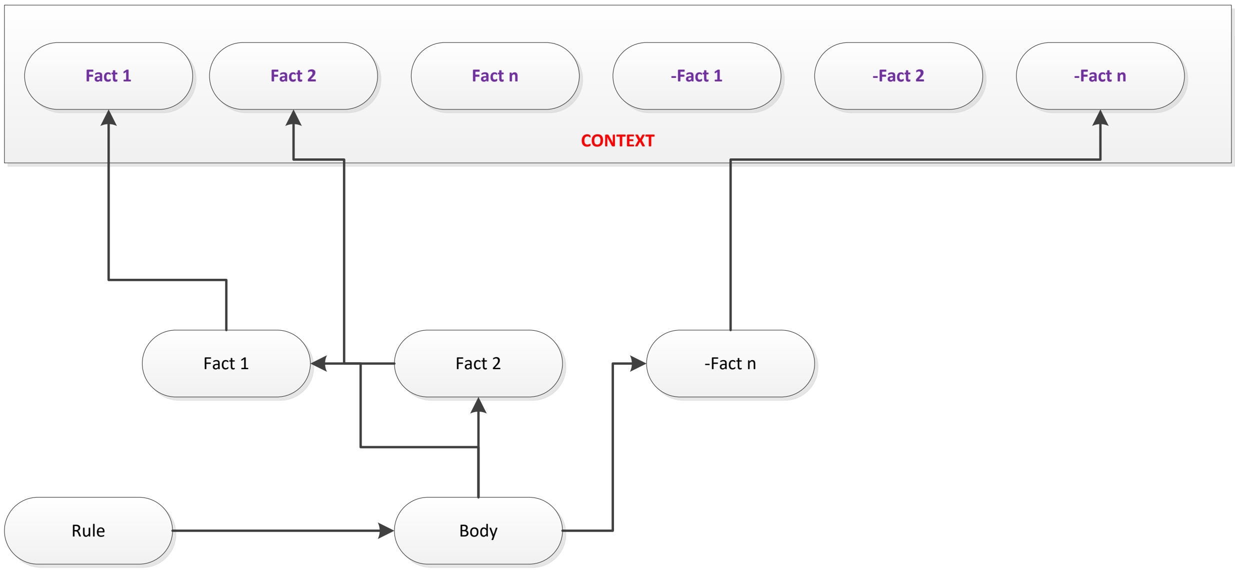


Figure 30

After checking the rules that are hold, the assistant creates the argumentation framework which displays the hierarchy of the rules that acceptable within the created context. It is obvious that we might have attacks between the rules supporting different positions yet the conflict ends by selecting the rule with the highest priority. The priority is established by the position which each rule has in the knowledge base. In other words, the newest rule has bigger priority than an older one because we can assumed that would be preferred over any other conflicting rule already included in the assistant's knowledge base (figure 31).

The user may provide feedback in four (4) ways which they have been declared in the Machine Coaching interaction protocol that is presented in (Michael, 2019: 85). Namely:

1. *Superfluous* rules, these rules are deleted as described in (Michael, 2019: 85).
2. *Incomplete* rules, these rules are updated (Michael, 2019: 85).
3. *Indefensible* rules, the user must add the correct rules (Michael, 2019: 85).
4. *Correct rules, these rules* do not need adjustments

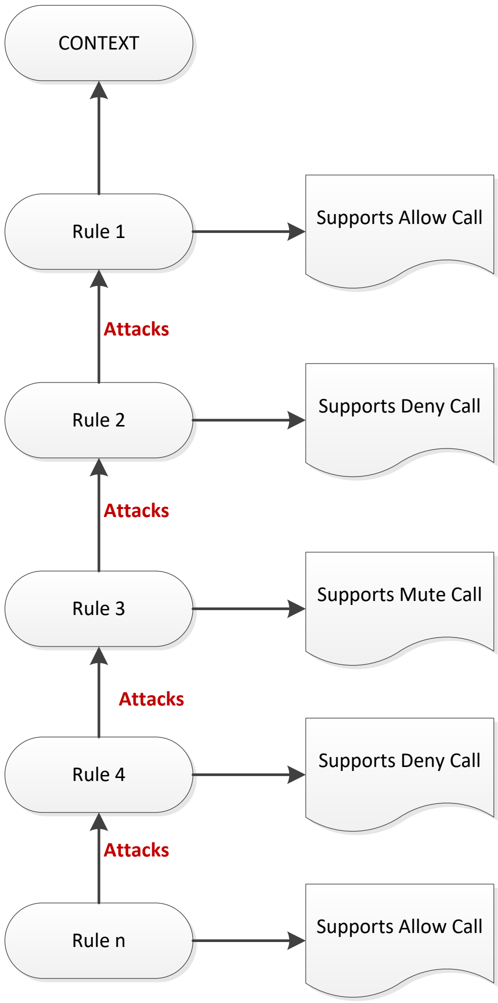

CONTEXT
Rule 1
Supports Allow Call
Attacks
Rule 2
Supports Deny Call
Attacks
Rule 3
Supports Mute Call
Attacks
Rule 4
Supports Deny Call
Attacks
Rule n
Supports Allow Call

Figure 31

Chapter 6
Evaluation

6.1 Evaluation Settings

The evaluation of the application is based on a list of **Cognitive Walkthrough Tasks** that were specifically designed to evaluate the usability and all of functions and operations of the Call Assistant. Each user had to execute the following steps

Cognitive Walkthrough Tasks

Task1 –Install and Run the Application

Sub-tasks

- Install the Application.
- Open the Application.
- Give the appropriate permissions.
- Open mobile's data or WiFi.
- Open mobile's position.

Scenario Description

The Stakeholder must install the Assistant, make the adjustments on his/her phone and give the essential permissions to the application so it can have access to various tools and modules of the phone.

Task2 – The Stakeholder provides his/her name

Sub-tasks

- The assistant checks if it knows the stakeholder's name. Since it is the first time which the application runs it does not have an answer from its knowledge base. So it asks the user's name.
- The user listen the App's question in a clear manner.
- The user provides his/her name correctly which can see in the corresponding field.
- The user adds his/her name

Scenario Description

The stakeholder must hear the Call Assistant clearly. Furthermore , the user must provide his/her name correctly by using the

1. Button labeled "PRESS THE MIC"
2. Pop Up microphone which the app will display for the user to provide the name
3. The text field labelled " Input message will be display here.." in which he/she can see his/her

successfully.

response

4. Button labeled "ADD YOUR NAME"

Task 3 – the Stakeholder identifies if the Application comprehends his/her current physical activity correctly

<u>Sub-tasks</u>

- The assistant checks the stakeholder's physical activity. It displays its comprehension and how certain it is.
- The user can read this information easily and evaluates the comprehension.

Scenario Description

The application displays the user's current physical activity (running, walking, on foot, on vehicle, still, etc.) and presents its confidence. The assistant's comprehension must apply with user's physical activity. If the assistant is not certain it will display unknown activity.

Task 4 – the Stakeholder adds a label to a location

<u>Sub-tasks</u>

- The user can see the current location.
- The user listens and reads the instructions of the Assistant and provides a label for the current location.
- The user can see his/her feedback as soon as he/she stores the data.

Scenario Description

The application identifies the current location. If it does not have information about the place, asks from the user to give a label (for instance, office, home, swimming pool, my gym etc). The stakeholder provides the label by using the

1. Button labeled "PRESS THE MIC"
2. Pop Up microphone which the app will display for the user to provide the label for the place
3. The text field labelled " Input message will be display here.." in which he/she can see his/her response
4. Button labeled "ADD LABEL TO THE PLACE"

Task 5 – The Stakeholder adds an action(s) to a place

Sub-tasks

- The user can add actions to a certain place by providing, the name of the action, the time period and the schedule of the action.
- The user can cancel an action.

Scenario Description

The user can add actions that are performed to a certain place. The stakeholder provides the action(s) by using

1. The Button "Set Action"
2. The Pop Up microphone which the app will display for the user to provide the action
3. The text field labelled " Input message will be display here.." in which he/she can see his/her response
4. The Button "Add Action"
5. The Open dialog framework that will be displayed in order to set the timetable of the action

Task 6 –The Stakeholder can read the toast messages when the phone rings

Sub-tasks

- When the phone ring the stakeholder can see the corresponding messages from the application. There is a message when the phone rings, message when the user answers th phone and a message when the user hung up the phone.

Scenario Description

The application uses a broadcast receiver that works in the background and senses the incoming calls. When the phone rings, it provides the name of the caller as well as the incoming phone number by using a message. The same does when the user does not answer the phone and when the user ends a call.

Task 7 –The Stakeholder can read the Call-Log and he/she can select the call which needs advice

Sub-tasks

- The user can open the call log of the incoming calls.

Scenario Description

The user must see the log of the incoming calls by using the button labeled "View All

- The user can see all the incoming calls with the corresponding advice. The log displays the incoming calls of the day.
- The user can select from the view the phone call which he/she wants to apply a rule

Calls". The user should see all the incoming calls and he/she can select the record which needs a rule by clicking on it.

Task 8 –The Stakeholder adds a rule and an advice

Sub-tasks

- The user selects and opens the advice section.
- The user can add the relation status between the caller and the stakeholder, if that is needed.
- The user can select from the choices of his /her response to add rule.
- The user constructs the rule.
- The user adds the rule.

Scenario Description

The user selects the phone call in which he /she want to apply a new rule. The user opens the advice mode. The application provides a number of details about the call. In case the assistant does not know the relation between the user and the caller it asks for details which the user has to provide by using the buttons "Press the Mic to pass the advice" and "Add relation". The user selects from the spinner the choice "Indefensible (add the correct rule)" and constructs the rule by using the Press the Mic to pass the advice". The app creates the rule and displays the advice which the user can add by using the button "Commit Change"

Task 9 –The Stakeholder tests the rule and the advice

Sub-tasks

- The user tests the efficiency of the app by receiving a phone call that would trigger the new rule.
- The user rates if the assistant treated the phone call as needed.
- The user view the log calls and see if

Scenario Description

The user receives a phone call and tests the efficiency of the rule. Does the assistant perform as expected? The user should see the management of the call. Then it should see from the log the recorded phone call, action and advice

the call with the action and the advice is recorded	The behavior of the application should be as expected: 1. Sense the phone call. 2. Select the rule. 3. Apply the rule. 4. Log the phone call and provide the user's advice

Task 10 –The Stakeholder edits a rule and an advice

Sub-tasks	**Scenario Description**
• The user selects and opens the advice section. • The user can select from the choices of his /her response to edit rule. • The user constructs the rule. • The user adds the updated rule.	The user selects the phone call in which he /she want to apply a new version of the rule. The user opens the advice mode. The application provides a number of details about the call. The user selects from the spinner the choice "Incomplete (edit the current rule)" and constructs the rule by using the Press the Mic to pass the advice". The app creates the rule and displays the advice which the user can add by using the button "Commit Change"

Task 11 –The Stakeholder tests the updated rule and the advice

Sub-tasks	**Scenario Description**
• The user tests the efficiency of the app by receiving a phone call that would trigger the updated rule. • The user rates if the assistant treated the phone call as needed. • The user view the log calls and see if the call with the action and the advice is recorded	The user receives a phone call and tests the efficiency of the updated rule. Does the assistant perform as expected? The user should see the management of the call. Then it should see from the log the recorded phone call, action and advice

Task 12 –The Stakeholder deletes a rule and an advice

<u>Sub-tasks</u>

- The user selects and opens the advice section.
- The user can select from the choices of his /her response to delete a rule.
- The user deletes the rule.
-

Scenario Description

The user selects the phone call in which he /she want to delete the rule. The user opens the advice mode. The application provides a number of details about the call. The user selects from the spinner the choice "Superfluous (delete the rule)" and deletes the rule by using the button "Commit Change".

Task 13 –The Stakeholder checks the Application's efficiency in the background

<u>Sub-tasks</u>

- The user opens the application and leave it open, working in the background (app is using data but the user is not actively use the app)
- The user receives a phone call and sees if the Assistant's broadcast receiver is triggered.
- The user evaluates the management which the Assistant performed for the call.
- The user finds the call, the action and the advice in the call log.

Scenario Description

The user runs the application in the background. The Assistant should perform in the same way as it works actively. The behavior of the application should be as expected:

5. Sense the phone call.
6. Select the rule.
7. Apply the rule.
8. Log the phone call and provide the user's advice

Task 14 –The Stakeholder closes the Application

<u>Sub-tasks</u>

- The user closes the application

Scenario Description

The user close the application by using the button "Exit Application"

Wharton *et al.* (1994) offer four questions to be used by an assessor during a cognitive walkthrough:

- Will the user try and achieve the right outcome?
- Will the user notice that the correct action is available to them?
- Will the user associate the correct action with the outcome they expect to achieve?
- If the correct action is performed; will the user see that progress is being made towards their intended outcome?

Each participant involved in this cognitive walkthrough had record the step in the process where he/she found an issue and what that issue was. After the completion of the tasks, the participants completed first, a **Demographics Questionnaire** in order to record their gender, age, degree, occupation, their experience of using mobile phones, previous experience in similar applications and second a **Post-task Questionnaire** for capturing the participants' opinion for using the IDE for each specific task. The questionnaire included questions that covered the various parts of the system invoked for each task. It included true/false questions, multiple choice questions, and questions in the five-point Likert scale. The questionnaire also included a section with questions from the **System Usability Scale (SUS) standardized questionnaire.** The questionnaires are shown in Appendix B.

6.2 Evaluation Results

Before the actual evaluation phase, we performed, with each of the participants, a simulation of evaluation identical to the actual one. Some of the evaluations are conducted by installing the application in a mobile phone while for others we had connected the mobile phone with the android studio in which the application was built and run.

After simulation the participants had received the Cognitive Walkthrough Tasks which had to execute according to the advice they had received in the simulation phase while we avoid to provide any kind of additional help. During each task, we recorded all observations, errors and problems which had occurred.

After the task each participant had to answer the Demographics Questionnaire and the Post Task Questionnaire on line. Each participant, before answering the questionnaires, had to read a statement of informed consent regarding the reason for the evaluation, and the data collection and data handling policy. This consent was mandatory for participating in the evaluation. All the participants ended up needing more than an hour to complete the tasks and to respond to the questionnaires.

Eleven people (five male, six female) participated voluntarily to the evaluation. The age varies between twenty to 59 years while all have Greek as a native language

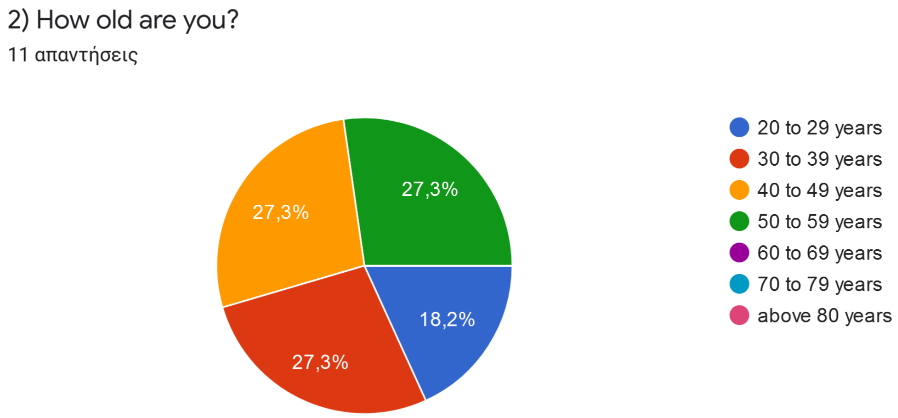


Figure 32

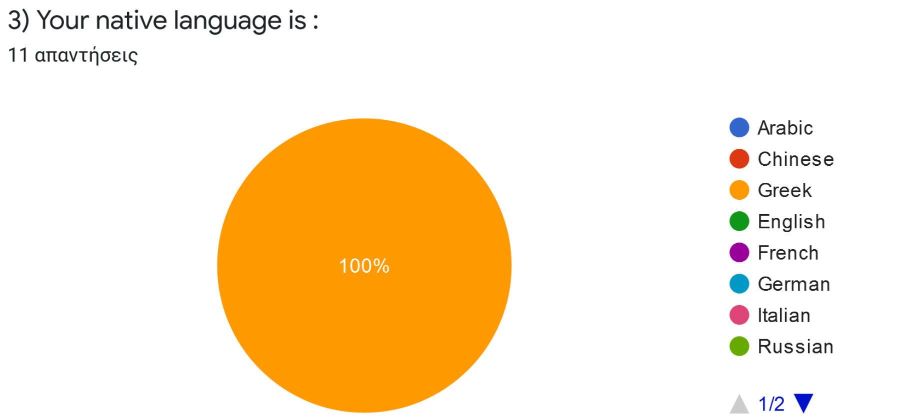


Figure 33

Most of them hold a bachelor degree while a significant rate (36,4%) holds a post graduate degree while the areas of studies vary.

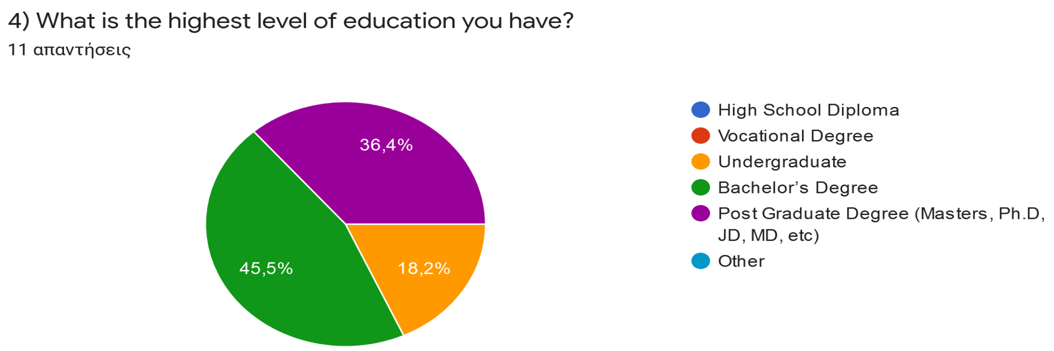


Figure 34

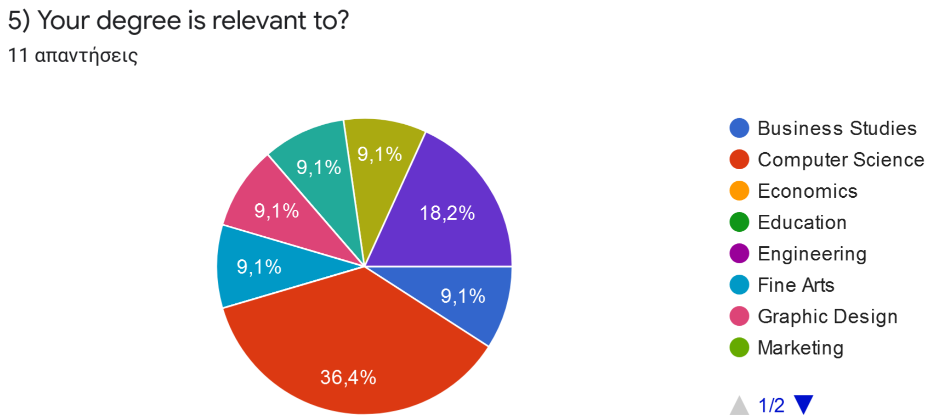


Figure 35

The majority works full time while a small rate studies, yet all of them are using their mobiles at work or at university.

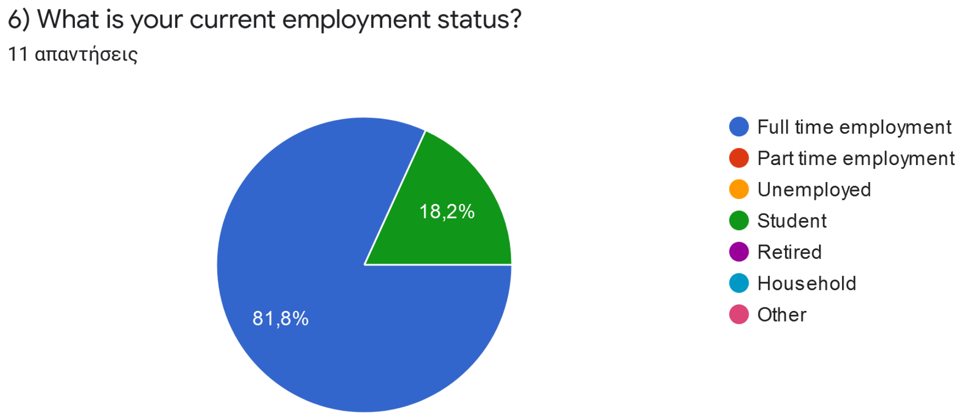


Figure 36

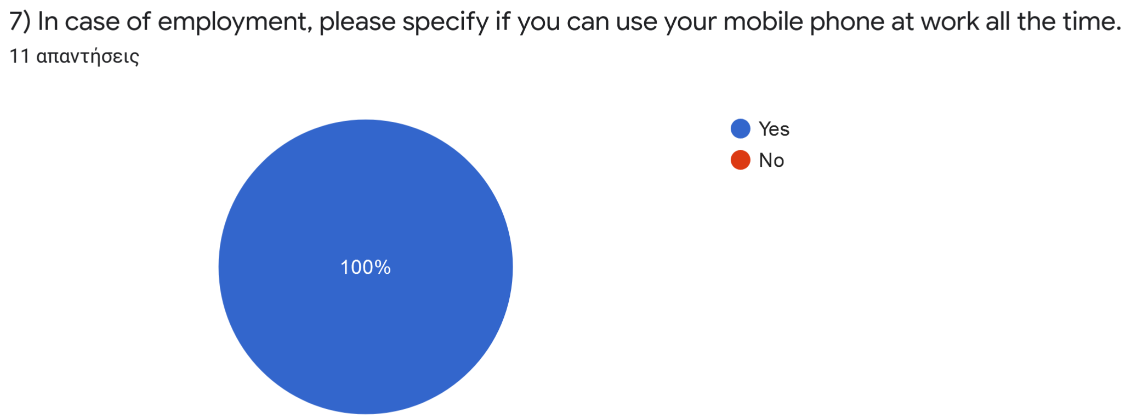


Figure 37

Finally, the majority of them is not married yet the 36% of the participants have children.

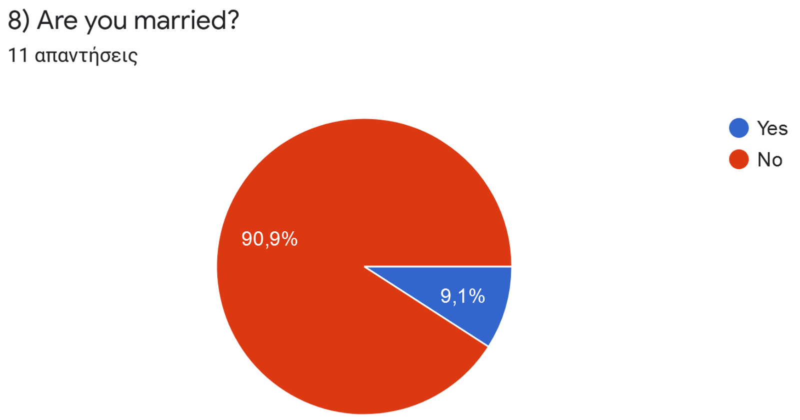


Figure 38

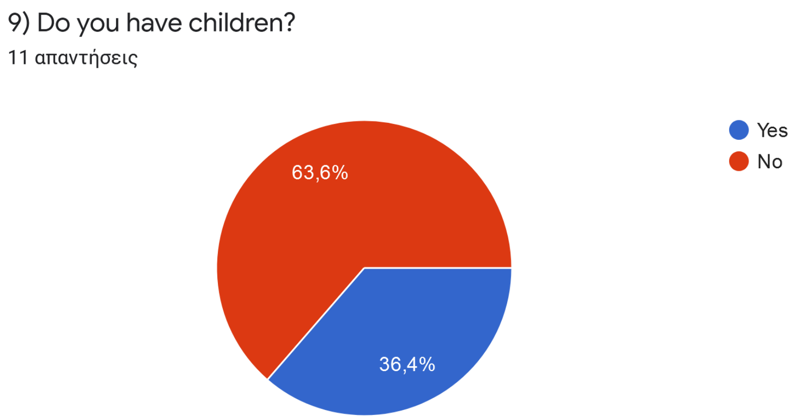


Figure 39

The majority has one mobile phone (figure 40) while it has more than two hundreds contacts registered in their contacts log (figure 41). There is no participant who spends more than three hours using the phone for incoming calls (figure 42) however the spam calls are not exceed the hour, in all cases (figure 43). The majority of the participants have to answer more than ten phone calls but less than twenty while there is a significant rate which has to deal with more than twenty calls (figure 44). All of them are familiar with the technology and all they believe that this kind of application is beneficial (figure 45).

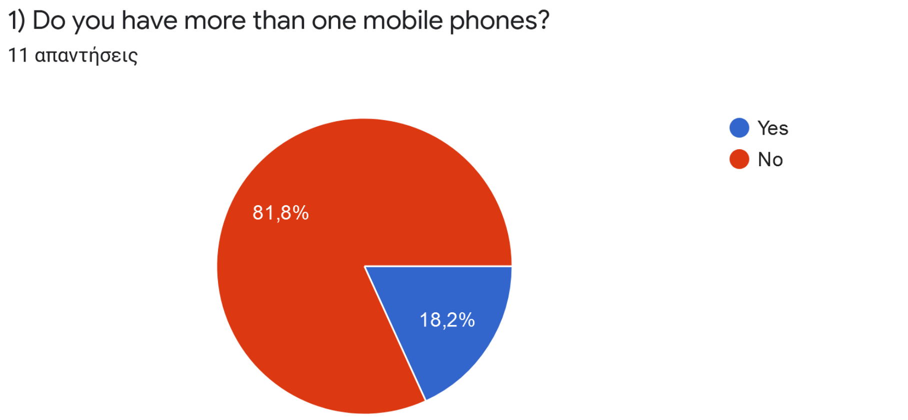


Figure 40

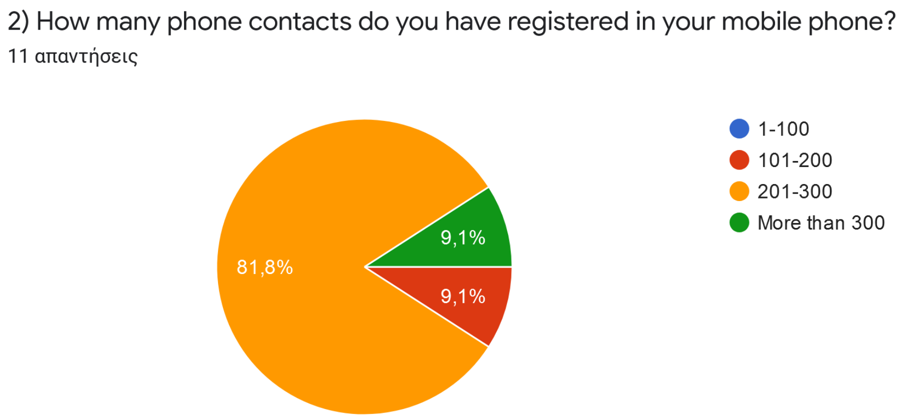


Figure 41

3) How many hours per day do you estimate that you use your mobile phone for incoming calls?
11 απαντήσεις

Figure 42

4) How many hours per day do you estimate that you use your mobile phone for spam calls?
11 απαντήσεις

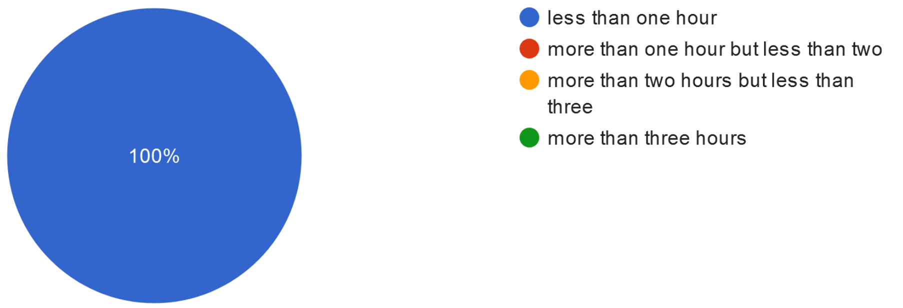


Figure 43

5) How many incoming phone calls per day do you estimate that you receive in your mobile phone?
11 απαντήσεις

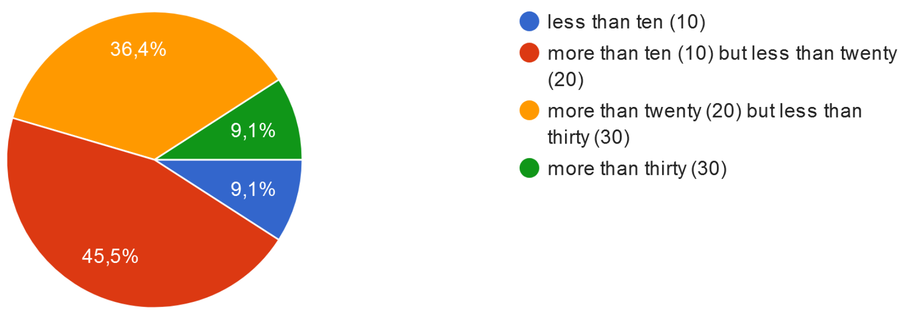


Figure 44

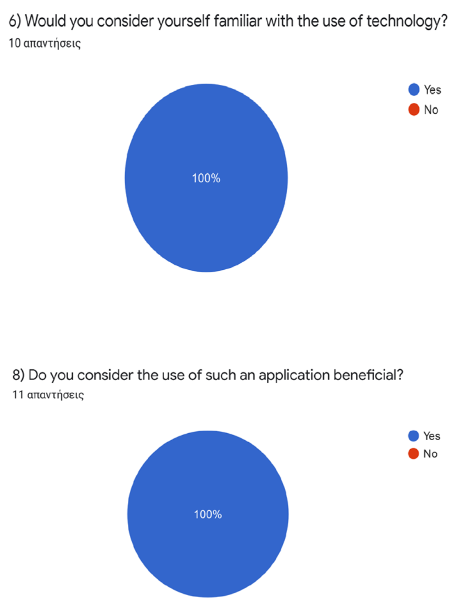


Figure 45

The findings from the **Cognitive Walkthrough Tasks** are the following:

Task1 –Install and Run the Application

All the participants have succeeded to install and run the application. One case found the installation procedure somehow uncomforted since she had to follow a different pattern than the usual. The application could not be installed through google play so each user had to accept an email with the APK file of the application. For all these cases, the users should ignore informative messages from the android in order to install the application; an issue which may have increased a negative point of view for the app.

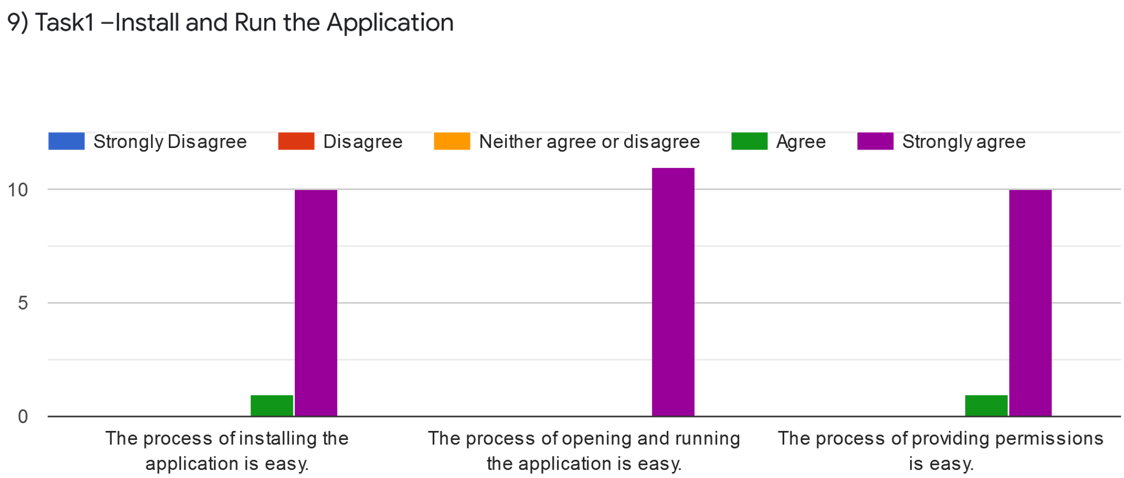


Figure 46

Task2 – The Stakeholder provides his/her name

In one case the application had crushed after a few seconds of running. The user had to re-run the application several times in order to start executing the task without success. Furthermore, there were five cases which the voice of the Assistant had different speed and pitch than those which had been set programmatically. The speed and pitch were changed independently. On the other hand, in cases which had tested the application through the android studio (we connect the device with the android studio) we did not face this issue. This problem has been published in various forums in order the developers to remedy the problem. A possible solution was published after a corresponding publishing of the issue (rate, T., G, R., & Kataykin, P. 2021). According to a developer's answer speed and pitch are defined in Android secure system settings which are the preferences that the user must explicitly modify through the system UI or specialized APIs (system apps or root access) for those values, not modified directly by applications.

One case found difficult to provide her name correctly while the others found the procedure rather easy. In seven cases the Google API for making speech to text worked fine yet, there were three cases which the users had to re-state several times their names. Finally, once the API worked correctly, according to eight users, the assistant stores the name with an easy manner while there were two cases that found the whole procedure rather neutral for a label.

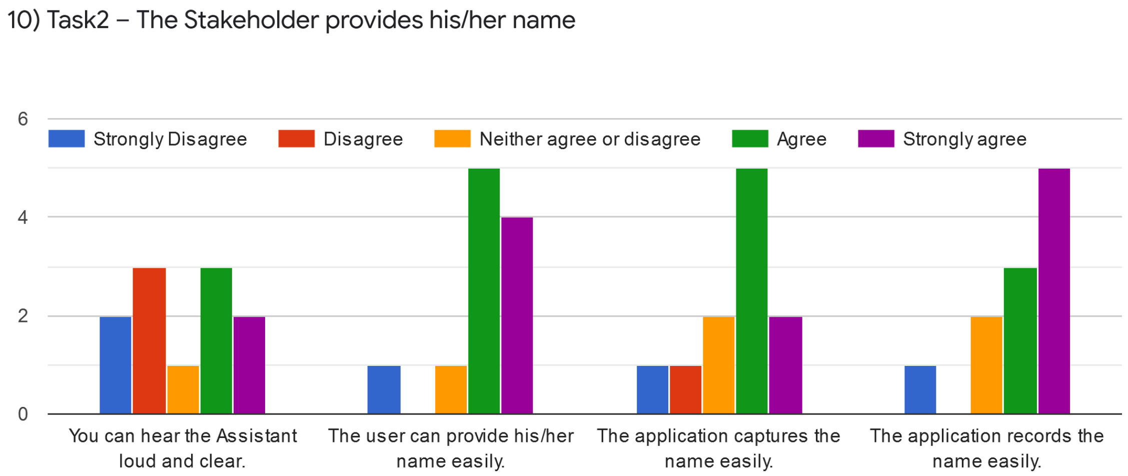


Figure 47

Task 3 – the Stakeholder identifies if the Application comprehends his/her current physical activity correctly

All the participants saw the Assistant's comprehension about this issue. Even the case in which the application did not run, she saw, in the screen, the activity which the assistant had detected with the corresponding confidence. They all state the correctness of the assistant's comprehension.

11) Task 3 – the Stakeholder identifies if the Application comprehends his/her current physical activity correctly

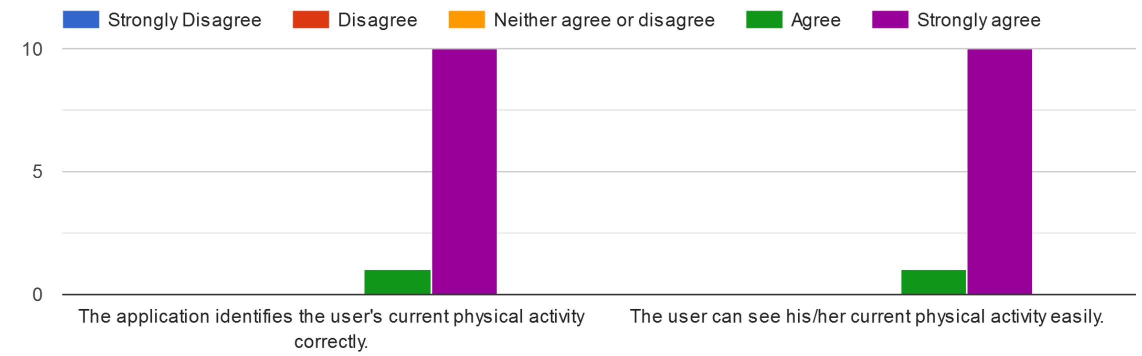


Figure 48

Task 4 – the Stakeholder adds a label to a location

All participants, except the case in which the application did not run, saw their location address in their screen, yet in two cases the assistant had changed several times the results before it concludes to the correct location. This issue is raised by other developers and an idea which may solve the problem is to use Fused Location API from Google Play Services.

One case had found difficult to pass a label to a place, again we face the problem of making speech to text. However all participants, after inserting their labels, saw their feedbacks in their mobiles

12) Task 4 – the Stakeholder adds a label to a location

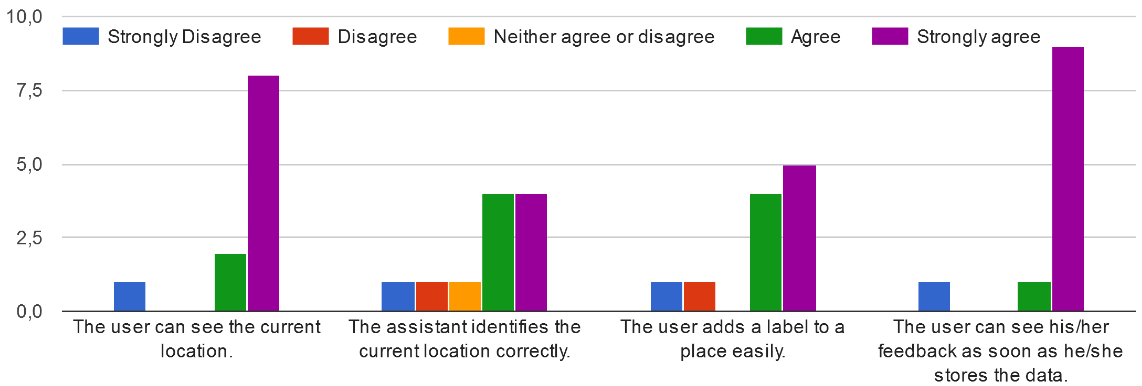


Figure 49

Task 5 – The Stakeholder adds an action(s) to a place

This action had been implemented easily by all the participants (except the case which the application did not run at all). The explanation is that the procedure of adding an action follows a more "conservative" method in which the user fills the data and some of them are selected by a spinner. This method is somehow guided so the possibility of making an error is rather low.

13) Task 5 – The Stakeholder adds an action(s) to a place

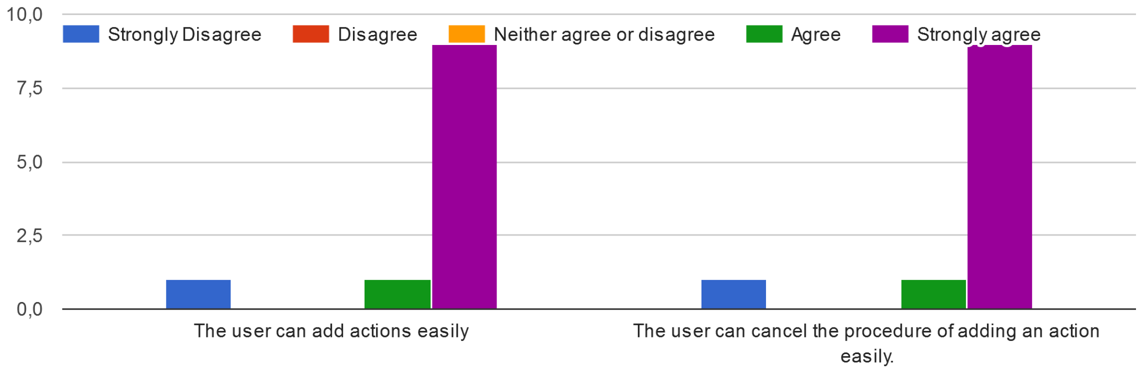


Figure 50

Task 6 –The Stakeholder can read the toast messages when the phone rings

These messages are triggered by the broadcast receiver, all the participants have saw and read the messages. One case found the size of fonts small, a useful note for using it in a newer version of the assistant.

14) Task 6 –The Stakeholder can read the toast messages when the phone rings

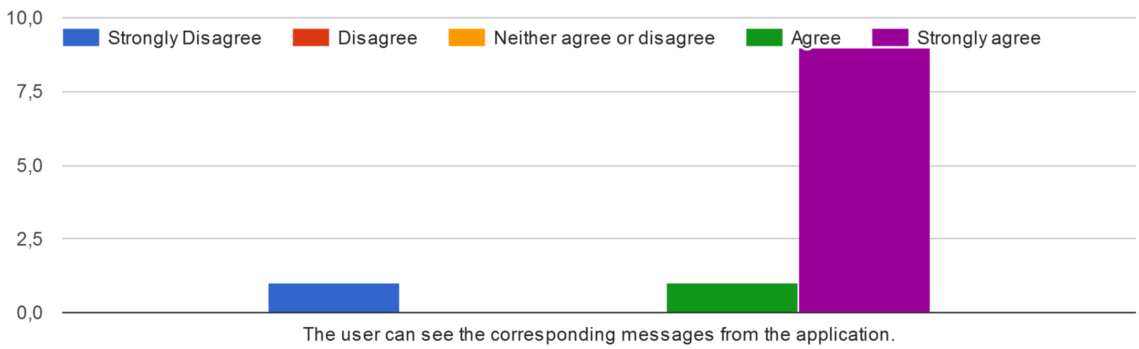


Figure 51

Task 7 –The Stakeholder can read the Call-Log and he/she can select the call which needs advice

All the participants have found the log of calls easily. They have not faced any difficulty to read all the incoming calls and to select the cases in which they wanted to apply a rule. The selection is based on clicking on the call making the participants' action quick and effortless. Again one case found the font size somehow small.

15) Task 7 –The Stakeholder can read the Call-Log and he/she can select the call which needs advice

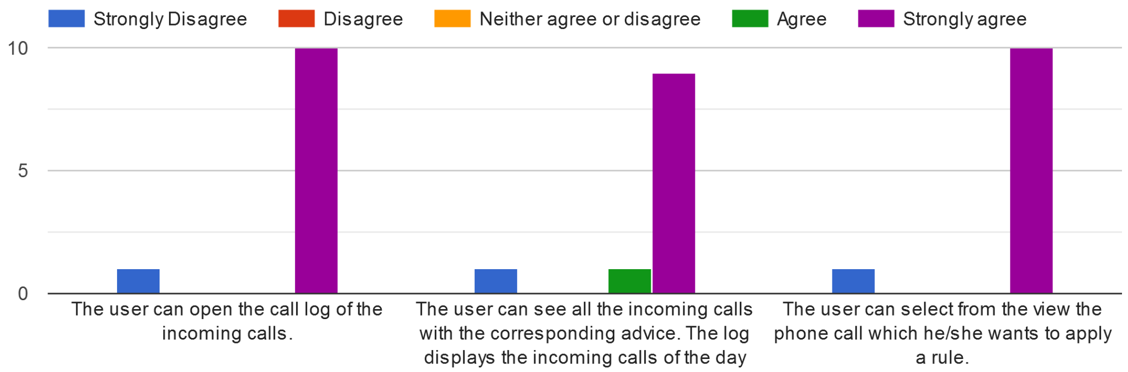


Figure 52

Task 8 –The Stakeholder adds a rule and an advice

All users have been driven to the advice section. In the advice section there were cases in which the assistant asked for the relation status between the caller and the user. One case has found difficult to inform the assistant about a relation while the other have not found significant difficulties.

All users have found the way to add a rule easily yet only four have succeeded to construct a rule without significant problems. The construction of a rule has to follow a pattern which in several cases needs for the user to make several attempts in order to construct the correct rule. The assistant needs to "hear" words that it already knows so if the user states work instead of job (a word that the user had learned the assistant) then the rule will not be constructed correctly.

Three people have found difficult to add the rule in the knowledge base (they were confused on which button they should press) while the rest have not faced an issue.

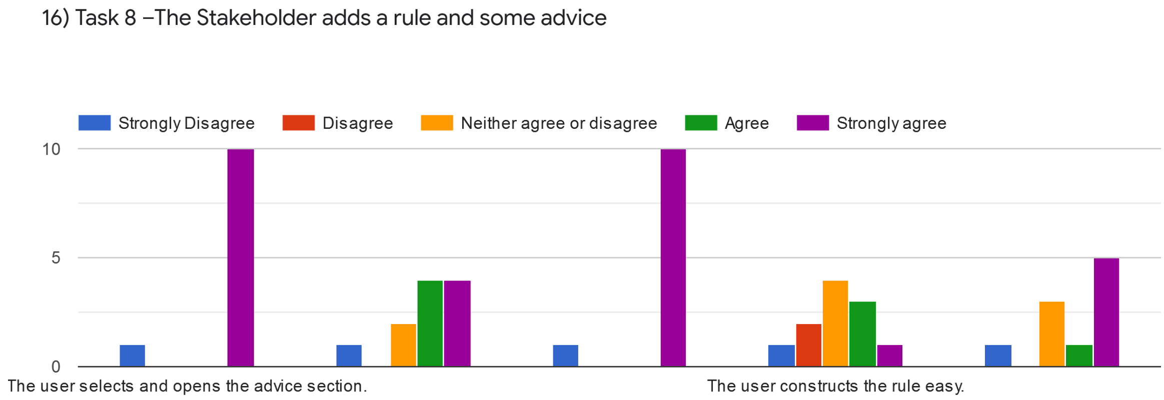


Figure 53

Task 9 –The Stakeholder tests the rule and the advice

All the participants tested the constructed rule successfully. The outcome in each case was the expected. The assistant has performed as it was advised by the users. Following, all the participants read in the log call the assistant's management of the incoming call and the explanation based on each user's advice.

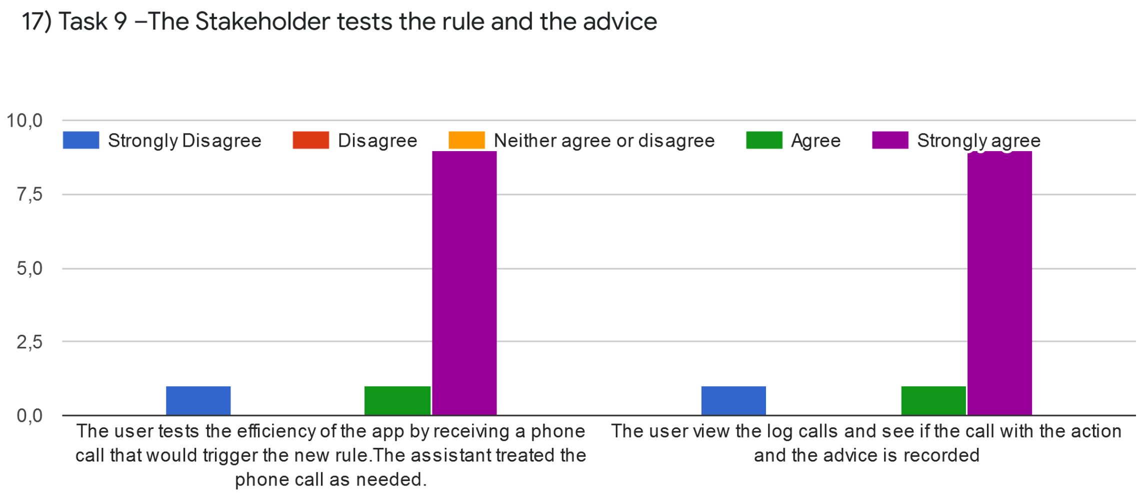


Figure 54

Task 10 –The Stakeholder edits a rule and an advice

In this task due to previous experience all users have easily the call in which they have wanted to edit the advice. Also they have found easily the update option from the spinner component which allows the editing of the rule.

However the actual editing of the rule seems to be a difficult procedure since three participants have faced serious difficulties to change their advice while other five had to deal with some issues. Only two have succeeded to edit the rule somehow effortless. The majority have managed to add the updated advice (after the correct alteration) in the knowledge base.

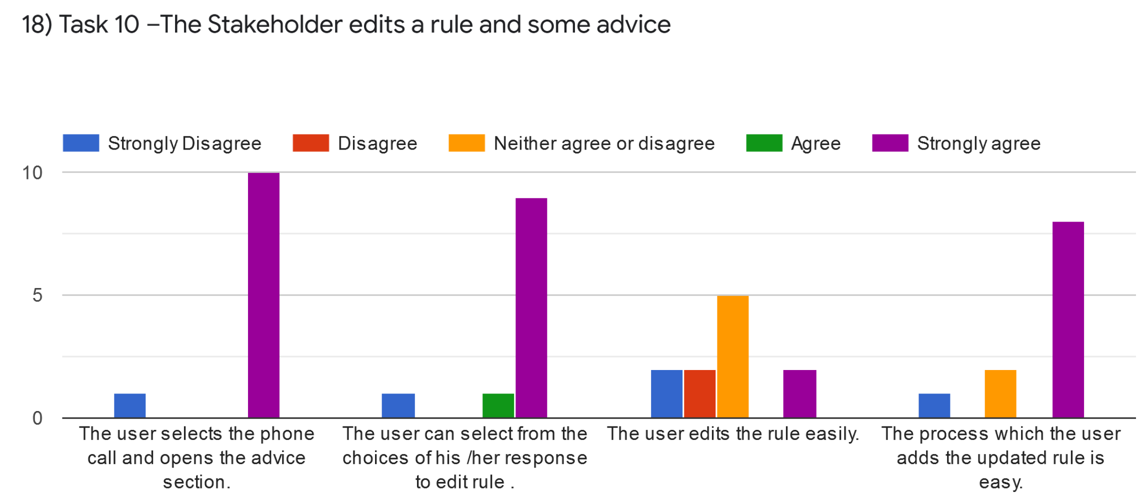


Figure 55

Task 11 –The Stakeholder tests the updated rule and the advice

All the participants have tested the updated rule and advice successfully. The assistant had performed as it had been expected.

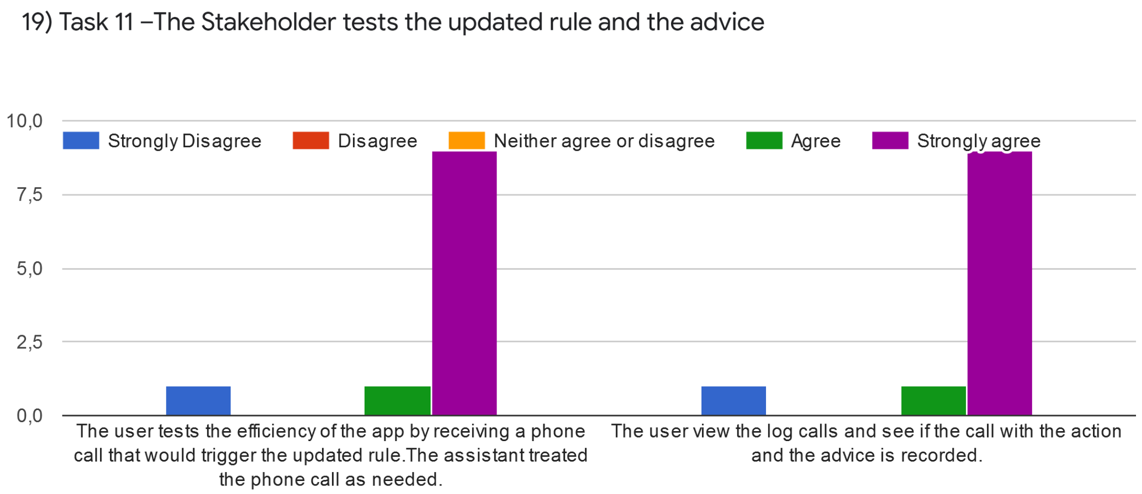


Figure 56

Task 12 –The Stakeholder deletes a rule and an advice

This was the easiest task for the participants. It is a two-click action. They had to select the delete option from the spinner component and then to press the corresponding button.

20) Task 12 –The Stakeholder deletes a rule and an advice

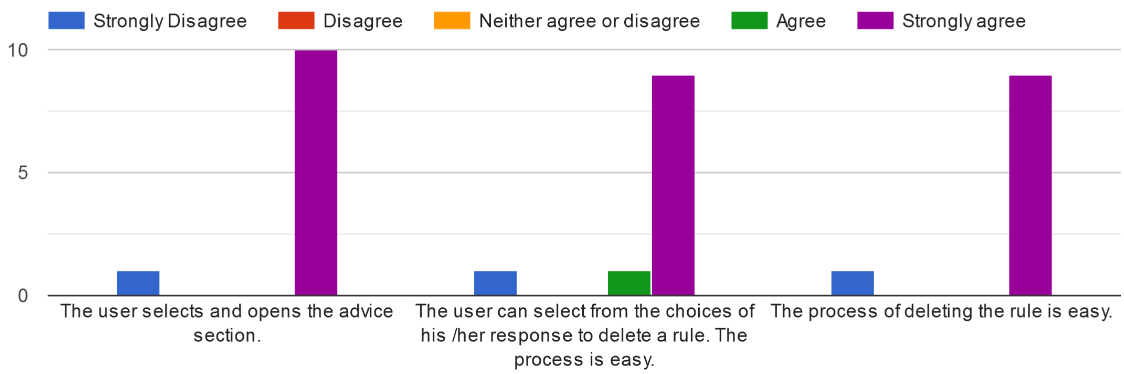


Figure 57

Task 13 –The Stakeholder checks the Application's efficiency in the background

The broadcast receiver is designed to work in the background of the application. The application worked as it was expected. The calls are managed accordingly to the advice which the assistant had in its knowledge base.

21) Task 13 –The Stakeholder checks the Application's efficiency in the background (The stakeholder opens the application and leaves it ope... but the stakeholder is not actively using the app)

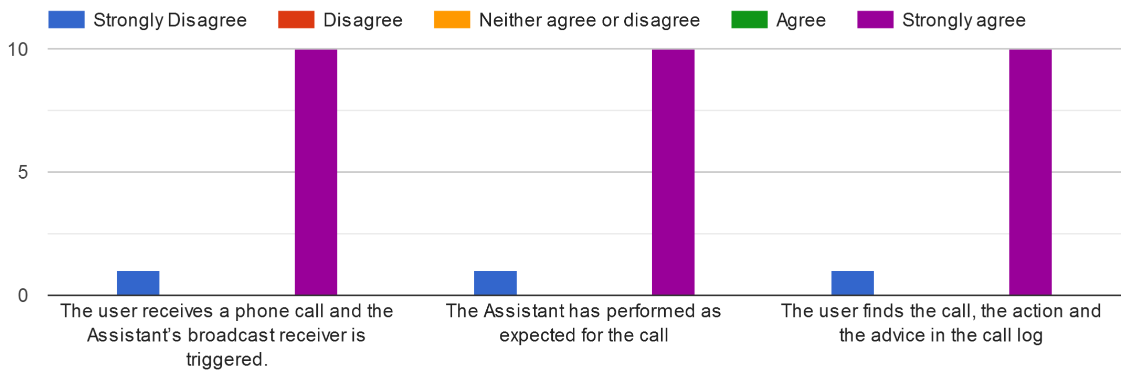


Figure 58

Task 14 –The Stakeholder closes the Application

No issues found on this matter. All the participants closed the assistant with no problems.

22) Task 14 –The Stakeholder closes the Application

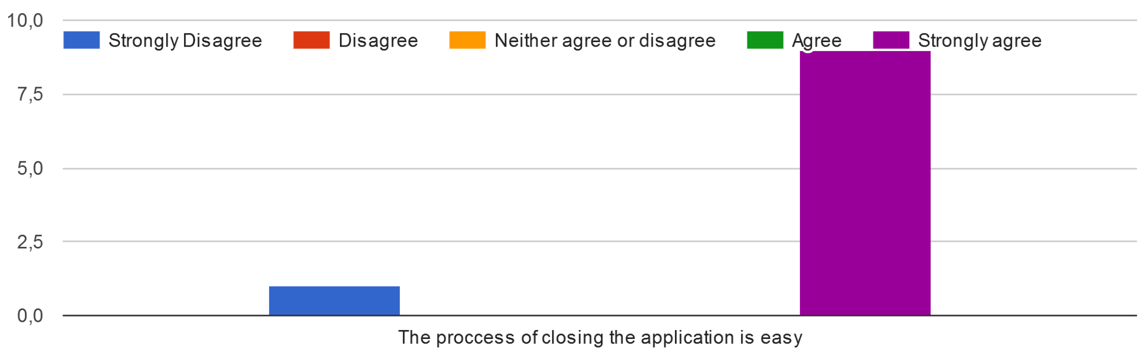


Figure 59

Results from the **SUS standardized questionnaire** showed the following:

Users' answers	Strongly Disagree	Disagree	Neither agree or disagree	Agree	Strongly Agree
I think that I would like to use the Call Assistant frequently.	0	1	9	1	0
I found the Call Assistant complex.	0	1	6	4	0
I found the Call Assistant easy to use	0	3	6	2	0
I will probably need assistance in order to use the application correctly	1	4	3	3	0
I found the Call Assistant incompleted	0	1	6	3	1
I found the Call Assistant	0	1	2	4	4

very awkward to use.					
I would prefer the Call Assistant should display less information.	0	0	7	2	2

SUS Calculation

Participants	**q1**	**q2**	**q3**	**q4**	**q5**	**q6**	**q7**	**SUS Score**
p1	4	3	3	3	2	2	4	**62,5**
p2	3	3	3	2	3	3	3	**60,0**
p3	3	4	3	2	3	4	3	**55,0**
p4	3	3	3	2	4	5	3	**57,5**
p5	3	4	2	4	4	4	5	**55,0**
p6	2	2	4	1	3	4	3	**62,5**
p7	3	3	3	4	5	3	3	**60,0**
p8	3	3	3	2	4	4	4	**62,5**
p9	3	3	4	3	3	5	3	**55,0**
p10	3	4	2	4	3	5	5	**50,0**
p11	3	4	2	3	3	5	3	**47,5**

The maximum score was 62.5 while the minimum was 47.5. The average score is 57.05 out of 100. The standard deviation was 4.86.

6.3 Analysis of the results

After the end of the trial process, we discussed with each user the details of the application's function and behavior and how the users themselves see the possibility of evolving the use of such an application that replaces to some extent the human factor.

After the implementation of the Call Assistant and the discussion we had with the users, we identify two issues which applications like the Call Assistant will face, at least in the near future. The first is the technological limitations and the second is the bias that humans have for such applications like the one we had developed.

A first technological limitation is the range of operating systems which the mobiles are using. The range is large and one of the challenges related to app customization. Many devices are still running on Lollipop, Marshmallow, and Nougat – operating systems from 2014, 2015, and 2016 respectively. In order to create an application like Call Assistant we have to take in mind the above limitation.
A second barrier is the differentiation problem which comes from various original equipment manufacturers (OEMs) that create devices which run the same Android version . Yet these mobiles may behave different. This create changes for the application's functionality and certainly affects how works on all smartphones. For instance, we have noticed that in Sony devices the application had different behavior from Samsung devices. Other problem can the screen size, or the memory capacity.
A third technological limitation is the storage of the facts which the knowledge base contains as well as the processing of the Call Assistant in general. In our case, the knowledge base is created and it is stored in each user's mobile. There is no connection with an external source of storage like a database in the cloud. Furthermore all the operations (sense, comprehension, decision policy actions etc) are performed by the mobile's resources. This task reserves great sources from the mobile's power and memory causing in some cases the application's crushing.
In case the application uses facts from a knowledge base in a cloud and it has to connect with a server in order to make a decision pulling a rule from the decision policy then we have to comply with the General Data Protection Regulation (GDPR). In this case the Assistant should follow the guidelines for the collection and processing of personal information from the users as these guidelines are described in the GDPR legal framework. Furthermore, based on the above scenario, it is more difficult for the users to accept the fact that a lot of their personal data are stored in the cloud. An explanation is that that the users make the assumption that a level of privacy has been lost. Even more, they are afraid the potential risk of third party that could have access to the knowledge base.
The humans' bias for using cognitive assistants, from our point of view, is one of the bigger limitations. The bias is a label that is used in argumentation theory and it is applied in two distinct types of cases which both characterized by the lack to look counterarguments to an existing belief. The first case is when we want to convince others for our position by using counterarguments and ignoring negative arguments to our theory, unless these negative arguments can be anticipated. The second case is when we have the absence of

arguments based on evidence of data; instead the production of reasoning is relying on the bases on our cognitive functions like memory, perception or intuitive inference.

The bias plays a critical role in people's trust for using tools like a cognitive assistant. Trust is an important and essential element that will allow Cognitive Assistants to be adopted by society smoothly. The world is more digitalized and cognitive assistants like Alexa, Cortana gain a lot of popularity and it seems that there is an acceptance of Artificial Intelligence and the technologies in general. However we could establish that the bias and the trust, in general, for using this kind of machines is affected by factors like a previous usage or a lack of experience. But in any case, s lack of trust in agents like a Call Assistant causes less people to using it.

Chapter 7 Conclusion

7.1 Future work

Based on the design pattern that we had followed and the evaluation of the Assistant's operation some critical steps could extend our current work. These extensions can improve almost all modules of the agent making the user interface and the interaction between the user and the agent even more efficient.

As we saw in chapter 4, the assistant sense the environment and the user's activities in order to provide facts for the creation of the phone calls' contexts. Under this frame we could integrate Facebook login feature in the application so the users could grant permissions to our app in order to retrieve information. This could lead to a better and more complete procedure of sensing the environment. This will benefit the creation of the context while it will allow the performing of actions on the Facebook account. Besides Facebook features provide more accurate information in various cases like location and places.

An interesting issue for research is the possibility to connect the Assistant with a smart watch. In other words can we integrate our application with android wear SDK and read data from the watch like steps and heart rate without making apps on android wear or Tizen? To go one step further how the Internet of Things (IoT) and the Internet of Everything (IoE) can be used by our application. IoT and IoE are emerging communication concepts that will interconnect a variety of devices (including smartphones, home appliances, sensors, and other network devices), people, data, and processes and allow them to communicate with each other seamlessly. This will permit our application to get real-time information such as location-based services and tracking.

Another component could be improved is the component of world knowledge and common sense. This could be enriched with more information that would allow the triggering of some beliefs that exist and make a sense. For instance, we can assume that during Christmas phone calls from work do not have priority.

date(25 December) implies period(FamilyTime) period(FamilyTime), relation(Colleague) implies assistantActions(mute)

The application's testing have displayed an issue that needs attention. This is the use of the natural language and how the assistant "understands" the advice and constructs the rules. But the answer to the above problem seems to have various approaches. The research on machine reading comprehension is to endow a computer with reading ability equal to a man, for instance, to make the computer read an article and then answer any questions related to the information embedded in the text. The efforts of the computer scientists, in order to be productive, are focused on the means that they could "applied" the way we comprehend the text in the machines simulating our cognitive system.

A technique for this comprehension by the machines is the Named Entity Recognition (NER) that originates from information extraction (IE). The task of IE is to transform unstructured data into structured information. In case of Natural Language processing the unstructured data are texts and speech so the extraction by a machine of the relevant information and conceptualization into a well-defined format as it mentioned above is not an easy task.

For example for humans the monitoring of news for the new virus covid-19 including various facts like deaths, vaccines or treatments etc. is simple task. This does not apply for the machine since it is rather difficult for them to identify answers to questions like "Who /When/ Where/ What". The solution to this problem came from the Named Entity Recognition (NER) method. The answers to these questions can be easily classified into classes based on their semantics (i.e.persons, organizations, locations, dates, times, etc.) and these classes are important independently of the monitored events.

NER tries to find and classify expressions belonging to these classes (named entities, NEs). In other words NER is the task of identifying the entities (names, events, objects, places etc.) in the text and assign semantic categories to them. The name detection and the assign to the most appropriate semantic category is not simple since the human language is characterized by a polysemy meaning that a name can be referred to multiple entities.

Techniques for NER are most often divided into two main streams: rule based approaches and learning based approaches. In the first case we have the designing and implementation lexical-syntactic extraction patterns and the use of dictionaries that can frequently identify candidate named entities. In the second case, the machine learning is a way to automatically learn to recognize complex patterns or sequence labelling algorithms and make intelligent decisions based on data.

Tools like coreNLP or Spacy are designed to make a use of NER for production use and they will help us to develop the agent in a stage that it could process and "understand" large volumes of text in order to extract information for producing the rules.

A second approach could be the use of an algorithm or a system that would translate the natural language (NL) sentences into first-order logic (FOL) formulas.

Increasing the sensing capabilities of the Assistant combined with a better handling of the natural language will promote the development of the assistant's knowledge base accordingly. This development will lead to a more efficient constructs of rules which is the key element for producing an optimum outcome, thus the assistant performs according to the coaching that it had received.

The assistant could make a decision introducing a kind of fuzzy logic method and using a learning algorithm recognizing patters of the user's behavior. Furthermore, we could make the assistant more flexible in using its own vocabulary in order to give explanations and not just providing back the user's advice.

Final, the design of the existing user interface requires a different approach making the application more friendly to users and efficient. Besides the purpose of the application is to show that the idea of machine coaching works and the agents can provide explanations which have learnt by the users.

7.2 Conclusions

The purpose of the thesis was the construction of an application, the Call Assistant, in order to review the hypotheses that argumentation-in the form of rules- is one of the tools which it can establish a common "language" that machines and humans can utilize when interacting through machine coaching.

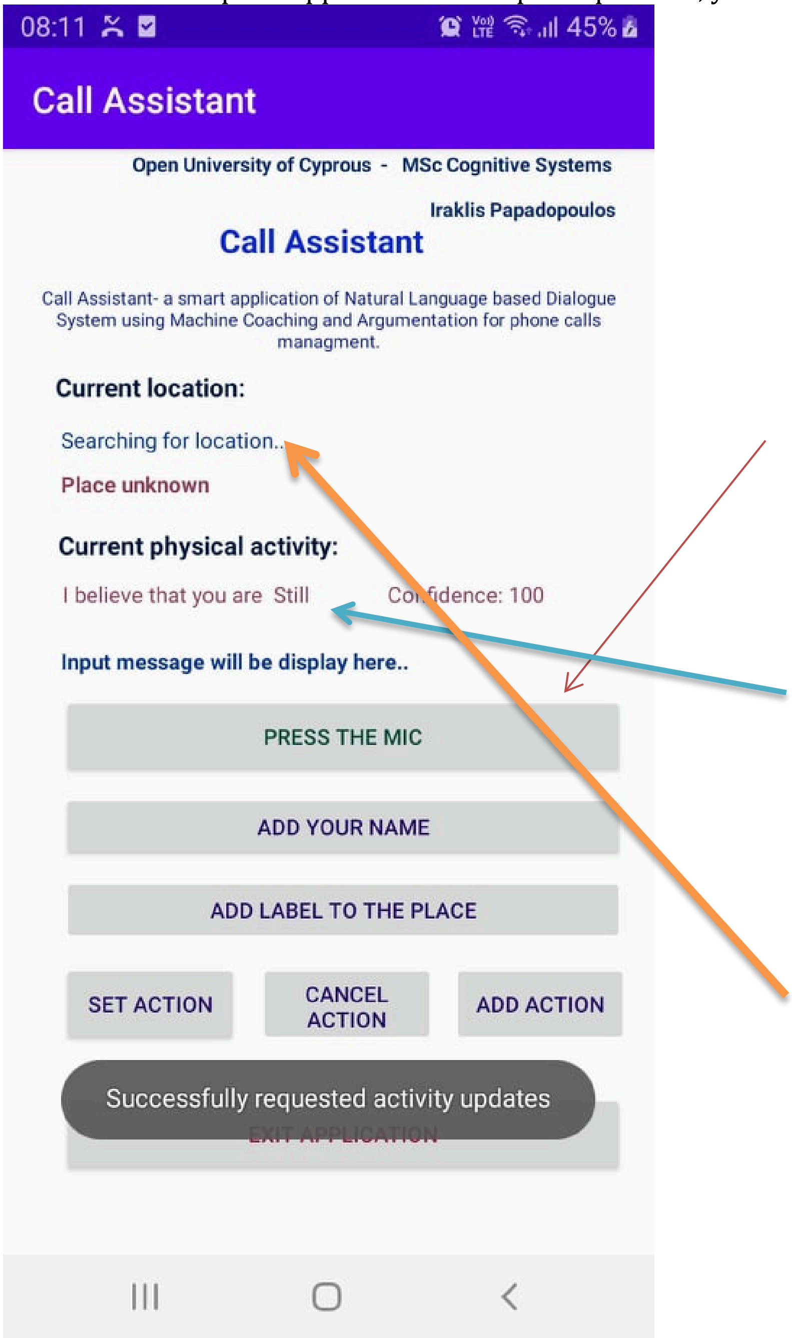
The idea of construction is based on Machine Coaching paradigm (Michael, 2019: 82-85) according to which we could "guide" or even more to "teach" a machine to reason like a human, making our assistant able to "think" and to manage ways of self-improvement.

We tried to manage two major problems, brittleness and transparency. As it has been mentioned the machines are brittle because they do not have the cognitive procedures to form the context of the various situations that occur in reality. Therefore machines break very often when they have to deal with a different condition comparing with the training examples that they have seen.

We tried to develop methods in which the assistant could comprehend the phone calls contexts and to anticipate regardless the necessity of the existence of training examples.
Additionally, we faced the transparency issue, the assistant provides provide explanations about its decisions.
Key element for the above concepts is the common sense. Under this frame it seems that our application "contains", in some level, the set of rules, arguments and actions of what is "efficient" and most logical to be executed according to the conditions and the demands of the environment in which the user has to perform.
We combined reasoning and argumentation in order to use them in a "debate" procedure between the user and the application while we apply in connection with machine coaching the use of natural language which is one of the core difficulties of artificial intelligence and a core difficulty in the current intelligent voice interaction and man-machine dialogues.
We developed an agent with reading and speaking ability, to cooperate with its users, capable to provide explanations and to receive coaching, thus to change behavior and actions in unstable and pre-set environments.

Appendix A
Short Demo

Call Assistant
An application which runs in mobiles for phone calls management. It interacts with the user using natural language while it develops incremented knowledge by using arguments through a procedure of coaching by the user.
The idea is to keep the application as simple as possible, yet effective and user friendly.

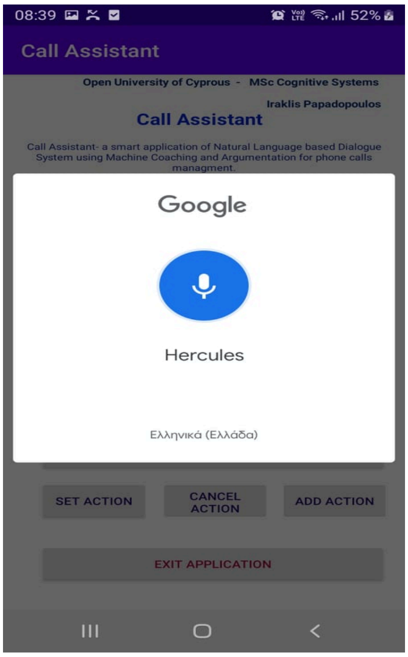


This is the initial screen of the application. Since it is the **first time of launching** the app welcomes the user and asks his name, by using voice, which the user listens from mobile's speaker.

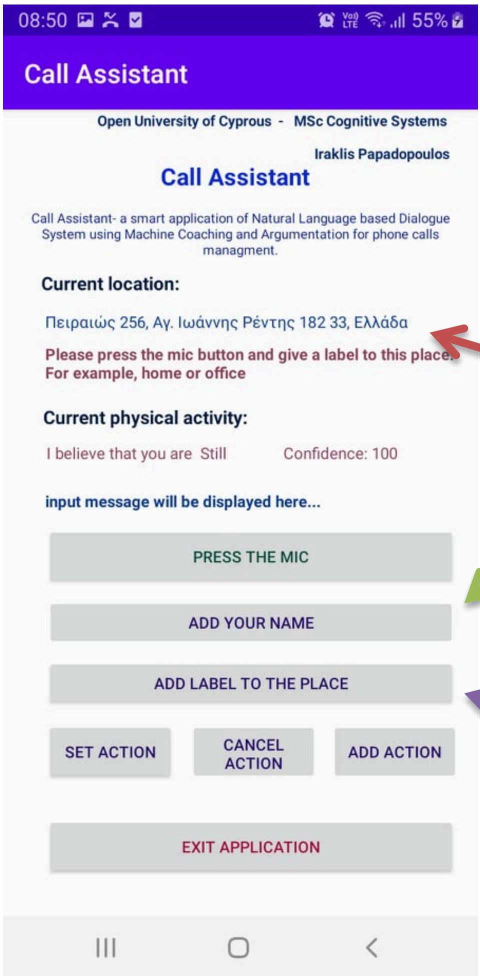
The user uses the button labeled "PRESS THE MIC" to inform the assistant his name

Meanwhile the app seeks and displays the physical activity from the user (Still, walking, running, in vehicle, on foot, on bicycle, tilting or in case it will not comprehend will display unknown activity) and how confident is for this activity.

Finally , the app search for current location



The user after pressing the mic, tell his name which the displays (create text from speech

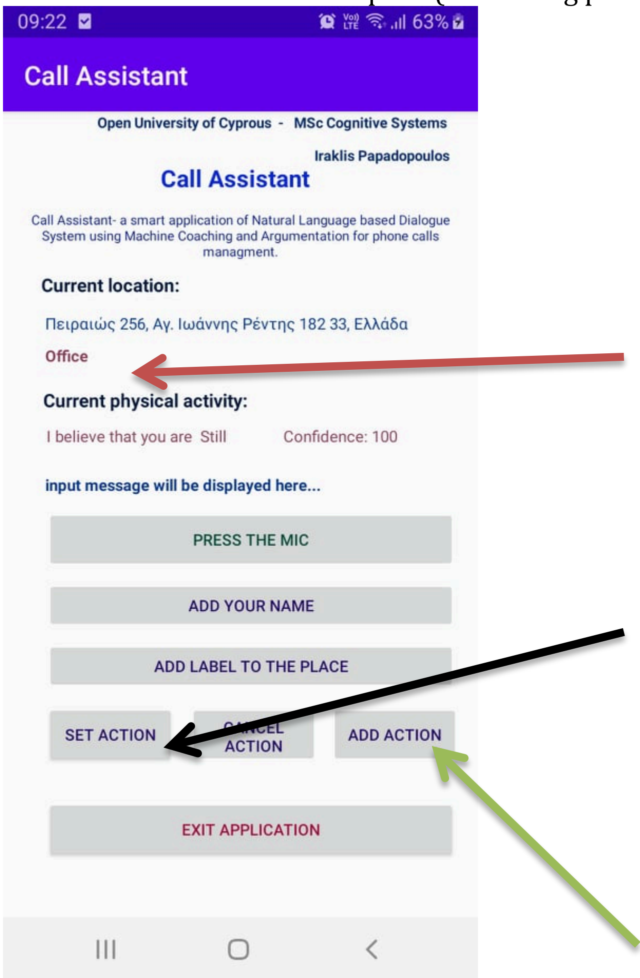


The user records his name by pressing the button labeled “ADD YOUR NAME”

The app has found the location’s address and displays it. Since it has no information about this address, using voice, asks from the user to give a label. Again, the user uses the microphone and sates the label, for example My office, or George’s Office and he presses the button “ADD LABEL TO THE PLACE” to record the data.

When the user records his name, the app creates a table named VOCABULARY which will hold all words that will be used for the creation of the rules and they are connected with context of a call. For instance if the user states that the location: Πειραιώς 256, Αγ. Ιωάννης Ρέντης 18233, Ελλάδα is Office then in the VOCABULARY we have the creation of a record indicating :
Office=>place(Office)
In other word, every time the user informs the application for a place, he also “learns” the

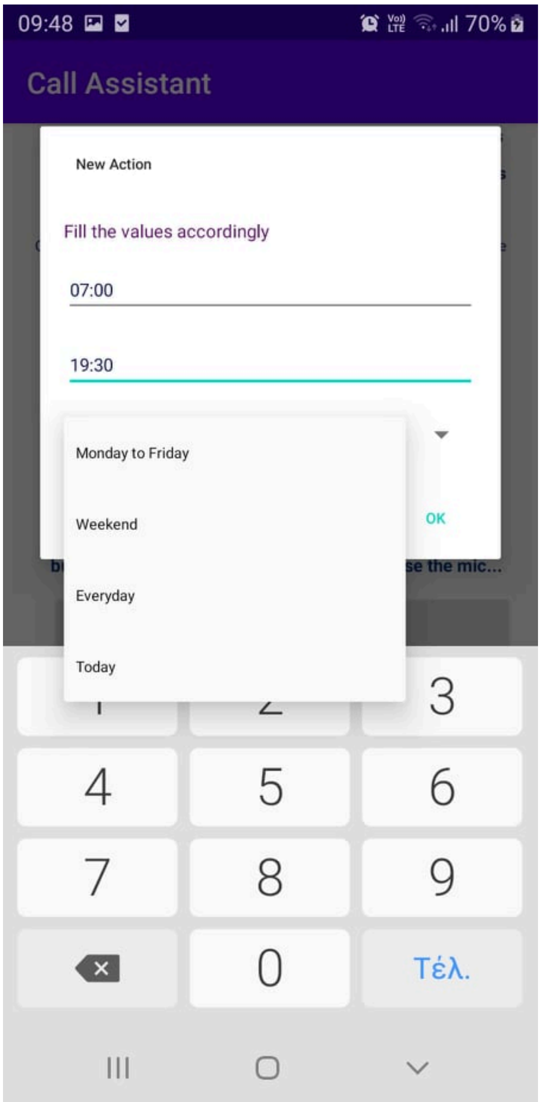
assistant new words which the assistant “learns” and “translates” into his language. In this case, this word refers to a place and the corresponding record is added in the VOCABULARY. So if the stakeholder adds three labels like Office, Home and swimming pool for three corresponding places we have three records like:
Office=>place(Office)
Home =>place(Home)
Swimming pool =>place(Swimming pool)
During the creation of the argument when the assistant listens the word “swimming pool” it will understand that it has to use place(Swimming pool)



Now the assistant is informed about the place, and displays his knowledge right below the address.

The stakeholder can add actions which he/she does in this place by pressing the button SET ACTION

When the holder presses the button the assistants using voice again, asks the label of the action. The user presses the mic and by using voice gives the label of the action, for instance, working, meeting with the manager etc and then presses the button ADD ACTION



After pressing ADD ACTION button the assistant displays a screen in which the stakeholder fills the time which the action starts and the time which the action ends. (The text fields receive only numeric values)
Additionally from a drop down menu selects if the action is from Monday to Friday, if it is performed during weekends or every day or only for today. After filling the values he presses the OK button and the action is recorded in the knowledge base.

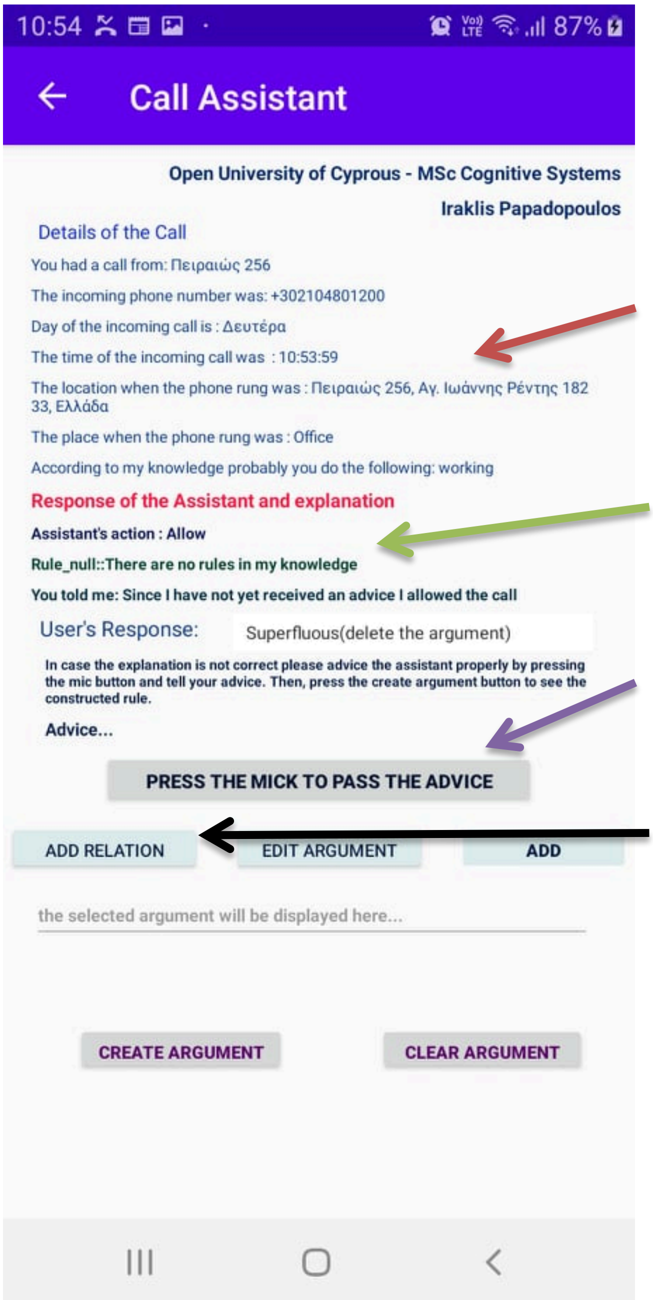
Like places, when the user adds an action he also learn the assistant a new word connected with his word action().
So if the user adds three actions like working in the lab, meeting with the manager and phone call meeting which are conducted in the place Office, then in the VOCABULARY we have the following records:
Working in the lab => action(Working in the lab)
Meeting with the manager => action(Meeting with the manager)
Phone call meeting => action(Phone call meeting)

The application uses a broadcast receiver which monitors the state of the phone during a phone call. So when a phone call is occurred the broadcast receiver knows when the phone rings, when the phone called is hooked or when it is answered.
When the receiver senses the phone call and **before** the first ring, it creates the context of the phone call, meaning that it gathers all the data which are essential for the assistant to decide for the action.
The data answer the following questions:

- What day and time the phone call occurred? => gives date, day and time, if is weekend or workday, if it is holiday, the period of day (early morning, morning, Afternoon, evening, night)
- Where is the user during the phone call? => gives place
- What is he/she doing ? => gives the action which the user does that specific time and also gives the physical activity of the user (running, in vehicle, etc)

- Who is calling? => gives phone number, contact name as it is record in the log of the mobile
- What king of relation has the caller with the user? => gives the relation between them.
- How many times the caller called the user during the day? => gives the number of calls for this day so far.
- What is the time difference between the last two calls? =>gives the time difference between the last two calls.
- What is the time difference this call the last call? =>gives the time difference between this call and the last call.

All data create the context which the assistant comprehends. In this context we have the application of the rules which the user provides.
In the beginning the assistant has no rules so it will allow the call. After the phone call either the user answers the phone call or reject the call the assistant will display the following screen

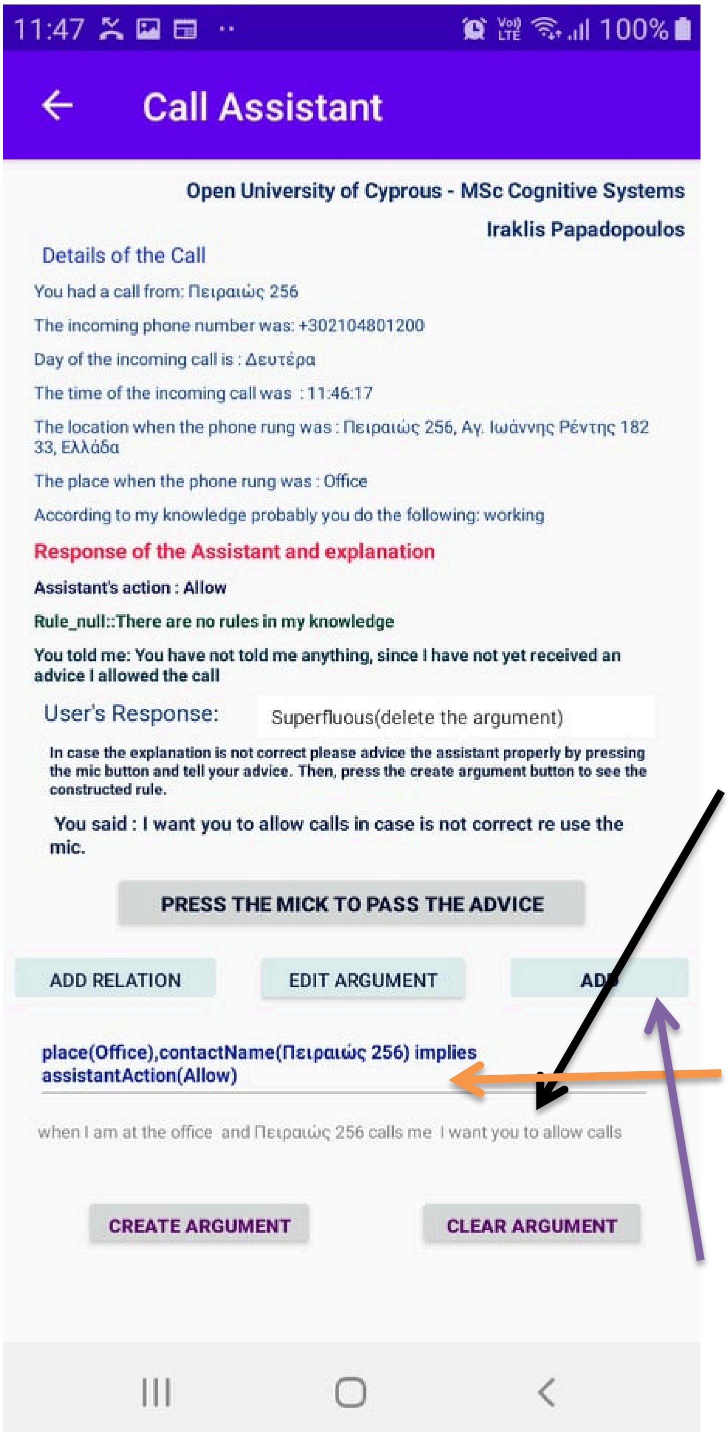


The assistant, in the top part of the screen, displays some of his knowledge, like the name of the caller, the phone number, the day, the time, the location, the place and the action.

The assistant informs the user that allows the call because it has no rules in knowledge base.

If the assistant does not know the relationship between the user and the caller then it asks ,by using voice, the user to tell him the relation. The user uses the MIC button and tells the assistant the relation, presses the ADD RELATIO button and record it the knowledge base.

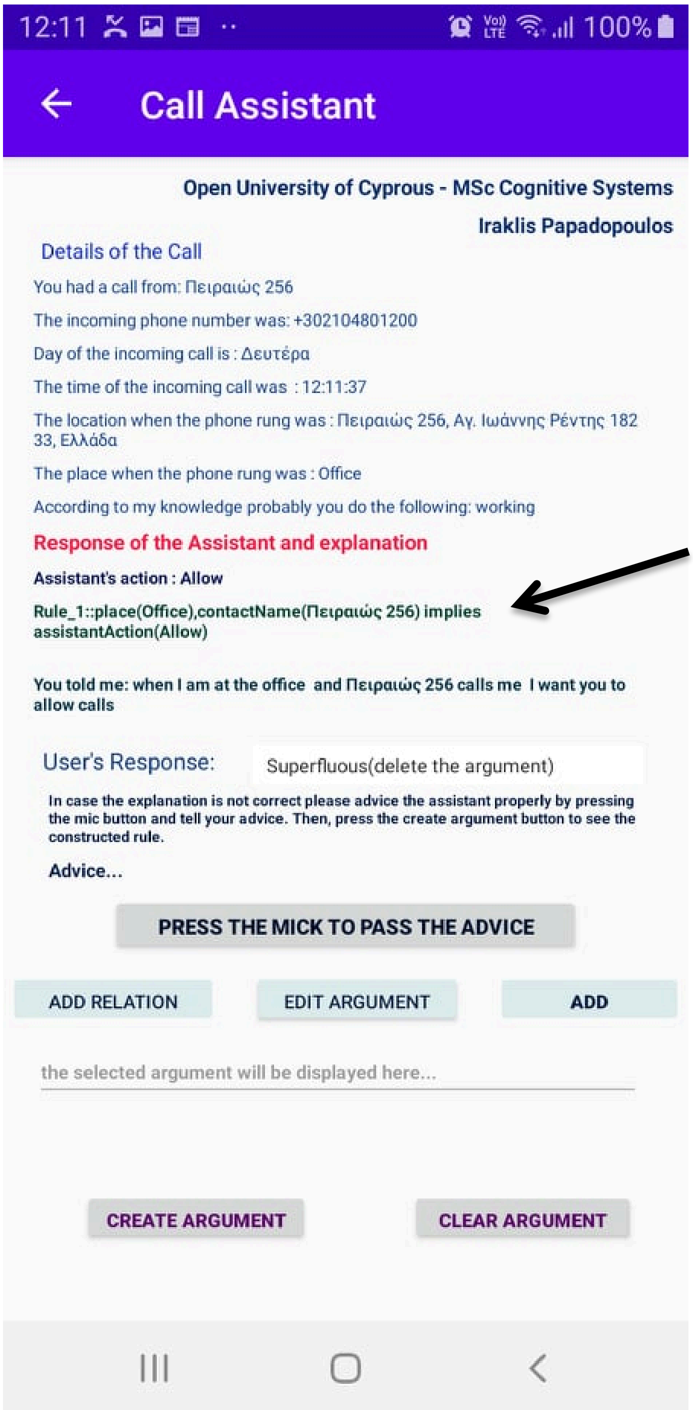
Like places and actions the VOCABULARY records the relation, so if the user states colleague we have:
Colleague =>relation(Colleague)



The user by pressing the mic tell the assistant the advice. The assistant breaks the advice into words and “cares” only for the words that have a meaning to him. So in this case the user tells

When I am at the Office and Πειραιώς 256 calls me I want you to allow calls

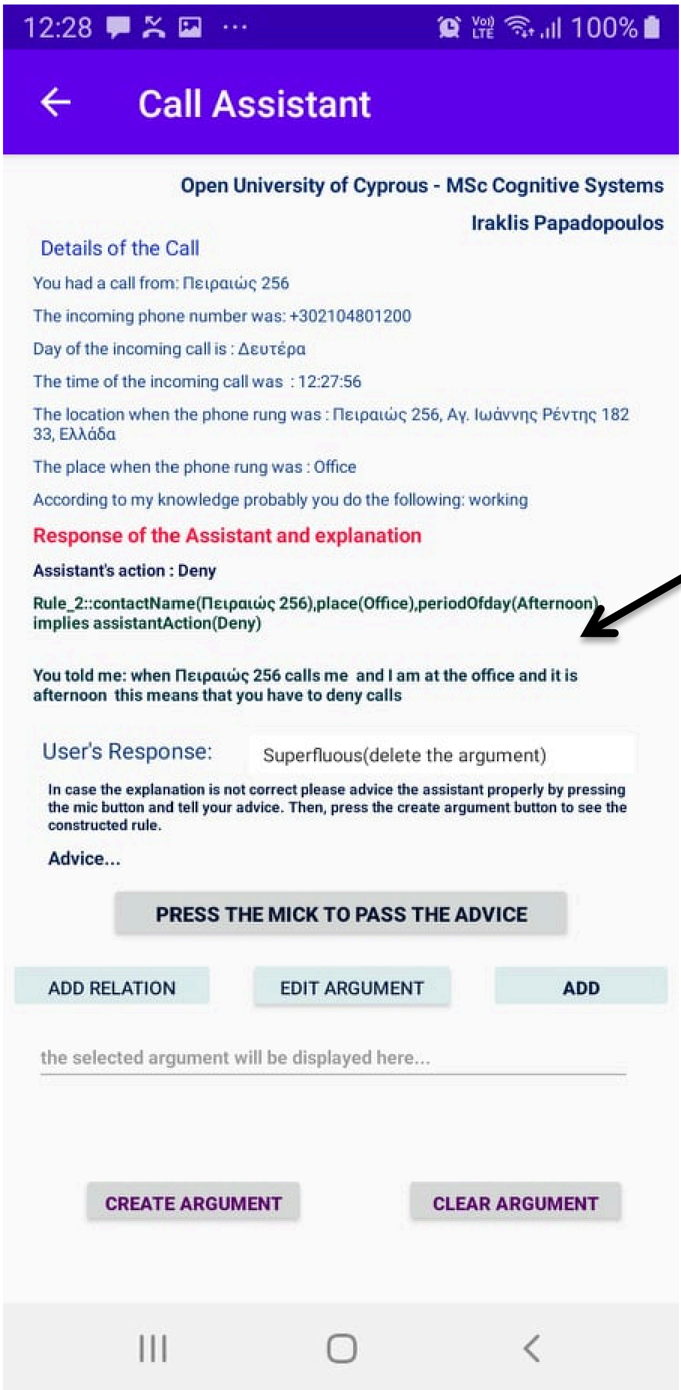
The assistant “understands” the Office which represent place and creates place(Office)
The word Πειραιώς 256 which represent caller and creates contactName(Πειραιώς 256)
The word want that represent implies and the word allow that represents assistantAction(Allow). Then the user press the ADD button and adds the rule with the advice in the knowledge base



Now, when the phone rings again and the user is at the office and the caller is Πειραιώς 256 then the rule 1 is triggered The assistant displays his action, the argument and the advice which the user gave

Now the user uses the mic button and adds a new advice
"When Πειραιως 256 calls me and I am at the office and it is afternoon this means that yoy have to deny calls"
The constructed rule is
contactName(Πειραιώς 256),place(Office),periodOfday(Afternoon) implies assistantAction(Deny)

now we have



The assistant deny the call since the place is office, the caller is Πειραιώς 256 and the time indicates that it is afternoon (the caller understands that the time between 12:00-17:00 is afternoon) so it displays its action, the rule and the user's advice

References

A Brief History of Cognitive Computing - DATAVERSITY. (2014). Retrieved 27 March 2021, from https://www.dataversity.net/brief-history-cognitive-computing/

Ahn, W.-k., Brewer, W. F., & Mooney, R. J. (1992). Schema acquisition from a single example. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18*(2), 391–412. https://doi.org/10.1037/0278-7393.18.2.391

Caroll, J. B. (1993). A Theory of Cognitive Abilities: The Three-Stratum Theory. *Human Cognitive Abilities,* 631-655. doi:10.1017/cbo9780511571312.018

Chai WJ, Abd Hamid AI, Abdullah JM. Working Memory From the Psychological and Neurosciences Perspectives: A Review. Front Psychol. 2018 Mar 27;9:401. doi: 10.3389/fpsyg.2018.00401. PMID: 29636715; PMCID: PMC5881171.

Baddeley A. Working memory: theories, models, and controversies. Annu Rev Psychol. 2012;63:1-29. doi: 10.1146/annurev-psych-120710-100422. Epub 2011 Sep 27. PMID: 21961947.

Byrne, R. (1989). Suppressing valid inferences with conditionals. *Cognition*, *31*(1), 61-83. doi: 10.1016/0010-0277(89)90018-8

CBSNews.com (2011, February 17). *IBM-Watson Defeats Humans in "Jeopardy!"* CBS News. https://www.cbsnews.com/news/ibm-watson-defeats-humans-in-jeopardy/

Chen M., Herrera F. and Hwang K., "Cognitive Computing: Architecture, Technologies and Intelligent Applications," in *IEEE Access*, vol. 6, pp. 19774-19783, 2018, doi: 10.1109/ACCESS.2018.2791469.

Committee, Nobel Prize, 2017. "**Richard H. Thaler: Integrating Economics with Psychology**," Nobel Prize in Economics documents 2017-1, Nobel Prize Committee

Donald A. Norman. 1994. How might people interact with agents. *Commun. ACM* 37, 7 (July 1994), 68–71. DOI:https://doi.org/10.1145/176789.176796

Edwards, G., Vetter, P., McGruer, F. *et al.* Predictive feedback to V1 dynamically updates with sensory input. *Sci Rep* **7,** 16538 (2017). https://doi.org/10.1038/s41598-017-16093-y

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., Schlaefer, N., & Welty, C. (2010). Building Watson: An Overview of the DeepQA Project. *AI Magazine*, *31*(3), 59-79. https://doi.org/10.1609/aimag.v31i3.2303

Fox, J. (2011). Artificial cognitive systems: Where does argumentation fit in? *Behavioral and Brain Sciences*, *34*(2), 78–79. https://doi.org/10.1017/s0140525x10002839

Fox J and Das S. (2000) *Safe and Sound: Artificial Intelligence in Hazardous Applications* Cambridge MA: MIT Press

Fox J, Beveridge M, Glasspool D. (2003) "Understanding intelligent agents: analysis and synthesis" *AI Communications*

Jackson, K. (2018, July 25). *A brief history of the smartphone*. Science Node. https://sciencenode.org/feature/How%20did%20smartphones%20evolve.php

Garrido, A. (2010). Mathematics and Artificial Intelligence, Two Branches of the Same Tree. *Procedia - Social And Behavioral Sciences*, *2*(2), 1133. doi: 10.1016/j.sbspro.2010.03.160

Gold, E. (1967). Language identification in the limit. *Information And Control*, *10*(5), 447-474. doi: 10.1016/s0019-9958(67)91165-5

Gonfalonieri, A. (2018, December 31). *How Amazon Alexa works? Your guide to Natural Language Processing (AI)*. Medium. https://towardsdatascience.com/how-amazon-alexa-works-your-guide-to-natural-language-processing-ai-7506004709d3

Handley, L. (2019, January 24). Nearly three quarters of the world will use just their smartphones to access the internet by 2025. *CNBC*. https://www.cnbc.com/2019/01/24/smartphones-72percent-of-people-will-use-only-mobile-for-internet-by-2025.html

Hastings J.S. , Shapiro J. M., November 2013, Fungibility and Consumer Choice: Evidence from Commodity Price Shocks*, *The Quarterly Journal of Economics*, Volume 128, Issue 4, Pages 1449–1498, https://doi.org/10.1093/qje/qjt018

HIRSCHMAN, L., GAIZAUSKAS, R. (2001). Natural language question answering: The view from here. *Natural Language Engineering, 7*(4), 275-300. doi:10.1017/S1351324901002807.

Hof, R. Google Tries to Make Machine Learning a Little More Human. (2015). Retrieved 27 March 2021, from https://www.technologyreview.com/2015/11/05/165175/google-tries-to-make-machine-learning-a-little-more-human/

Höffner K. , Walter S. , Marx, E., Usbeck R. , Lehmann J. Ngonga Ngomo A. (2016). Survey on Challenges of Question Answering in the Semantic Web. Semantic Web. 8. 10.3233/SW-160247.

IBM Watson products. (n.d.). IBM. https://www.ibm.com/watson/products-services
IT Infrastructure. (n.d.). IBM. https://www.ibm.com/it-infrastructure
"Intelligence," March 18, 2021. https://en.wikipedia.org/wiki/Intelligence.

Kakas, A & Michael, L. (2016). Cognitive Systems: Argument and Cognition. IEEE Intelligent Informatics Bulletin. 17. 14-20.

Kakas, A & Michael, L & Toni, F. (2016). Argumentation: Reconciling Human and Automated Reasoning.

Kakas, A. C., Moraitis, P., & Spanoudakis, N. I. (2019). GORGIAS: Applying argumentation. *Argument & Computation*. https://doi.org/10.3233/AAC-181006

Kampakis, S. (2020, November 26). *What deep learning is and isn't*. The Data Scientist. https://thedatascientist.com/what-deep-learning-is-and-isnt/

Kahneman, D. and A. Tversky. 1979. Prospect Theory: An Analysis of Decision under Risk. Econometrica 47, 263-291.
Kent. L. P. (2016) Working Memory: A Selective Review, Applied Neuropsychology: Child, 5:3, 163-172, DOI: 10.1080/21622965.2016.1167491

King, J. (2019, November 21). *Americans Check Their Phones 96 Times a Day* [Press release]. https://www.asurion.com/about/press-releases/americans-check-their-phones-96-times-a-day/

Kintsch, W. (1998), The Representation of Knowledge in Minds and Machines. *International Journal of Psychology*, 33: 411-420. https://doi.org/10.1080/002075998400169

Lake, B., Ullman, T., Tenenbaum, J., & Gershman, S. (2016). Building Machines that Learn and Think Like People. *Behavioral And Brain Sciences*, *40*, 17. doi: 10.1017/s0140525x16001837

Lin, Z., & Carley, K.M. (1993). Proactive or Reactive: An Analysis of the Effect of Agent Style on Organizational Decision-making Performance. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 2, 271-287.

Lippi, M., & Torroni, P. (2016). Argumentation Mining: State of the Art and Emerging Trends. ACM Trans. Internet Techn., 16, 10:5,10:18,10:19.

Making a thinking machine. (2021). Retrieved 27 March 2021, from https://www.apa.org/monitor/2018/04/cover-thinking-machine

Markoff. J How Many Computers to Identify a Cat? 16,000 (Published 2012), The New York Times, Retrieved 27 March 2021, from https://www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html

Matei, A. (2019, August 21). Shock! Horror! Do you know how much time you spend on your phone? *The Guardian*. https://www.theguardian.com/lifeandstyle/2019/aug/21/cellphone-screen-time-average-habits

Michael, L. (2019). Machine Coaching. *IJCAI 2019 Workshop on Explainable Artificial Intelligence (XAI @ IJCAI 2019)*

Michael, L., Kakas, A. C., Miller, R., & Turán, G. (2015). Programming cognitive. In *CEUR Workshop Proceedings* (Vol. 1510, pp. 3–18). CEUR-WS. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84961248082&partnerID=40&md5=dded16a9b49a07a201f3fe59b8a1ef07

Mitsikas, T., Spanoudakis, N.I., Stefaneas, P., & Kakas, A. (2017). From Natural Language to Argumentation and Cognitive Systems. COMMONSENSE.

Mozina, M. (2018). Arguments in Interactive Machine Learning. Informatica (Slovenia), 42.

M. Mozina, C. Giuliano and I. Bratko,(2009) "Argument Based Machine Learning from Examples and Text," *2009 First Asian Conference on Intelligent Information and Database Systems*, Dong hoi, Vietnam, 2009, pp. 18-23, doi: 10.1109/ACIIDS.2009.60.

Mpelogianni, V., Groumpos, P.P. Re-approaching fuzzy cognitive maps to increase the knowledge of a system. *AI & Soc* **33,** 175–188 (2018). https://doi.org/10.1007/s00146-018-0813-0

Nielsen, M. Neural Networks and Deep Learning. (2019). Retrieved 27 March 2021, from http://static.latexstudio.net/article/2018/0912/neuralnetworksanddeeplearning.pdf

Ojha, V. K., Abraham, A., & Snášel, V. (2017). Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence*, *60*, 97–116. https://doi.org/10.1016/j.engappai.2017.01.013

Patanella D., Ebanks C. (2011) Gardner's Theory of Multiple Intelligences. In: Goldstein S., Naglieri J.A. (eds) Encyclopedia of Child Behavior and Development. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-79061-9_1870
Physics Boosts Artificial Intelligence Methods. (2021). Retrieved 27 March 2021, from https://www.caltech.edu/about/news/physics-boosts-artificial-intelligence-methods-80127

Prelec, D., & Loewenstein, G. (1998). The red and the black: Mental accounting of savings and debt. *Marketing Science, 17*(1), 4-28.

Prokop, D. (2021). Von Neumann–Morgenstern utility function | decision theory. Retrieved 19 April 2021, from https://www.britannica.com/topic/von-Neumann-Morgenstern-utility-function

rate, T., G, R., & Kataykin, P. (2021). Text to Speech get System default pitch and speech rate. Retrieved 5 May 2021, from https://stackoverflow.com/questions/47084476/text-to-speech-get-system-default-pitch-and-speech-rate

READ, D., LOEWENSTEIN, G., & RABIN, M. (1999). Choice Bracketing. *Journal of Risk and Uncertainty, 19*(1/3), 171-197. Retrieved April 19, 2021, from http://www.jstor.org/stable/41760959

Sanjay D. , Vaishali S. (2013). Research and Reviews in Question Answering System. Procedia Technology. 10. 417–424. 10.1016/j.protcy.2013.12.378.

SHEFRIN, H., & STATMAN, M. (1985). The Disposition to Sell Winners Too Early and Ride Losers Too Long: Theory and Evidence. *The Journal Of Finance*, *40*(3), 777-790. doi: 10.1111/j.1540-6261.1985.tb05002.x

Simon, H. A. (1955). A Behavioral Model of Rational Choice. Quarterly Journal of Economics, 69, 99-118. https://doi.org/10.2307/1884852

Sternberg, R. J. (1985). *Beyond IQ: A triarchic theory of human intelligence.* Cambridge University Press.

Thaler, R. (1985). Mental Accounting and Consumer Choice. *Marketing Science*, *4*(3), 199-214. doi: 10.1287/mksc.4.3.199

University of Pennsylvania School of Medicine. (2006, July 28). Penn Researchers Calculate How Much The Eye Tells The Brain. *ScienceDaily*. Retrieved March 27, 2021 from www.sciencedaily.com/releases/2006/07/060726180933.htm

Vernon D., Metta G. and Sandini G., "A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents*," in IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp. 151-180, April 2007, doi: 10.1109/TEVC.2006.890274.

Vignesh, M. (2017, September 4). *Introduction to Cortana Intelligence Suite*. Microsoft TechNet Articles. https://social.technet.microsoft.com/wiki/contents/articles/36688.introduction-to-cortana-intelligence-suite.aspx

Vodafone. (2017, December 4). *25 years since the world's first text message*. Vodafone.Com. https://www.vodafone.com/news/technology/25-anniversary-text-message

Yorke-Smith N. , Saadati S., Myers K.L., Morley D.N. (2012) The Design of a Proactive Personal Agent for Task Management Int. J. Artif. Intell. Tools, 21 (01) , p. 1250004

Wharton, C., Rieman, J., Lewis, C. and Polson, P. 1994. *The Cognitive Walkthrough Method: A Practitioner's Guide*. Tech. rep., New York, NY, USA.

Zhou, S. (2016) The Newsvendor Problem Based on Multiple Reference Points. *Open Journal of Social Sciences*, **4**, 227-240. doi: 10.4236/jss.2016.49019

Zimmermann, H. (2010). Fuzzy set theory. *Wiley Interdisciplinary Reviews: Computational Statistics*, *2*(3), 317-332. doi: 10.1002/wics.82
.