DEGREE PROJECT, IN CINTE , FIRST LEVEL
STOCKHOLM, SWEDEN 2015

Intelligent Kernel of Emotions - IKE

AN APPROACH TO AN AI-COMPLETE SYSTEM

OSKAR ANDERSONS AND FILIP BARK

KTH ROYAL INSTITUTE OF TECHNOLOGY

ICT

Abstract

Ever since the creation of the first computers, humans have pondered the possibility of a computer capable of human-like thought and reason. Many sci-fi authors have explored the possibilities and consequences of a sentient computer and Artificial Intelligence. However, so far no one has managed to come close to an actual thinking computer. This is possibly because the core structure of a computer is based on binary truths, "ones" and "zeros", while the human brain does not work with only true or false.

This thesis presents the creation of an AI prototype called IKE. It is not an AI that can act like a human, but rather one that is a step closer to that final goal of a thinking computer. In this thesis, this is done by merging simulations of the five major senses with artificially created feelings, which are then fed into the system's memory. These are in turn embedded into objects and stored in the memory as instances of those objects. The goal with this method is that when you give the system an object, it can tell you how it feels about the object and what experiences it associates with it. This is to give the computer a sense of simulated cognition which will allow it to answer in a human-like way.

The result of this work is a system that can be seen as an approach to an AI-complete system. It is an AI that has feelings and senses, as well as the ability to express them. However, it cannot gather or make up new feelings nor experience them in any way, as this is left for a future project to solve.

Keywords

Cognition, AI-Complete System, Artificial Intelligence, Simulation of senses

Abstract

Sedan de första datorerna skapades har människan funderat över möjligheten att skapa en dator som kan tänka som en människa. Många sci-fi författare har utforskat möjligheterna och konsekvenserna med en tänkande dator och Artificiell Intelligens. Dock har ingen ännu kommit i närheten av en dator som faktiskt tänker. Detta kan bero på att datorn i grunden endast arbetar med ”ettor” och ”nollor”, medan den mänskliga hjärnan inte alltid svarar sant eller falskt.

I den här avhandlingen presenteras utveklingen av en AI-prototyp kallad IKE. Det är inte en AI som kan agera som en människa, utan en prototyp som är ett steg i rätt riktning, närmare målet att skapa en tänkande dator. Detta görs i denna avhandling genom att slå samman de fem sinnena med artficiellt skapade känslor som matas in systemets minne. Där sparas de i minnet some instanser av objekt med associationer till de faktiska objekten. Målet med denna metod är att givet ett objekt ska systemet kunna säga vad den tycker om objektet utifrån tidigare erfarenheter. Detta ger systemet en känsla av simulerad kognition som ska gör det möjligt att svara på ett mänskligt sätt.

Resultatet av detta arbete är ett system som kan anses vara ett steg närmare AI-Komplett. Det är en AI som kan spara, ta hand om och uttrycka sig om känslor och sinnen. Dock har den ej funktionaliteten att på egen hand samla in nya känslor eller att uppleva dem naturligt, utan detta lämnas till ett framtida arbete.

Nyckelord

Kognition, AI-Kompletta System, Artificiell Intelligens, Simulering av sinnen

Table of Contents

1 Introduction

Early in the 20th century, the first computer was conceived based on the work and findings of British mathematician Alan Turing[1]. Since then the possibilities for the wonder machine have been endless, limited only by software developers' imagination and skill. An idea that has captivated scientists and developers alike for many years is Artificial Intelligence, or AI for short[2]. Since the term was coined during the 1950's, development of applications utilising AI has been thriving[3]. This is especially true in recent times where the world is becoming increasingly digitised and most homes in the civilised world have a computer and almost everyone has access to the internet.

One area where AI has gained a lot of attention is in games where the player's opponent is controlled by the computer, for example in chess. Science fiction has also taken a liking to AI, where the capabilities of AI do not necessarily have to be bound by the progress of research and the abilities of a highly advanced AI can be explored. For one such example, *HAL9000* was a well-developed AI seen in the feature film *2001: A Space Odyssey*. It was capable of reason, thought and action, albeit being a bit homicidal[4]. Murderous tendencies aside, the *HAL9000* was able to complete very complex tasks in a way similar to that of a human, which is by far more advanced than any AI currently in existence[4]. But that is something that AI researchers are trying to create and the problems that have to be solved in order to implement the various missing features are aptly labelled "AI-Complete" or "AI-Hard"[5]. These problems are the main hindrance standing in the way of achieving something akin to a sentient computer program, a program capable of thought and reason[5]. The IKE system contained within this report is an attempt to address a subset of these problems, as it will incorporate feelings. The name that IKE was given stands for Intelligent Kernel of Emotions, but there will be more on this later.

1.1 Background

Before discussing AI-Complete further, the concept of AI needs to be further explained. The term Artificial Intelligence implies a way of making a computer think or having it act as if it were a human. The word act here is key, because despite how far technology has come, the world of sci-fi and intelligent robots is still far away. What separates the current day AI from the potential future AI is the human part, the ability to imagine and dynamically come up with solutions to problems, to think outside the box[6]. The term for a system that has all of these features is "AI-Complete" and the problems surrounding the creation of such a system are in turn AI-Complete problems.

These problems are not very well defined, but it is generally agreed upon that solving one of the problems would lead to being able to solve the other problems as well[7]. In a way, it could be said that an AI-Complete problem is only labelled as such once researchers realize that they require more work than was originally expected[7]. However, among the myriad problems there are some that are specified, two of which are the matters of how to simulate the senses and emotions of a human being[8]. These problems could be classified as "digital cognition", the ability for an AI to observe and think about objects, including itself[5].

1.2 Problem

Sentience, or self-awareness above an instinctive level is what most AI-Complete problems are about. Whether it is granting eyesight or the capacity for speech and the sharing of thoughts to an AI, the core problems of AI-Completeness are not actually that well-defined yet. As it turns out, scientists today are not entirely sure what it is that is still missing.

In this thesis, the problem of creating an AI smart enough to mimic or even get close to rivalling human beings will be brought up while not exactly being addressed, but it will affect the work and ideas going into IKE. The main problems with IKE, while certainly AI-Complete, are not sight or speech, but one level above that. Instead, it is emotions and feelings, which build on what IKE experiences with the eyesight and speech. At this stage in development and research, the core was to at least give IKE the capability to register and handle feelings and to give a framework for handling simulations of the five human senses in relation to the feelings.

Therefore, the question to be answered within this thesis is this: how could a system containing all of the five human senses and feelings be constructed and how would its memory architecture look?

1.3 Purpose

The aim of this thesis is to present the development of a prototype of a partial AI-Complete system, including a model and an implementation thereof. That is, a system with some small semblance of digital cognition. The model will be programmatically implemented in order to test that it truly works. The digital cognition will enable IKE to recognize an object and share its thoughts, memories, feelings and other knowledge of the object.

1.4 Goal

The goal of this system is to provide results for researchers within the field of AI. A complete solution to any AI-Complete problems will not be given in this report, as it is meant to be a small step on the way to a solution. This is done in order to implement what is known today, but more importantly to find out what is still missing before the AI-Complete problems can be solved and a truly sentient computer program can be created.

1.4.1 Benefits, Ethics and Sustainability

The primary benefactors of this project are researchers in the field of AI. While the contribution of this report to the research field will be small, the lessons learned during this project aim to be of benefit to someone else trying to do the same or build upon what has already been created. This particular approach to creating an AI that has emotions is taken in order to create a more human-like computer. Human emotions are paramount to achieving a kind of empathy, which in the future could lead to robots being able to do jobs that require a certain amount of empathy, like for example working at a day care center or caring for disabled and elderly people. In this sense, once the researchers benefitting off this thesis can work out the remaining problems, AI may finally become classified as human.

As for ethics, when creating an AI one might consider it akin to creating life. However, what is done in this thesis is not granting life and taking it away every time the program is restarted just to do some minor changes[9]. Another important ethical question to keep in mind is how the AI is to be used. While this thesis aims to further AI research, there is the possibility that an AI might be used for more malevolent purposes. For example, the armed forces of certain countries today are already looking into the usage of AI in unmanned drones for the purpose of warfare[10]. While IKE is aimed to further the research of AI and enhance the lives of humans, others might instead use the system to take human lives and make them miserable. IKE as a system is not inherently malevolent and as such the fault would lie with the user who uses IKE in unintended ways.

Regarding sustainability, the focus in this thesis will be on environment, economy and society in general[11]. For the project to be sustainable, it has to have more positive associations than negative in these three areas.

As far as the environment is concerned, there might not be any apparent impact from the development of this system, but this is not necessarily true. While there are indeed no direct emissions that would impact the environment, there will be indirect emissions in the form of electricity being consumed by the computers required for both development and testing of IKE as a system. However, the electricity used is nowhere near levels that would affect overall electricity production and as such these emissions are negligible.

The economical factor is a non-question as there is no monetary compensation for the development of the system and the lack of compensation does not cause any loss of money.

The societal question is more complicated. As already mentioned, the ethical question of who will use the system and for what purposes has to be carefully considered. While the intention is that IKE is to serve as a step towards AI-Completeness, there is still the possibility that someone may attempt to exploit the system with ill intentions. This is where IKE's feelings and empathy would enable it to make its own decision whether to follow an order based on its own experience. However, for IKE, an incomplete AI system, this is a moot point as there is no functionality implemented which can be exploited. All IKE does is answer questions.

As such, IKE is a sustainable system in the sense that it has no real negative impact on the world, be it environmentally, economically or on society as a whole. On the contrary, if future research can build upon what is uncovered in this thesis, the completion of an actual AI-Complete system could be realised.

1.5 Methodology

When taking on a project such as this thesis, choices regarding procedures to be used have to be considered very carefully[12]. In many cases, these choices are made retrospectively, diminishing the effectiveness of the entire study. This part will cover the philosophical assumptions, research approaches and methods applicable to this type of project.

1.5.1 Quantity vs. Quality

To reduce the amount of actually applicable methods, a major choice to be made is whether to do quantitative or qualitative research[12]. While these are fundamentally different to each other, they are not mutually exclusive. There is a third option, triangulation, which is a combination of the two[12].

- *Quantitative* – As the name implies, quantitative research mainly values quantity of data over quality. Meaning that a vast amount of data is studied and by using various statistic methods a hypothesis can be tested[12]. This is a good idea for experiments and testing where the hypothesis can be proven or disproven by numbers[12].

- *Qualitative* – On the other hand, the qualitative view of quality over quantity places a heavier focus on a smaller number of tests yielding more valuable and concrete information[12]. Where quantitative research is concerned with proving a hypothesis, qualitative research is more about reaching an understanding for the chosen subject before forming a hypothesis about it[12].

- *Triangulation* – Triangulation is the combination of the two previous, where both are applied but only one at a time. One option could be to first do qualitative research to reach a hypothesis and then quantitatively prove it.

As this thesis is interested mainly in the developing a model, qualitative research is the preferred option. While quantitative research is good for proving hypotheses, what is needed here is to find valuable information and create a solid understanding needed to develop a system.

1.5.2 Philosophical Assumptions

Once a type of research has been decided, the overall mind-set with which to go about gathering information, the philosophical assumptions have to be decided. The available options described next are positivism, realism, interpretivism and criticalism.

- *Positivism* – A good mind-set for evaluating systems and their performance. It works under the assumption that the observer has no impact on the results and that conditions under which the experiment is done, the results are always objective[12].

- *Realism* – A more direct mind-set where any observable truths are considered truths[12]. Assumptions can only be made based on what has been observed and need to be proven with facts found in the collected data[12]. Understanding and knowledge can be developed from there[12].

- *Interpretivism* – Useful when developing solutions "artefacts", which will be further explained in chapter 3. In terms of computer systems interpretivism, it is assumed that results are subjective, meaning that depending on who is doing the experiments and how affects the results[12].

- *Criticalism* – Much like with interpretivism, criticalism works on the assumption that reality is subjective and mutable. However, rather than looking at the changes, the cause behind them is of greater interest and is usually found in history and culture, rather than the observations[12].

For this thesis, interpretivism has been chosen as the predominant mind-set. This is because its focus on experiences works well with qualitative research. The strengths of interpretivism come through well in the creation of software systems. Most of the data will be collected from reports and books and then be compared to create a solid understanding of the subject before building upon it.

1.5.3 Research Approaches

But before any research, the choice of how to observe and think about the results, the approach to the research has to be made. Rather than how information is gathered, it is more about how it will affect the end result. There are three main ideas for this and they all centre on a theory and its proof. The available options described here are inductive, deductive and abductive.

- *Inductive* – The inductive approach starts with observations and works towards a hypothesis and culminates in a theory[12]. In other words, the first step is gathering vast amounts of data, after which the data is studied in order to get a general grasp of the subject[12]. Then conclusions are drawn and a theory can be formed which can later be proven[12].

- *Deductive* – The deductive approach is the polar opposite of the inductive. Here, you start with a theory that needs to be proven[12]. That is, the deductive approach is used to prove a theory, rather than constructing it[12].

- *Abductive* – The abductive approach is a combination of the two, where theory is both formed and proven[12]. This approach is often used when the data collected does not encompass a complete data set. Preconditions are then used in order to infer or explain conclusions[12].

With qualitative research and interpretivism already chosen, the logical choice is to go with the inductive approach, since there is no initial hypothesis to prove. Instead, the goal is to find out what is still missing and what needs to be further developed. That will be the hypothesis: what needs to be fixed before an AI-Complete system can be realised.

1.5.4 Research Methods

Once an approach has been decided on, the next decision to make is what methods to use in gathering data. There is a wide variety of different methods, all with their own advantages and disadvantages. Should one method prove to be insufficient and unable to fill all needs, it is of course possible to incorporate ideas from other methods, within reason[12]. The available options described here are experimental, non-experimental, descriptive, analytical, fundamental, applied, conceptual and empirical.

- *Experimental* – Useful for evaluating performances of systems, the experimental method studies the results gained from experiments[12]. The variables, the relationship between them, causes and effects are the core interest with this method[12]. It's possible to see what a change in one variable can cause while other variables are kept unchanged[12].

- *Non-experimental* – In contrast, a non-experimental method instead uses previously known scenarios to draw conclusions[12]. Rather than the details surrounding variables and their relationships, the bigger picture is more important[12].

- *Descriptive* – This method studies phenomena and aims to describe characteristics, but as with the non-experimental method it ignores the causes for situations[12]. This method works well with both quantitative and qualitative research, often combining or stemming from surveys and case studies[12]. The core idea is to find new data in already existing data by looking at a large set of known facts[12]. Also sometimes called "statistical research".

- *Analytical* –If the purpose is to prove a hypothesis, this is the method of choice[12]. Much like the previous two methods, it uses already known facts, analysing it to prove a hypothesis through a critical evaluation of facts[12].

- *Fundamental* – For research purposes, this is a method which results in new ideas, solutions, principles and theories[12]. It is often called "basic research" or "pure research" and is mostly driven by genuine curiosity[12]. It can also be applied to challenge or modify existing theories, but the core idea is to gather more data by studying and observing different phenomena[12].

- *Applied* – This could be interpreted as a combination of fundamental research and analytical research, as it is mostly used to prove or disprove hypotheses[12]. It draws from both existing research and own research and the results and circumstances are studied in detail[12].

- *Conceptual* – Conceptual research is a much less concrete method. Rather than reaching a conclusion or to proving a theory, it is mostly used to get a general idea of a concept or a subject[12]. This has more practical use when developing a theory or when learn the history of a subject[12]. An example of when this could be used is when writing about the history of AI or explaining the concept of AIC.

- *Empirical* – Empirical research is based entirely on experience and focuses on facts proven or obtained first-hand. From these facts and observations, conclusions are drawn about the world and the people in it[12]. Theories, characteristics, evaluations and relationships are key[12]. This is a no nonsense type of methodology that requires irrefutable evidence[12].

While perhaps not as apparent a choice as previously, the most fitting method is conceptual method, as the goal is to create a conceptual system.

This choice is mainly supported by the need for a solid understanding of the background when creating a system.

1.5.5 Data Study

With all necessary choices being made, the research can be started. In order to gather as much data as possible in the shortest amount of time, the internet is a very handy tool. More specifically, the Compendex database for scientific reports and papers is the main search tool[13; 14]. A simple search in this database might produce hundreds of thousands results, but with a bit of refinement it is possible to find a smaller collection of only the most relevant results. After cutting down on the amount of initial results, critically examining the abstracts of each result will leave only the few most relevant results that have to do with either AI or AI-Complete, the two areas of interest for this thesis.

Regarding AI, mostly history, which is grounded in reality and has concrete information written about it, is of interest. A lot of studies and reports on various applications of AI will be documented during the study, both as reference material during the actual work and for the historical context.

As for AI-Complete, most interesting are recent applications trying to approach AI-Completeness in order to see how others have tackle the problems of the subject.

1.6 Delimitations

Despite the knowledge that can be gained on both AI and AI-Complete, several features that might be considered essential to IKE cannot be implemented and some topics will not be brought up after this point in the paper.

The history of AI will only be briefly mentioned. A select few highlights that lead to the current concept will be brought up, but a more thorough analysis will not be given.

Furthermore, IKE is to be capable of thought and reason regarding objects through the knowledge and senses that human possess, brought together similarly to the human brain. For this, assignment of feelings and input from senses is a necessity, but as these are technically full-scale projects on their own, they will not belong to IKE’s core functionality. As such, any sensors required for the emulation of human senses such as sight and hearing will not even be mentioned past this point. To reiterate, the focus of this thesis is to make the feelings and senses work together and not to enable IKE to gather new input.

Another important aspect of any system is its user interface, it's UI. As IKE is not really intended for end users, focus will not be on the development of a visually pleasing interface. Instead there will only be a simple command line prompt with which to query IKE.

Furthermore, a way for IKE to assign feelings will not be developed either, since this is a complicated matter which includes more to psychology than is relevant at this point.

Another vital feature for a system of this kind is a dynamic memory, but this system will work with a smaller, static memory. As such, the only editing or adding of memories that will or can be done for this system is to be done by editing the stored "memory file".

Without the ability for IKE to learn of new experiences and to experience them for itself, its memory will consist of a hardcoded selection of objects and feelings. These hardcoded examples to be implemented will be decided further on and will be detailed together with the reasoning behind their inclusion in the fourth chapter.

Additionally, while IKE will be able to simulate the feelings it is given, it will not be able to act differently depending on whether it is, for example sad or happy. Implementing such functionality is yet another full project in its own right.

1.7 Outline

The next chapter of the thesis, Background gives a more detailed description of the background of AI and a brief look into the computer's history as the two are closely related. Besides history, a look into some concurrent and related work will also be given.

Following this, the third chapter, Methods will list some of the available methods for the phases of pre-development research and actual development of IKE, much like the paragraph on methodology in chapter 1.5. There will also be a brief discussion of what methods for system development exist. Apart from listing the available options, choices will be made and explained.

After this, in the fourth chapter, Prototyping is a more detailed explanation of the chosen method of system development, detailed to a level that every part of it is explained, but not detailed enough to give a full description of the method. Information on that level would require more than a few pages, possibly even an entire book.

Naturally, after the description on how work has been done is the chapter called Developing IKE that details the actual work that went into creating IKE. Details will not be on the level of a diary, but will deal with the work that has been done in a broad sense, describing things such as what was done, why and how. The results of the work are detailed next, in the sixth chapter called IKE. Details there include IKE's current functionality.

Thereafter is a short chapter which was named Results, which evaluates IKE, determining whether or not all requirements were met and how they were met, as well as an overview of some possible user interactions. These will be reflections on IKE as a system.

Reflections on the project as a whole are in the final chapter called Conclusions. This chapter will also contain a discussion concerning what went well, what did not and what could have been done differently. Finally, there is also a short description of future work and what is left until IKE can be considered a full-fledged AI-Complete system

2 Background

The second chapter of this report will expand on the background, history and theory behind Artificial Intelligence, as well as the definition of AI used in this thesis. When discussing the history of AI as it is known today, it would be almost impossible to not mention the history of the computer, as the two are very closely related. There will also be a segment on some of its current applications. Following the historical background there will be an explanation of the concept of AI-Complete. As it is not very well defined, it is a difficult subject to discuss. The main reason for this is because it encompasses every aspect of AI that is not yet defined or solved. Finishing up the chapter is a definition of IKE’s senses.

2.1 A Brief History of Artificial Intelligence

While there are a few different opinions on when the concept of AI was actually conceived, most agree that the expression “Artificial Intelligence” was coined in 1955 in a preparatory document by John McCarthy and his colleagues for the 1956 Dartmouth Conference[2; 3]. Some argue that the father of AI was actually the British mathematician Alan Turing, who in turn based his work on the efforts of Kurt Gödel[2].

The very same Kurt Gödel who laid the foundations of computer science theory when he, by constructing an axiomatic self-referential system for statements, proved that contemporary math had to evolve[2]. This was all done during the 1930's and about ten years before the first actual computer was constructed during World War 2 by the British in order to decipher encrypted German intelligence[1]. It was called "Colossus" and was completed towards the end of 1943 by Thomas H. Flowers[1]. He then helped Turing create the more famous computer that cracked the Enigma code and is commonly mistaken as the world's first computer[1].

Other than machines for deciphering German intelligence, the 1940's also saw the development of the first programming language and the first application of it, a chess program which can be considered a sort of AI[2]. Turing also devised a test in his paper, "The Imitation Game", which was used to decide whether or not a computer was intelligent[2; 15]. Furthermore, what

Alan Turing did when building upon Gödel's work was construct his now well-known Turing Machine, which laid the foundations for computers and in turn AI[2].

Then came the 1956 Dartmouth Conference where John McCarthy et al coined the term "Artificial Intelligence", after which the subject came to attention and researchers around the world started investing time in it[3].

2.2 Artificial Intelligence

Before to the topic of what has happened in the field of AI research and where it is now, a brief explanation of the Turing Machine introduced previously is in order. Of course, that is not all Alan Turing contributed with to the field of AI research. Another theory that is often referenced when researching AI and especially when evaluating them is his other theory. But as with everything in the world of science, not everyone is of one mind as to whether or not any one theory is truly applicable or even if there is any benefit in applying a certain theory to use as a measurement for the intelligence of an AI.

2.2.1 Turing and AI

Alan Turing has contributed with two vital parts in the creation of AI and the assessment thereof. The first, the Turing Machine, is what all modern computers are based on. His second contribution is the famous Turing Test, explained in his article "The Imitation Game" in the 1940's[15].

The Turing Machine was not a physical machine, but rather a hypothetical computation device whose purpose was to transform numbers according to a predetermined set of rules[16]. The machine laid the foundations for computer science and all modern computers are based on it[2]. In extension, it can be said that the Turing Machine laid the foundation for all AI research, as the theoretical machine tested all possibilities and made choices, "knowing" how to proceed, much like the current concept of AI[16].

As for the Turing test, the "Imitation Game", it is a practice used to test whether a computer has intelligence or not[17]. The test is however not unanimously accepted as formal proof of a proper AI, the reason being the fundamental structure of the test which requires counter-productive compromises to the intelligence in order to "win"[17]. The Turing Test takes the program you want to test and two humans. One of the humans acts as judge while the other acts as counterpart to the computer[17]. The judge asks questions to both computer and human, compares the results and decides which of the two is most likely a computer[17]. This is where the counter-productive nature of the test comes into play[17]. In order to pass the test, having the computer make certain errors makes it seem more human[17]. For example, during one test, a human was mistakenly classified as a computer with the explanation that "no one could possibly know so much about Shakespeare"[15].

However, intentionally limiting the capabilities of a computer to make it appear human is not what Artificial Intelligence is about and this is where researchers start to argue[17]. When creating an AI, it is worth considering whether achieving something of human-like intellect is good enough for an AI or whether scientists should aim higher[17].

Some argue that human intelligence is the pinnacle and that the goal of AI development is to be able to mimic that, whereas other scientists, such as J. B. Pollack, argue that scientists should indeed aim higher, a so-called "Mindless intelligence" with the human factor eliminated[18].

2.2.2 Definition of Artificial Intelligence

While the Turing Machine was not an AI, Turing did come up with a way to test whether a program was "intelligent". However, both the Turing Test and the general nature of AI was questioned[18].

The Turing Test can also serve as a definition of AI, as AI scientists Stuart J. Russell and Peter Norvig tell in their book *AI: A modern Approach*[19]. They also give three other definitions of AI, totalling four distinct definitions of AI. The first definition, drawing from the Turing Test, is that the system in question can act human-like[19]. The second is that a system actually can think and reason like a human without having to pretend[19]. The third and fourth definitions are the system's capability of rational thought based on previous experiences and its capability of acting rationally[19]. As these definitions coexist, an AI system according to one definition is not necessarily labelled AI according to a different definition[19].

In fact, another scientist, Robert Trappl defines AI through four other statements, as described in his introduction to AI. His first statement is that AI is simply a way of making computers smart[20]. The second states that AI aims to better model human intelligence[20]. The third statement and longest statement says that the goal of AI is to make machines intelligent enough to move and act like humans in a dynamic environment[20]. Lastly, the fourth states that AI just aims to make machines increasingly useful[20]. As with Russell and Norvig's definitions, any one of these statements can serve as a definition on its own[20].

This together with J.B Pollack's opinion that AI should strive for something greater than simulating human intelligence leads to a rather diffuse definition, a definition with several different statements which in reality might not be so far apart[18; 19; 20]. Instead the definition of AI might lie in what they are capable of rather than some philosophical or physiological term.

"But what can they do?" asked the children of an elementary school when Hod Lipson showed them small, self-designed and 3D-printed machines[6]. His machines could learn new things on their own and were made from 3D-printed parts which had evolved in a simulation[6]. Though his robots were no doubt intelligent, as they had not only designed themselves but also taught themselves complex skills such as locomotion, Lipson brings up an interesting point: "If it is just doing what it was designed to do, is it truly intelligent?"[6]. He argues that an AI should not only be able to solve the task it was designed for, but also use its knowledge to synthesize new solutions to problems that may not be its primary task[6].

For this thesis, the definition of AI is a system that is capable of acting like a human, much like the ones accepted by the Turing Test. An AI system complex enough to fool a human into thinking the system is in fact human. But for such a system, several parts are required, among them parts that are not yet complete by today's standards. In fact, there is a word for a system with all of these parts incorporated: AI-Complete, which will be discussed later after a short introduction of some concurrent work in the field of AI.

2.2.3 Related work in AI

While an intelligent computer is certainly an attractive prospect, what is really interesting to most people is what they actually do. In his introduction to AI, Trappl lists some fields where AI research is prevalent: Knowledge representation, reasoning, learning, natural language processing, vision and robotics[20]. A combination of these fields has resulted in what is known as "chatbots". These are a type of AI that tries to simulate a human in a chat environment[21]. One such bot is called Wallace, which builds upon a well-known chatbot called Alice[21]. The main problem with the chatbots is that they have a fixed amount of hard-coded knowledge. Wallace works around this by looking for new and seemingly relevant information on Wikipedia while talking to someone[21].

Another application of AI is a program to sort amber gemstones[22]. This is a slow process when done by a human and is based on the weight, size, colour and other arbitrary factors of the gemstones[22]. Researchers are working on a program that can scan the amber to receive this information and then sort the gem stones in a quicker and more efficient way[22].

Yet another example is Webcrow, an AI application with the primary function of solving crossword puzzles[23]. A crossword puzzle requires more than just an understanding of natural language, it also requires a vast knowledge base spanning a great number of topics[23]. Webcrow solves the need for knowledge like Wallace, gathering information from the internet to find relevant information with which to solve the crosswords[23]. What makes Webcrow interesting is that the task it tackles, solving crosswords, is a challenge that has been labelled AI-Complete[23].

2.3 AI-Complete

When imagining what AI could become, it is easy to be drawn into imagining famous examples of advanced AI from science fiction movies and books such as *HAL9000* from *2001: A Space Odyssey*[4] or the homicidal *Skynet* from *Terminator*[24]. These were AI capable of thought on par with, or sometimes greater than, human intelligence. Needless to say, the crossword solvers and gemstone classifiers of today are a far cry from the intelligent computers found in fiction and what the missing pieces actually are is what scientists today have yet to agree on. This problem is what has come to be known as *AI-Complete*[7].

2.3.1 Definition of AI-Completeness

In the Jargon File, Eric S. Raymond writes that the term AI-Complete is "used to describe problems or subproblems in AI, to indicate that the solution presupposes a solution to the `strong AI problem' (that is, the synthesis of a human-level intelligence). A problem that is AI-complete is, in other words, just too hard."[7]. Basically, an AI-Complete system would be a full-fledged AI that is on par with fictional AI like Skynet and HAL9000.

The AI-Complete definition chosen for this thesis is an AI that can be interacted with in a natural way. It should be capable of thought and reason, synthesising solutions to problems, as well as having feelings. Scientists are still far from reaching this near impossible goal but progress is steadily being made towards it, detailed below are not only progress but also what is lacking.

2.3.2 Current Progress

That there is something still missing is obvious, but before any progress can be made there has to be a clear direction in which to proceed. Another quote from the Jargon File gives: "Examples of AI-complete problems are `The Vision Problem', building a system that can see as well as a human, and `The Natural Language Problem', building a system that can understand and speak a natural language as well as a human..."[7].

The Vision and Language Understanding problems, as well as handling any unforeseen consequences that may arise when solving them, are among the biggest problems that need to be solved before an AI-Complete system can be realised, but these are not the only problems. A further example is "Problem Solving", the ability to deconstruct a larger problem into smaller problems and synthesize a solution based on its previous knowledge, much like what Lipson was working on[6; 8].

There's also the problem with "Knowledge Representation and Reasoning", to store, use knowledge and gather new knowledge, like Webcrow and Wallace[8; 21; 23].

Yet another AI-Complete problem is solving CAPTCHAs ("Completely Automatic Public Turing test to tell Computer and Humans Apart")[8; 25]. CAPTCHAs are an implementation of the Turing test and they are used to keep automated programs, or "bots", from pretending to be human, mostly on the internet[25]. A CAPTCHA usually consists of a question and accompanying picture, where the picture contains the answer to the question, but displayed in a complicated manner which is supposed to make it more difficult for a machine to decipher[25].

2.3.3 Related Work in AI-Complete

Most of the previously mentioned still has a long way to go, but as far as the CAPTCHA solvers go, they are created on a case by case basis, entirely dependent on the CAPTCHA at hand, and they are constantly evolving at the same pace as the CAPTCHAs themselves[8]. While there is no CAPTCHA solver that can handle every single case, it is still a step in the right direction and it helps in seeing how far development has come and what is yet to come. Following are a few other examples of work going on in the field of AIC, or otherwise closely related fields.

Human Performance Models have been used for a while and they are tightly related to AI research as they are used to model human behaviour in certain situations[26]. They are usually divided into two disciplines, "Neck Down" and "Neck Up", namely to study the effects on the human mind or on the human body[26]. Researchers are currently trying to combine these two disciplines in order to study both mind and body, which could lead to something closer to an AIC system[26].

Those working with Neck Up are focusing on trying to combine the functions that make up the human mind[27]. They are also thinking about how the two can be put together, but are more focused on the inner workings of the mind. What they are developing is a system capable of interacting with the real world by both thinking and acting upon those thoughts[27].

In relation to the neck up approach, some researchers are looking to neuroscience for a better understanding of how the brain actually works in order to get closer to a true cognitive AI[28]. They suggest that all researchers working with AI should learn some neuroscience, because according to them the difference between the human brain and a computer is one of the greater problems in realising an AI-Complete solution[28]. This is of course no new realisation, but the problem of how to accurately model a human brain with its complex structure of cells and synapses using a computer is still a formidable challenge[28].

Another group of researchers is trying to implement this by taking a "sensorimotor" approach to the problem[29]. When discussing consciousness or cognition, there are generally two types, Access Consciousness and Phenomenal Consciousness. The former is the knowledge about a feeling and when that feeling should be felt while the latter is the ability to actually feel the feeling[29]. For example, a robot may be programmed to feel pain when certain conditions apply, but only when it actually feels the pain can it be said to possess Phenomenal Consciousness[29]. By giving a robot sensors and having it act on the signals it receives from the sensors, the result is Access Consciousness[29]. To achieve the kind of AI-Complete consciousness that is Phenomenal Consciousness, these researches propose the "Sensorimotor Approach", which is to not only focus on the feeling itself, but also on how 'engaged' you are with the feeling and its qualities[29].

Then there is one thought that the key to creating a human-like AI is to make the computer able to feel like a human. One approach to this is presented by a researcher at KTH[5]. In theory, a fully-fledged, AI-Complete system could be achieved by implementing a way for the AI to receive stimulation to the five human senses and to combine these senses with human feelings[5].

However, to properly be able to implement senses and feelings, whether for Access or Phenomenal consciousness or any form of sentience, an important task is to give a definition to the actual senses and feelings. As such, the next part of this thesis will bring up and discuss the five senses and a selection of feelings that will be implemented into IKE.

2.4 The Senses of Artificial Intelligence

Much of what has been mentioned above in terms of ongoing research is regarding cognition and consciousness and this is of course no coincidence. Especially the last example is closely related to this thesis as it describes a way in which senses could be simulated digitally. The system known as IKE developed throughout this thesis possesses consciousness of the Access type. That is, the system will know the feelings and senses and how they relate to an object but it cannot actually experience these in a natural way or really be affected by them the same way a human would. Regardless, an explanation to how the senses could be implemented in AI-Complete systems is given and it is heavily influenced by the KTH researcher's definitions of the senses[5]. Following is a rundown of the five senses and how a digitalised version of them could be constructed in order to resemble those of actual humans. The five senses described below are sight, hearing, taste, smell and touch.

- *Sight* – Arguably the most important of the senses is sight and it accounts for almost 90% of what most humans experience and remember most vividly, something that will surely be reflected in the AI-Complete systems of the future[5]. The memory of sight is a collection of images. In humans, these images are collected by the eyes, but an AI-Complete system would use cameras and image processing to make sense of the images. A human might be capable of looking at an entire scene at once, but for an AI such as IKE the senses must be divided into parts for it to properly know what it is sensing. That is, the picture has to be decoded and analysed to make sense of it. In the case with sight, the logical choice is to decode an image as the depiction, colour or scale and sharpness of the picture. The depiction is the object that is seen, such as what its shape is or its facial features if applicable. Colour or scale indicates what colour the image has or which shade of grey it is if it is in grey scale. Lastly, the sharpness is how sharp or blurry the image is.

- *Hearing* – Second to sight, hearing accounts for the next big way for humans to gather information from the world around us[5]. Rather than the human ear, an AI-Complete system would use a microphone and some sort of decoding device or software. Rather than being stored as audio files, the sounds need to be decoded and categorised. Hearing can be broken down into two main types of sounds, vocalisations and voices. Vocalisations are all arbitrary sounds while voices are all sounds that contain spoken words in natural language, regardless whether the words in question are actually understood by the AI or not. Hearing is a special sense in that the two parts are mutually exclusive; either the system hears an arbitrary noise or it hears a word or several words. It cannot be both at the same time.

- *Taste* – Unlike sight and hearing, taste is a trickier subject as it is depends on both situation and the person doing the tasting. For humans there are many variables that can affect the taste of an object, such as the mental or physical state of the person tasting the object. The taste is also affected by the air that passes through the nose cavity[5]. As an AI cannot become sick, this nuance is lost on AI-Complete systems and the same goes for the nasal cavity which is not actually connected to the mouth in the same way. Instead, the system focuses solely on the actual taste of an object. For this, there are three main factors to consider, the object's temperature, texture and flavour.

- *Smell* – For an AI-Complete system smell is also a tricky subject because for a human it is highly subjective and every human might perceive a any arbitrary smell differently[5]. Therefore the AI-Complete system will put focus on objectively interpreting smells. Objectively, the smell of an object can be broken down into two main factors, olfactions and pheromones. The olfactions signal the presence of substances in the air that humans can detect via smell. An example would be the presence of chemicals. Pheromones on the other hand allow for the differentiation between objects or people, though no focus will be put into distinguishing whether a smell is interpreted as pleasant or unpleasant.

- *Touch* – The last of the human senses is touch and it is special in the way that it has the largest receptors among the senses, as the entire human body acts as a receptor[5]. For an AI-Complete system, this would instead be gathered by specially designed receptors placed strategically across the robot serving as the body of the AI in an efficient way for the system in question. IKE does not possess a body, so instead the sense of touch is entirely simulated. The sense is divided into the three factors, contact, temperature and pressure. Contact is both the size of the object and where it touches the receptors, or how many of them are sending feedback from the touch. Temperature is self-explanatory, as it was in taste and pressure is how hard the object is touching the receptors. This can vary from very light to very hard.

3 Methods

This is a chapter containing most of the methods relevant to theses in general. The choices are made and explained in appropriate detail.

3.1 Possible Methods

First for the list of the potential methods, because of the sheer number of methods only a brief explanation will be given for each of them. After an explanation on the various methods, more in-depth explanations for our choices will be given. The different methods are covered in Research Strategies, Data Collection, Data Analysis, Quality Assurance and System Development Methods.

3.1.1 Research Strategies

A bit less abstract than the methods described early in chapter 1.5, the research strategies can be seen as applied methods. When data acquisition is mentioned in a method, it is most often obtained using one of these ways. Strategies are not mutually exclusive and several can be chosen at the same time, rather than just borrowing ideas from others as with methods. Of course, only the strategies applicable to a thesis of this kind will be covered. The possible strategies are experimental, ex post facto, doing surveys or case studies, action research, exploratory, grounded theory and ethnography. After a short explanation of each of these strategies the choice for this thesis will be given and explained.

- *Experimental* – Resulting in vast amounts of data, this strategy is a means to obtain information regarding variables, their relationships and the aforementioned causes and effects. As the name implies, experimentation is key and a hypothesis to prove or disprove is vital. Variables and their relationships are explored by changing or keeping certain variables, but this research method also identifies dependent and independent variables[12].

- *Ex Post Facto* – Much like experimental research, this is a strategy to prove or disprove hypotheses and provides the relationship between cause and effect between variables. However, identifying whether a variable is dependent or not is not a key point. That is because it is a research/evaluation method applied after experimentation has concluded[12].

- *Surveys* – Surveys are divided into two distinct variations offering different results, the "cross-sectional" and the "longitudinal" surveys. The main difference is that longitudinal surveys are conducted over a longer period of time to see how the results change, while the cross-sectional surveys are only interested in the result at a specific point in time. They are similar in that both provide large amounts of data and provide the ability to examine relationships and frequencies of variables[12].

- *Case Studies* – When the line between phenomenon and context is not apparent, the case study is an empirical alternative strategy to investigate the phenomenon. It investigates the phenomenon in a real life context using multiple sources of evidence. The studies themselves can be based on both quantitative and qualitative evidence[12].

- *Action Research* – This is a systematic and cyclic method where planning, action, observation, evaluation and critical reflection is repeated several times. This is done in order to alleviate problematic situations to improve the way issues and problems are addressed or solved. Because it is usually applied to restricted data sets it is often used in tandem with qualitative methods[12].

- *Exploratory* – Exploring different options the main idea. Usually, this is done through surveys and such and it "provides a basis for general findings by exploring the possibility to obtain as many relationships between different variables as possible". The goal here is not to get definitive answers to problems, but rather to identify the issues and what they depend on. Qualitative data collection is therefore preferable[12].

- *Grounded Theory* – Much like empirical research, this strategy is strictly based on data and the analysing thereof. Largely classified as an inductive method it allows for the development of a theoretical accounting of the topics featured[12].

- *Ethnography* – Stemming from anthropology, the study of human behaviour. This method aims to place phenomena in a social and cultural context. This is achieved by way of relating commonalities between people and cultures under descriptive studies[12].

For this thesis, the best option is grounded theory. Grounded theory revolves around existing data and does not require additional testing to gather more data for a theory. As this thesis is interested in creating a new system from existing research, grounded theory is a natural choice.

3.1.2 Data Collection

These methods above all share a common trait; they all rely heavily on data and the collection thereof. Again, many different schools of thought exist regarding how the data is to be collected. The most commonly used options for data collection are questionnaires, case studies, observations, interviews and gathering by examining language and text. Once more, after an account on the most common and applicable options, a choice and its corresponding explanation will be described.

- *Experiments* – This is done to obtain large sets of data and the way the experiments are done depend on the type of hypothesis to be proven[12].

- *Questionnaire* – A questionnaire is a set of questions given to a group of people and the collected answers are then documented and categorized[12].

- *Case Study* – This method is used with the case study research method where a select group of individuals are studied, giving a smaller set of qualitative data instead of the larger set of quantitative data gathered through the previous two methods[12].

- *Observation* – A method of observing the behaviour of participating individuals or culture groups in the real world[12].

- *Interviews* – Directly conducting interviews with a person or a group of people regarding the subject at hand. This is done in order to gain an understanding for the participants' views and opinions on the subject[12].

- *Language and Text* – Collecting verbal or written information originally gathered by others[12].

The obvious choice here, much for the same reasons as the previous choices made in chapter 1.5, is language/text and interviews. This is mainly due to the qualitative nature of the research to be made in order to gather a solid understanding of the concept of AI and AI-Completeness. It also goes well with the already chosen inductive approach.

3.1.3 Data Analysis

After collecting large amounts of data, the data has to be carefully processed and analysed to get anything useful out of it. Data analytical methods that can be applied are statistics, computational mathematics, coding, analytic induction with grounded theory and narrative analysis. These methods are different ways in which to represent and efficiently put gathered data to use.

- *Statistics* – With statistics, data is divided into different representations, such as graphs, by calculating frequencies. This can be done both for the representation and for evaluation of the data[12].

- *Computational Mathematics* – While it shares the usage of mathematics with statistics, the aim of these calculations is to create models and numerical methods or simulations[12].

- *Coding* – This method is a way to turn qualitative data into quantitative data, done by analysing interviews and observations. The goal is to find commonalities, concepts and strategies for labelling, often used in combination with statistics[12].

- *Analytic Induction and Grounded Theory* – These methods are iterative and they alternate between the collection of data and analysing the data. An iteration ends when no contradictory proofs remain, in other words when the theory has been validated[12].

- *Narrative Analysis* – This is a method used when the data consists of texts and recorded information, such as data collected with the language and text method of information gathering[12].

The choice here will be coding, as mentioned in the data study in chapter 1.5.5, as most of the data collection is conducted via researching published papers. The papers are on the subject of AI and no experiments or such are conducted, the data is the statistically analysed to gain a solid understanding of the concept.

3.1.4 Quality Assurance

Having analysed the data properly, it still has to be validated to make sure it is not incorrectly represented. This is especially true with second-hand data such as data gathered through observation of language/text. Again, a variety of different methods to assure the quality of the data and observations exist. For qualitative research, the most important methods of quality assurance are validity, reliability and replicability. For quantitative research, they are dependability, confirmability and transferability. There is one more quality that has to be assured for both qualitative and quantitative research and that is ethics. Approaches to quality assurance include validity, reliability, replicability, ethics, dependability, confirmability and transferability. No choice for a specific approach will be made in this case, as all of the ones for qualitative research have to be met. These are the available methods.

- *Validity* – For quantitative research this means making sure that instruments used are accurate and do not corrupt the results. When qualitative research is concerned, it means that the research is conducted correctly, following a set of rules with results that can be readily confirmed by participants[12].

- *Reliability* – The measurements have to be consistent for every test, as such this is to ensure that the data does not vary too much. This is mainly a concern when dealing with the large amounts of data in quantitative research[12].

- *Replicability* – The experiment and its results must be achievable by another scientist in similar circumstances. The results have to be reliable and consistent in the sense that they cannot be a one-time occurrence. This does not only require carefully performed experiments, but also well-documented proceedings. Again, this matters mostly in the case of quantitative research[12].

- *Ethics* – This entails the physical and mental protection of both participants and researchers. That is to protect the privacy and integrity of involved individuals unless integral to the research. It also includes things such as getting participants' consent in written form and treating all material with proper confidentiality. This applies to all kinds of research[12].

- *Dependability* – The qualitative counterpart to reliability where results are ensured to be consistent. It is the process of confirming the correctness of the conclusions through auditing[12].

- *Confirmability* – Again mostly concerning qualitative research, this confirms that personal assessments which could influence the results have been dissociated from the tests. Data collected is free from any personal opinions or preconceptions[12].

- *Transferability* – Means that the resulting data can be used in future research done by other researchers. It is not very relevant to quantitative research[12].

As this thesis focuses on quality over quantity, only the quality assurances of validity, dependability, confirmability, transferability and ethics need to be confirmed. Below is a short account on how these are to be confirmed.

To address the requirement of validity, the work conducted attempts to adhere to the requirements set for the entire project. This means that all work that is done is done so in order to fulfil one of the goals in order to be easily confirmable.

The dependability of the project will be ensured by unrelated testers to find whether or not it does what it claims to do.

Ascertaining the confirmability of the project can be done by making sure the project properly follows the requirements set out for it.

The transferability of the project is assured in by way of keeping details at a fairly general level without delving too much into specifics that make the results inapplicable for future projects.

3.1.5 System Development Methods

Once choices have been made about the research and its methods, the next important part is the actual development of a product. There are several viable methods for the development of a product, but only the methods applicable for this thesis will be given and described. The applicable methods are prototyping, waterfall and agile.

- *Prototyping* – Prototyping is often divided into four phases with several steps in each phase. The first phase is the planning of the project to ensure what needs to be done and how. Following this, the second phase is about creating a specification for the prototype. The third phase is the phase of actual prototyping, where work is conducted in accordance with the plan and within the constraints of the defined specifications. The last phase is where the results are analysed with the intent of making sure the prototype fulfils the specifications[30].

The strength with prototyping lies in that it is easy to learn and use. Prototyping is also flexible in the sense that certain steps can be compromised for the benefit of other steps. If time is of the essence a prototype of less fidelity can still be produced to ensure it is completed within its timeframe, after which it can be adjusted later. The key questions are: "Will the design work properly?", "Can the design be produced economically?", "How will users respond to the design?" and "Which approach can be taken to get from concept to product?"[30].

- *Waterfall* – This is the traditional development method. Like with prototyping, tasks are divided into different steps. Requirements are gathered, a design is made and then the designed program is made. The program then goes through quality assurance and if the requirements are satisfied the process moves on to deployment. Every step is separate from the other steps and only when a step is deemed complete can the process move on to the next step. One big problem with the waterfall model is that each step could be done by completely different teams at different times, so when one team is done with their part on the project they start working on their next project and are no longer available for the project at hand. While it is an effective method in terms of human resource management, the resulting program can suffer from unexpected problems[31].

- *Agile* – While there might not be any 'silver bullet' that solves all problems in software development and increases productivity ten-fold, but agile development is at least a silver alloy. Agile development works in cycles where each cycle is a condensed version of the waterfall method. This makes for more flexible development where bugs and problems are encountered earlier and can be resolved before the full product is deemed complete. These cycles can vary in length, but the key point is that they are repeated time and time again until the product is finished[32].

Prototyping was chosen due to the relative ease of both learning and using, as well as the structure lending itself well to developing a prototype when the aim is not necessarily to fully implement the system. The repeating model of agile development is seen as great, but for the small scale of this thesis, the repeated cycles will more likely hamper the progress than promote it.

4 Prototyping

In this chapter, the prototyping method and how it was applied for this project is explored. There are entire books dedicated to detailing the prototyping method, as such only the necessary explanations are given here. Following there is an explanation on how it has been applied, including each phase with subsequent steps. A list of the major problems encountered and their solutions is also included.

As already mentioned, the prototyping method is one of the easiest methods of development to learn and use. This is largely due to the way it is divided into separate phases, each with a clear goal and purpose. The four phases are planning, specifications, design and results.

4.1 Phase I – Plan

The first phase is where investigation takes place to ensure the project's sustainability and possibility to be completed. A big part is also making sure that subsequent phases go as smoothly as possible. In essence, planning is made to ensure a smooth workflow. For a solid plan leading up to a better product, steps recommended are verifying the requirements, developing task flows and defining the content and fidelity.

4.1.1 Step 1 – Verify Requirements

When a product or prototype is to be made, as discussed in the first chapter, for a project to be sustainable there has to be a need for the product, which means that there are certain requirements that the product has to fulfil. But the purpose of the product is only the business requirement. There are three other distinct types of requirements for any one prototype, the functional, the technical and the usability requirements. All four types of requirements have to be assessed before development can start[30].

Once assessed and asserted, the requirements have to be listed in order to use them later as a gauge of progress towards the completion of the prototype. Besides listing them, it is also important to prioritise and make sure each item on the list appear in order of importance to the system as a whole[30].

4.1.2 Step 2 – Develop Task Flows

The second step is to develop a task flow. But before that, an actual list of tasks based on the previously compiled list of requirements has to be made. Once this task list has been drafted, the order in which tasks have to be completed, the dependencies, have to be identified[30].

Once all dependencies have been identified, the tasks once again have to be organised in accordance with the dependencies. At this point, some dependencies may prove to be redundant, in which case they should be removed[30].

4.1.3 Step 3 – Define Content and Fidelity

The final step of the first phase requires an understanding of fidelity. In this case, the fidelity of the product or task is how detailed it is. For example, a high fidelity prototype is very close to the finished product while a low fidelity prototype might be just a very rough sketch or model of the product. Obviously, there are both pros and cons for each. A high fidelity prototype is very time-consuming but also very detailed, but the low fidelity version is much faster and cheaper to produce[30].

To make an informed decision, the focus and objective of the project have to be clearly defined. Certain objects benefit more from a higher level of fidelity than others. This means to put more emphasis on what is important for the prototype and de-emphasis on what is not[30].

4.2 Phase II – Specification

The second phase is mainly concerned with defining the specifications from which the prototype and the finished product are to be built. Again, this preparatory phase includes three steps, determining the characteristics, choosing a method and choosing a tool[30].

4.2.1 Step 4 – Determine Characteristics

The characteristics specify what kind of prototype is to be developed and the important characteristics to determine are audience, stage, longevity, expression, style, medium and fidelity. When determining these characteristics, choosing a smaller number to focus on is preferred[30].

Audience is one of the most important characteristics. What is determined here is who will be using the prototype. The two kinds of audiences are internal and external, whether it will be used by a member of the development team or by a client or actual end user[30].

The next step to consider is at what stage the prototype will be shipped to the audience, as it is still a prototype and not the finished product. It could be shipped at an early stage as a proof of concept or perhaps when it is nearing completion[30].

The longevity of the prototype concerns how long it is expected to last[30].

As for expression, it is whether the prototype is conceptual or experiential. That is, how close the prototype is to the final finished product as far as user experience is concerned[30].

Style determines how the prototype is interacted with. It could be a narrative experience which shows how the product will be used or it could be a prototype which works as a demo for the user to try out[30].

Next, the medium for implementing the prototype are primarily chosen from either physical or digital and, depending on the type of prototype, this choice might limit the available choices for style[30].

4.2.2 Step 5 – Choose a Method

Choosing a prototyping method is one of the most important parts so far. The available methods are card sorting, wireframe, storyboard, paper, digital, blank model, video, Wizard-of-Oz and coded prototyping[30].

Card sorting is an abstract method where several developers use cards and to decide on the best composition of product and prototype early on in the development. The cards themselves may contain functions or requirements and are sorted into a kind of prototype[30].

Wireframe prototyping is, like card sorting, a narrative prototyping method. The idea is to make a high-level or abstract sketch as a proof of concept or a very conceptual prototype[30].

Another narrative method, storyboard prototyping is about visualising the entire prototype as either a kind of story or a user interaction scenario with the desired outcome of the project at hand[30].

The first interactive method mentioned is paper prototyping in which a sketch of the imagined interface with all the buttons and responses sketched out on different sheets of paper[30].

Digital prototyping can be seen as a digitalised version of the paper method, except it can also be used as a narrative method in the form of a slideshow[30].

A quick method to craft a low-fidelity physical prototype of the finished product is the blank model prototyping method which makes use of basic materials to show the shape of the finished product or prototype[30].

Yet another narrative prototyping method is the video prototyping method, which is like storyboard prototyping with a user interaction scenario, except in video form[30].

Wizard-of-Oz prototyping is good for situations when developing implementations of difficult problems such as those labelled AI-Complete. In practice, a "man behind the curtain", another developer acts as the finished prototype. The prototype is more of a narrative, digital, semi-interactive slideshow. The man behind the curtain interprets any input from the user during testing and acts as the program by showing the proper results[30].

While most of these methods are best used early on in the development of a product, coded prototyping is best applied at a later stage, as it is basically developing a barebones version of the prototype or sometimes even the finished product[30].

4.2.3 Step 6 – Choose a Tool

Having selected a method, a tool applicable to creating the prototype has to be chosen[30]. First the method has to be mapped to the available tools.
The tools available depend heavily on the type of prototype that is to be created and even the type is limited by the method chosen. For example, the blank model method relies on physical tools whereas other methods, such as coded prototyping, requires digital tools[30].

Another important part in this step is to determine the timing. This is closely tied to the longevity characteristic, but it is not limited to only determining how long the product will last. Also of concern is the amount of time it will take to create a prototype. Certain tools might take a long time to set it up but speeds up the actual development process and these are important factors to consider[30].

Once an estimated timing has been determined and the available tools have been mapped to the characteristics, the main task remains of selecting the tools for the team to use[30].

4.3 Phase III – Design

The third phase may have fewer steps than the other phases but it is still where most of the project is spent. Now that the requirements and tools have all been identified, the actual work can begin. The two steps of this phase are establishing the design criteria and creating the design.

4.3.1 Step 7 – Select a Design Criteria

There are many parts included in the design of a product. Among these, there is the interface and the visual design. The visual design is the look of a physical or digital product, whereas the design of the interface might be how buttons are laid out on a website or how a program's output is formatted. As with the other steps of prototyping, guidelines or thoughts on what is most important for these two have to be considered and decided. Some guidelines for the visual design can be information flow or balance; a badly planned information flow will make the product much harder to understand and more frustrating to use. The balance of the visual design would be how well thought-out the design is. For the user interface, some options are to consider efficiency and learnability. Essentially, the efficiency is how effectively a user can navigate the program, while learnability is somewhat related to information flow as in how easily the user can take part of information to make using the program easier[30].

4.3.2 Step 8 – Create the Design

The eighth step is where the actual work towards creating the prototype is done. Implementing the previously created list of tasks requirements in the way specified by the subsequent phases, using the tools picked out is what results in the prototype[30].

During the work process, documenting what is done and why is very important. It is also important to keep in mind the goals, requirements and rationale discussed previously and always make sure that no unnecessary work is done to ensure that time and resources do not go to waste[30].

4.4 Phase IV – Results

With a prototype constructed, it has to be validated. That is, to verify that all requirements are met in a satisfactory manner. The last phase to validate and potentially launch the prototype is divided into three steps, reviewing, validating and deploying the design.

4.4.1 Step 9 – Review the Design

Now that there is a prototype, the intended audience comes into play. The audience should be given an explanation on what to expect at this stage in the development of the prototype[30].

It is also important to consider how the prototype is presented to the audience and it should be carefully planned. After gathering feedback from an initial internal presentation a road forward is planned, whether it be continuing to the next step of validation or starting a new development iteration[30].

Sharing with the audience the rationale behind decisions is important to avoid any misconceptions about the prototype which in turn could lead to less useful feedback[30].

4.4.2 Step 10 – Validate the Design

After the feedback from an internal review, the external audience is called in to test the prototype and give more feedback[30].

If there is a lot of constructive feedback on certain features of the prototype, these features will likely come out much better after the next iteration, while features with less feedback possibly will not change much at all. Of course, this can change from iteration to iteration, which is also part of the reason why the prototyping method is employed in the first place[30].

4.4.3 Step 11 – Deploy the Design

Depending on the consensus of the feedback, the deployment of the prototype will be either developing it into the final product or setting it up for another iteration[30].

In the event that the prototype is handed off to a different developer for further development, it is of much importance that the prototype is well-documented[30].

The two main formats of distribution at this stage is either open and closed. An open format allows for the developer to develop further after distribution, while a closed format sees the prototype finalised. In the latter case, it is still possible to develop it further if it is needed, but without changing the core. However, at that point any further development is not necessarily done by the original developer[30].

Lastly, as the entire development process needs to be documented, this obviously holds true for the prototyping results as well. This is important as it will serve as reference for any further development[30].

After this final step of the prototyping method, the developer's work on this particular project is finished.

5 Developing IKE

After the description of the prototyping process is a record of the creation of the prototype; the system IKE, which stands for Intelligent Kernel of Emotions. IKE is as already mentioned designed to be a small step towards achieving the dream of an AI-Complete system.

5.1 Phase I

Following this is the documentation of the work done in each phase of the entire development process. First is the phase with the three steps regarding requirements, planning of tasks and task flow as well as the definition of the final product's fidelity.

5.1.1 Verifying Requirements

In order to verify and order the requirements they first need to be collected. The obvious choice for this is to have a meeting with the employer for the product of this thesis, a Swedish researcher within the field of AI at KTH. Several requirements were agreed on when the project was accepted, but for the sake of clarification the meeting is held to make sure there are no misunderstandings between the two parties.

The meeting is held on the Kista campus of KTH in Stockholm, Sweden on 31st January 2015 with the employer who requested IKE to be developed and includes all personnel dedicated to developing the system. The meeting is not particularly well-structured and serves the purpose of bouncing ideas more than being a formal interview. During this meeting, most of the already agreed upon requirements are then further explained and clarified. As the meeting concludes, the six main requirements agreed upon are the following:

- IKE is to have a memory which can be extended in a near-unlimited manner for as many objects as it encounters during its life span.
- IKE is to have feelings which can be added to and which can get as complicated as necessary.
- IKE is to have all five human senses of sight, hearing, smell, taste and touch implemented.

- IKE is to have several versions of each object it remembers, one for each experience with the object as well as a separate abstracted instance of the object.
- IKE is to be able to reply when queried about an object, regardless whether it has details about the object or not, as well as any feelings and senses IKE knows about the object.
- IKE is to be able to reply with first an abstraction and then to be able to pull out more specific information should it be requested.

These were the six main requirements and they can all be deconstructed into several smaller requirements which can then be ordered, but at least for the first three requirements they are mentioned in order of importance. The memory was the most important, followed by feelings and then the senses. For the last three requirements, it becomes apparent that there has to be a number of objects "hardcoded" into the system in order to properly test the functionality of the memory. The decision here is that three types of objects would be implemented and defined, balls, clocks and sweets. These are chosen because of their relevance to each of the human senses. Balls have a very familiar look and touch, clocks have a very memorable sound and sweets are relevant to the senses of taste and smell.

5.1.2 Developing Task Flows

With the requirements in place, work begins on developing a task flow. While the most important part to develop is the memory, the order of the dependencies is the opposite. After all, there is no use for memory without anything to remember. As such, the dependencies for developing the memory are objects to memorise, which in turn require that the senses and feelings are in place.

With these dependencies in mind, the tasks are to first implement the senses and feelings and make sure they work as expected. Second is to add the pre-defined objects and map them to their associated senses and feelings. Left for last are the tasks about creating, structuring and editing the memory.

Again, as with the requirements the task flow contains more detail than what is written here.

5.1.3 Defining Content and Fidelity

Defining the fidelity of the product is the last step of the first phase. Early on in development it is decided that there would be two deliverables from this thesis, the first being the model represented as a blueprint with a detailed explanation and the second being an implementation of the system in order to properly test it and make sure it works as intended.

The fidelity of the implementation only needs to be of a good enough quality to be able to confirm that it works, while the model needs to be of a higher fidelity in order to properly explain it.

5.2 Phase II

As written earlier in chapter 4.1.2, the first step of the second phase is to determine the characteristics of the project. There are several options for each characteristic and they all have to be considered as they affect the end result in different ways. There are both pros and cons to each choice, depending on the project at hand.

For this thesis the audience was the employer and other scientists around the world researching AI-Completeness. Since the team working on the project are only two, this means that the internal audience is negligible.
The second characteristic, the point in development when IKE is to be unveiled is decided to be a small first review when the design is done and then later a larger release once the implementation is finished and a complete system can be presented.

As for the third characteristic, longevity is a harder choice seeing as IKE will have to last until it can be succeeded by an AI-Complete system or at least by a newer version, though at the time of writing no plans for further development are being made.

Then expression is considered. In the case with IKE, it is to be a mostly conceptual design in the form of a model of the system accompanied by a naive implementation to display its functionality. IKE as a system is a conceptual prototype, but the implementation is an experimental addition.

One of the last choices to be made is style. The style of the system is narrative, but the auxiliary implementation is interactive. The main point of this thesis being the system IKE, the project as a whole is narrative.

The second last choice is the medium in which the system is to be implemented. As previously mentioned, IKE is to be released as a written description accompanied by an implementation of the system. The implementation is chosen to be presented as a command line interface in the chosen language with only a view for issuing commands and receiving responses.

The last characteristic, fidelity was already discussed and is decided to be high for the system design but lower for the implementation.

5.2.1 Choosing a Method

The next very important choice is the choice of a method. Previously several possible methods were discussed. As IKE is to be delivered both as a design and an implementation the choices are not few, but the best compromise is wireframe prototyping, a narrative method. This is a good choice as it is used to create conceptual high level sketches of a system, which works perfectly in IKE's case.

As for the implementation, the secondary method of coding is chosen. It is not really as much a method as it is simply filling in the blanks with code.

5.2.2 Choosing a Tool

As far as tools are concerned, there are several options but they have little impact on the prototype. Some options are to use graphical programs such as Adobe Photoshop or Microsoft Visio, but instead the both time-efficient and inexpensive method of using pen and paper is chosen.

However, for the implementation of IKE the choices are more limited and far more important. As far as programming is concerned there are two main schools, imperative and functional languages. There is also a third, the logical languages. Imperative languages include the classic languages such as Java and C++, whereas some functional languages are Erlang and Haskell. The third option, the logical languages really only consists of one well-known language, Prolog.

The choice to use Prolog for this project is because it is especially well suited for projects relating to AI. One important feature it has is backtracking, where options are tested and then re-tested until a match is found or all options are proven false. Another is the pattern-matching capabilities within tuples and lists which come in handy in the development of the memory.

Most, if not all of this can be implemented in Java or Haskell, but Prolog comes with these features as a main part of the language without the need to develop advanced features to perform the same tasks.

5.3 Phase III

With most of the preparatory work done, the biggest phase begins. The third phase contains the actual development work. But first there is still one small step left before the work can begin.

5.3.1 Establish Design criteria

To proceed with the work, design criteria have to be established. However, this has little relevance to IKE, as the design criteria are mostly relevant to prototypes with actual graphical user interfaces. Due to IKE being made primarily for its functionality, no real design criteria are established. Regardless, some thought goes into the design of IKE's input and output. IKE's UI at this stage is as previously mentioned decided to be just a command line prompt for sending commands and receiving responses. Due to the functionality provided by Prolog, IKE's answers are not definite but can be changed should they prove unsatisfactory. As such, the only design decision left at this point is how output should look, whether it should be given as comprehensible sentences or simply as a list of relevant data. The decision is a hybrid mix of the two, at juncture points in the query text will be explaining what is happening, but the actual returned data will be output as a list of facts as soon as they have been identified.

5.3.2 The Work

The working step begins with first defining what information has to be gathered for the senses to process, along with a specification for the senses given by the employer. However, the specification was lacking some information which has to be complemented. It is also made apparent that some of the information specified only really makes sense if IKE can gather the information on its own, but more on this in the next chapter.

The designing of the system is not much of a challenge, but the implementation instead proves to be more of a challenge. This is largely due to the high level the design started at, looking more like a mind map than anything else. As work continues, the design becomes increasingly detailed, revealing more features, how they work and how they are to be implemented.

In addition, most of the problems with the design do not become apparent until they are being implemented, which of course is to be expected. In fact, one of the reasons to create the implementation was to highlight errors that the higher-level conceptual design missed.

On the second iteration of prototyping, coding on the implementation begins with a better plan and a reviewed design. With the design well planned and almost made in pseudo code, implementing it was more or less a matter of filling in the blanks.

One major feature that proves itself difficult to implement properly is the object handler. The memory structure is decided, but the order to search for objects is not. The solution here is to use a breadth first algorithm, meaning that the system looks at all main types first before going the lists of subtype. This increases the speed of the program once a greater amount of subtypes are added.

Another problem that makes itself apparent near the program's completion is how to properly handle backtracking. More specifically, this is how to list all occurrences of an object once it has been found. This required a minor restructuring of the memory and the list handler, but it is a quick fix.
Apart from these two problems, no other important problems surfaced and IKE could be completed efficiently according to plan.

5.4 Phase IV

As previously detailed, there were to be two releases. The first release when the design of IKE is finished and then again when the implementation is completed and the IKE system is complemented to answer any criticism received during the first showing of IKE.

5.4.1 Review the Design

Before showing IKE for the first time a final check is made internally to try and identify any potential blunders. No major issues arise before the first release, much due to the fact that a high level design leaves little room for visible bugs, as it is mainly a concept.

But when it comes to the second, final, release of both design and implementation, the implementation is tested thoroughly by the developers. Again, no major bugs that had not already been discovered and addressed are found. This is partly due to the small scale of the memory present, as the system becomes more complex as the size of the memory increases.

5.4.2 Validate the design

The first public showing of IKE's design is held on 15 April 2015 at the Kista campus of KTH with the employer and the two developers of IKE, the same people present at the first meeting. The meeting this time is also conducted in a rather arbitrary manner and is mostly improvised. A discussion is held about the choices behind the design and where how development is to proceed.

The name of the system is discussed. At this point in time, the name is yet to be set in stone but it is tentatively given the name IKE during this meeting.

The main idea with this phase of validation is to garner feedback from the employer and to make sure the project is headed in the right direction before the next big iteration.

After the next big iteration is the real release, but before IKE is released to the public, two testers get their hands on the implementation to see if it can be improved further. One tester has followed the development of the system and has read this thesis as it has come together, while the other is given the implementation with minimal information. The testing is conducted in a somewhat special way. The first tester is sent all files and documentation required to test the program via email and is given free rein to test the program.

The second tester is not even aware of what AI-Complete is. This tester was actively chosen in order to get an opinion from someone completely uninvolved. This test is conducted in the home of one of the developers with both developers and the tester present.

Neither of the testers return with any notable criticism, either by way of bugs or ways to improve the system, as both of them simply confirm that the program is doing what it is supposed to do.

5.4.3 Deploy the Design

As detailed in the first part of this chapter, the last step of the prototyping process is to decide whether or not to enter another iteration of prototyping or to release the prototype and call it finished. As already mentioned, there were two major iterations of prototyping for this project.

The first iteration was already decided from the beginning that it would be repeated, though the second time an implementation of the system was to be made while fixing potential errors in the design.

By the end of the second iteration and a brief final review of the system, it is deemed ready and handed over to the employer.

6 IKE

As already mentioned, the system was named IKE, the Intelligent Kernel of Emotions. The main reason for this being that the whole point with this project was to create an AI with the ability to possess emotions and senses, how they are connected rather than how they affect the actions of the AI (read chapter 1.7).

In any case, the fifth chapter of this thesis IKE will be further explained in detail. First is a description of IKE and its feelings and senses. Following this, there is an explanation of the functionality these feelings and senses result in. Again, for the feelings to actually exist they have to be bound to memories, which in the case of IKE are represented as objects, but they might as well be occasions or specific points in time. There will also be a short explanation of the hardcoded objects chosen and their details, which will lead to a written example of how the interaction with IKE is supposed to work.

6.1 Feelings

Generally speaking, emotions are highly subjective. Something that is interpreted as absolutely horrifying to one person might instead be pleasurable to another. This introduces controversy when programming an AI such as IKE. IKE currently has no way to define its personality and draw its own conclusions, so far all feelings are associated with objects chosen arbitrarily based on personal experiences.

6.1.1 The Feelings

Regardless, the current version of IKE possesses the feelings it is supposed to possess, those being close to the same feelings a human being would possess. The way IKE is implemented, adding or removing feelings is easy. There is however no additional system functionality added by way of feelings. IKE's feelings do not currently affect IKE's judgement, but again this was not planned as a feature in the first place

6.2 Senses

While feelings are subjective and depend on the person, the senses are instead normally consistent for every person; most humans are born with all five of the senses. This is where the previously briefly mentioned senses, sight, hearing, smell, taste and touch are given more detail. Many of IKE's variables are currently placeholders added for the sake of functionality and usability. The final form of IKE is not meant to be interacted with in the way it is now. Instead, IKE is supposed to be in the backend where all information stored in the memory will be utilised by functions in the frontend. Basically, IKE is a database with a framework for storing, combining and handling the senses. IKE's true functionality will be added as parts of other programs that utilise this information, as well as through the sensors gathering this information.

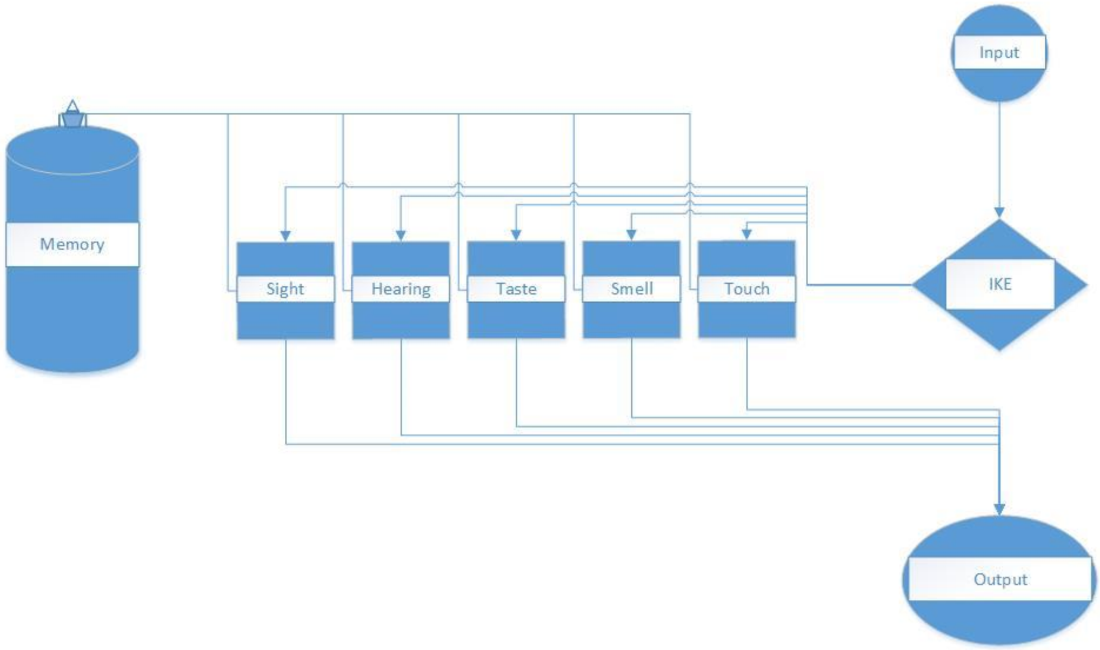

Figure 1 - The architecture behind IKE

6.2.1 Sight

As discussed in chapter 2, sight is the sense humans use to perceive most of the world around them. IKE's sight is implemented with the future in mind. The variables IKE stores are depiction, colour/shade and sharpness. Depiction is currently stored as a text string of whatever shape the object has, such as *round* or *square*. Colour or shade are also stored as text strings describing the colour, while sharpness is an integer value between 0 and 100 representing a percentage of how blurry or sharp the image is.

6.2.2 Hearing

For hearing, the second largest sense, the important part is to differentiate between noises and spoken words. IKE knows these as vocalisations and voices. Vocalisations are noises from any sources whereas voices are sounds with actual words in human language. Vocalisation is currently stored as a variable best representing the sound or onomatopoeia the object makes if any at all, for example the text string *rrrrrrrr* describing the sound of an alarm clock. If it is a voice variable it is instead stored as a string containing the spoken piece of dialogue.

6.2.3 Smell

Smell is one of the smaller senses. For IKE, smell is divided into three variables, olfactions, pheromones and intensity. Olfactions are the presence of something detectable via smell, pheromones are how the detected substance actually smells and the intensity is how strong it is. The olfaction variable is stored as a Boolean value, yes or no, showing whether the object can be detected by smell or not. The pheromones variable however is currently stored as a string best describing how it smells, for example *cinnamon* for a cinnamon bun. In the future it will be replaced by an alphanumerical code representing the substances that are detected and is generated by whatever sensor is collecting the smell. Intensity is much like sharpness for sight, shown by a scale of 0 to 100%.

6.2.4 Taste

Taste is stored with three variables, temperature, texture and flavour. Temperature and texture speaks for themselves while flavour currently describes the object's objective taste. The temperature variable is stored as an integer value in degrees Celsius. Texture and flavour are currently stored as strings describing the features of the object. These will be replaced with code values like in the sense of smell once proper sensors are in place.

6.2.5 Touch

The last sense, touch has five variables to keep track of how touches occur. The variables are contact, temperature, pressure, texture and phase. Contact is with what part IKE comes into contact with a particular item. The temperature is once again self-explanatory. Pressure is how hard the object is pressed against the contact area. As with a lot of the variables above, the current values for the variables are placeholders and will be replaced with data from the sensors. Currently however contact is stored as a string with the body part that is in contact with the object, temperature is an integer for the temperature in degrees Celsius and pressure is an integer for the pressure stored in the SI unit Pascal. Texture and phase are again simple string representations.

6.3 Kernel

The human's senses and feelings are all linked together by the neural network in the brain and the spine. In contrast, IKE as a computer program does not have an actual body. Instead, the code that IKE runs off is stored in the cloud. All of its senses and feelings are also stored in the cloud inside a specifically designated memory file where the data is waiting to be accessed or edited depending on the needs of the user. However, an end user might not have much need for IKE in its current implementation, as it is nowhere near a full AI-Complete system and still needs to go through many iterations of development on different types of functions before it will have any real functionality as an AI, but more on that later. What is important here is the actual memory itself. The memory makes up the core structure of IKE and is, in a word, its kernel. The memories all contain their own associated feelings and IKE, an AI with its intelligence focused on feelings, is an Intelligent Kernel of Emotions.

6.3.1 Memory

IKE's kernel, its memory, is its most important part, but it is actually only implemented as a regular text file without any sort of protection or encryption. It is a readable and writeable text file which anyone with access to the file could open and edit at will. Apart from the raw facts about stored objects, the memory also contains a part where it stores types of objects, with the aim to create a higher level of abstraction concerning objects.

IKE may have experience with many different types of balls, such as golf balls or footballs, and with it IKE tries to extract what a ball actually is to reach that higher level of abstraction. When questioned about an object, IKE will first reply with an abstraction of the object, but if inquired further it will return specific instances of known balls.

The actual memory is structured as a large list containing all objects. In that list there is another list containing all types of objects, in which there is yet another list for every specific object of that type. Each specific object is in turn stored as a list of tuples for the senses and another list for the feelings associated with the object, which means that the actual feelings and senses are stored at a list depth of four.

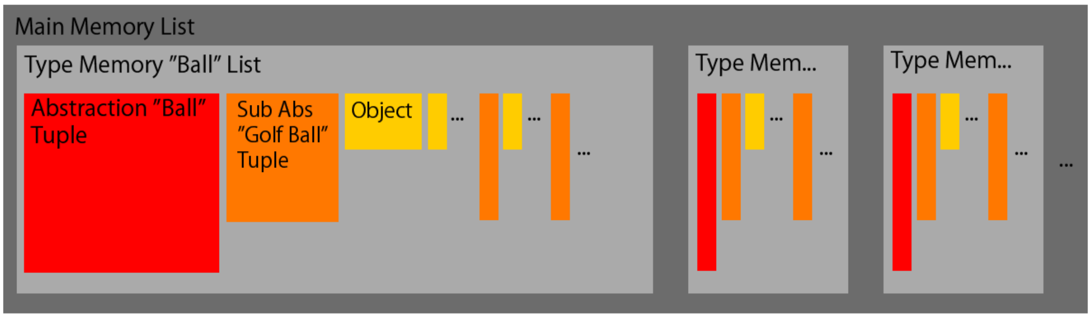

Figure 2 - Memory Architecture

Apart from tuples for each sense, any specific object's list also contains its subtype. Every element in the sense tuples can be specified as either a variable with the name of that particular sense's parts, for example the variable *Intensity*, or as a special notation, *N/a*, for a value that makes no difference to the object in question. These two elements signify that there is a value that is of interest or that that there is no valuable data respectively.

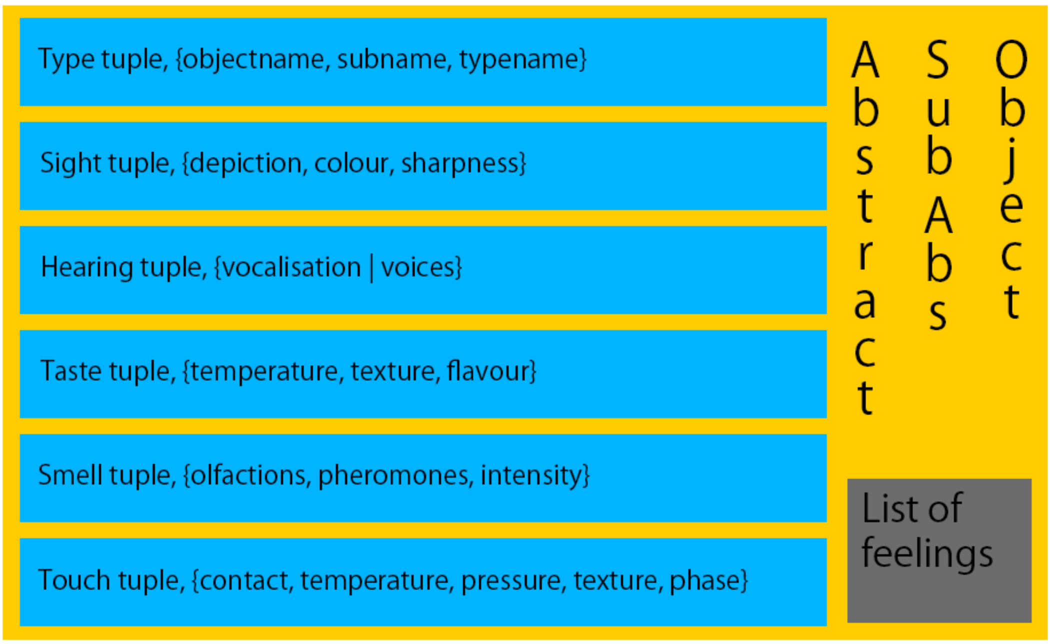

Figure 3 - Object and abstraction architecture

To give the idea more context, *N/a* would be used when the data is not relevant for the specific object, for example a conceptual idea might be seen as an image, but it would not necessarily have to have a level of sharpness as it was not originally an actual image. The variable element is instead used to signify that the element has an importance.

The program will primarily try to find the highest level of abstraction of whatever object that it is asked about. In order to do this, IKE checks all subtypes and looks for common elements in the tuples for each of the senses. For example, if every ball that IKE remembers is round then that means that all balls IKE knows are round and it can then conclude that round things have a high likelihood of being balls. As such, the abstraction of a ball will be an object that looks round and has all other common properties of stored types of balls.

6.4 Functionality

Its memory, feelings and senses are the main part of IKE. The aim here is to describe what this functionality is actually good for. As explained in the first chapter, the goal with IKE is to further research, but also to help robots and AI facilitate emotions. This is in turn to give the AI empathy which could be helpful in a variety of ways. For IKE there will be a few predetermined objects bound to certain feelings as described below. IKE's main functionality in its current iteration is to be able to reply to queries about objects or specific emotions. For example, it can be asked what makes it feel happy, or how a specific object makes IKE feel. The long term goal for IKE is to be able to facilitate and actually act on its feelings, as well as actually experiencing them in the sense of having a phenomenal consciousness.

6.4.1 Predetermined objects

The first three objects that IKE is hard-coded with are clocks, balls and cinnamon buns. These may seem arbitrarily, which they are, but they were chosen for their specific qualities and the feelings bound to them in order to make sure that each sense would be well represented in at least one object. Though as already mentioned, each time IKE interacts with or recognizes an object, a new instance of that object is stored in the memory, meaning that there each object has to have its own subtypes.

- *Clock* – The first object chosen to be implemented in IKE is the clock. The primary subtypes here are alarm clocks and wristwatches. These clocks were chosen for their easily recognisable sound and look, as well as the antagonistic quality of the alarm clocks, interrupting your sleep and waking you up again. On the other hand, wristwatches were chosen mainly for the sake of ensuring that the subtype system works as intended. When thinking of a wristwatch, the antagonistic quality of the alarm clock isn't something that usually pops up, even though the functionality is more or less the same between the two, the ability to tell time. At least for the writers of this thesis, the alarm clock invokes more negative emotions than the wristwatch, which is instead seen as a rather useful tool.

- *Balls* – The second type of object chosen for IKE is the iconic ball. This is an object which has many different values for its various properties. For the ball, there are many more subtypes than the other objects, though with less important senses bound to them. The subtypes implemented here are basketballs, footballs, beach balls, golf balls and rubber balls. These balls differ in materials. The football is made of leather, the beach ball of plastic and the golf ball of very hard rubber. The texture of each ball is different as well, and even within each subtype of ball there might be additional differences depending on how inflated the ball is, if it is broken, wet or anything else that might be different.

- *Cinnamon bun* – Cinnamon buns, which are actually a subtype of sweet rolls, are of course chosen mainly for the sense of taste and smell, as well as the warm and happy feelings that they bring. Subtypes of the cinnamon bun are freshly baked, stale and cold. Of course, these different subtypes come with different feelings, tastes and smells.

6.4.2 Abstraction

As already mentioned, the purpose of having types and subtypes is for IKE to be able to come up with an abstraction for a given object. The abstraction of a subtype contains the elements common between the senses and the feelings bound to the subtype, which it will then extended upwards to the main type. That is, the subtype golf ball has an abstraction based on all of IKE's experiences with golf balls. This abstraction then serves to help create the main abstraction of the object ball. This is where the variable names might be left in for the instances where conflicting memory details appear. This is because certain objects have features that are not necessarily bound to the actual object in question. For example, a ball might feel wet because it has been dropped in water, not because of the fact that it is a ball. In this case, the abstraction will show that the texture of a ball is inconclusive, because not every ball IKE has experienced has had the same texture.

Detailed below is a description of what a typical interaction with IKE could look like for the purpose of making it more obvious what an abstraction means for IKE and in what way these can be used. IKE will first aim to respond with the abstraction of an object, but if that fails it will get more specific and try to look for the subtype that the user wants to know about. If the user still isn't satisfied with IKE's answer or if it fails to find the requested object, IKE is designed to list all the experiences it has had with the different subtypes of the object. The idea is that, when IKE is queried about an object, a specific instance of the object is not necessarily what the user wants, which is why IKE strives to find the most specific abstraction rather than the exact object. Of course, given the limited range of objects currently implemented this may not make much sense, but when more and more objects are added it will become a hassle to keep track of every specific instance. The more information IKE has about an object, the closer IKE can get to a specific abstraction or a *true form* of an object. For instance, a ball is round and its other qualities only actually matter when considering what type of ball it is. The only fact shared by all balls is that they are round, spherical objects.

6.4.3 Sample output/input

Below is a screenshot describing a typical interaction with IKE, detailing most of the functionality of the system. Included is the ability to search for a main object type, a subtype within each object type, to request a specific property about any object. This is also used to demonstrate the backtracking feature.

```
?- ike(ball).
{ball,ball,ball}
{round,na,na}
{vocalization|voices}
{temperature,texture,flavour}
{olfactions,pheromones,intensity}
{contact,temperature,pressure,texture,phase}
[happy,sad,confused,amazed,warm,good,fun,stressed,mentallychallenged,dead]
true .

?- ike(alarmclock).
{alarmclock,alarmclock,clock}
{square,yellow,10}
{rrrrrrrr}
{18,metal,metal}
{no,na,0}
{hand,18,hard,metal,solid}
[tired,sad,angry,confused]
true ;
{alarmclock1,alarmclock,clock}
{square,red,15}
{rrrrrrrr}
{18,metal,metal}
{no,na,0}
{hand,18,hard,metal,solid}
[tired,sad,angry,confused]
true .

?- ike(football, smell).
{yes,leather,5}
true .

?-
```

Figure 4 - Sample interaction

As seen in figure 4, when asking IKE about a *ball*, it will give the abstraction of the object. The rest of the variables have conflicting information which results in the use of variable.

The second example shows a search for *alarmclock*, where the first response is rejected and the backtracking is shown off. Here the two examples are identical because there is only one object to base the abstraction on, meaning that as far as IKE knows this is the way all *alarmclocks* look and behave. Here is an example of the sound variables placeholder. In the future this variable will be replaced by a deconstructed audio file, but currently it is printed as the onomatopoeic sound *rrrrrrrr*.

The last example shows the ability to ask about specific details about an object. The smell of a football is asked for and a placeholder for the smell of *leather* is used. This will also be replaced by a proper value once the sensors are in place to give accurate information.

7 Results

The seventh and second last chapter of the report evaluates the system in its current state, after all the work is done and the project is nearing completion. First is the evaluation as performed by the developers, followed by a summary of experiments with external subjects who got to try out IKE for the first time and their reactions. Lastly is a review of the system and what could be done in terms of further development.

7.1 Evaluation

As mentioned above, first is the evaluation of the system by its developers and the writers of this thesis. The evaluation goes over the requirements established with the employer at the start of the project. Following this is an evaluation of the user experience compared to the original vision and what could have been. Lastly is an evaluation of the actual performance of the system.

7.1.1 Criteria

In the previous chapter it is noted that there were several requirements collected after the initial meeting with the employer, these being that IKE was to have memory, senses, feelings and objects stored in the memory with senses and feelings bound to them. Following this is an evaluation of the IKE as it turned out and how well it adheres to the core requirements of memory, feelings, senses and the included objects.

- *Memory* – As memory was supposed to be the core part of IKE, it was of paramount importance. It works as well as intended, but it is far from a perfect implementation and it leaves much to be desired. Early on in the report it was mentioned in the paragraph on delimitations that the memory would be very limited in the sense that it would only really store a list of objects and how the objects are related, without encryption or anything of the sort.

  Disregarding these features, the memory is working as intended and it can be edited and read by anyone with access to the files.

However, it cannot be read on the fly and a readme would likely have to be supplied in order to explain how to edit the memory file.

- *Feelings* – The feelings implemented currently have very limited functionality. They can be added to the memory and bound to objects as described, but reading the preparatory research and in particular about phenomenal consciousness it becomes apparent that IKE still has a long way to go.

  IKE's feelings leave much functionality to be desired as it has no real way to experience, assign or otherwise use the feelings for anything other than listing them when told to. The requirement is however fulfilled, since the point of IKE is to combine all the senses and feelings in one system rather than to explore them.

- *Senses* – As with the feelings, the functionality behind the senses is more or less the same; IKE can list the senses bound to an object when asked to. However, currently they are only lines of text. Still, the requirement is fulfilled and IKE has a framework with which to handle and list the senses. It simply has no way to gather them on its own at this point in time.

- *Objects* – This requirement is very much dependant on the first three requirements and is technically three separate requirements on its own, namely the abstractions, queries and ability to store multiple copies of the objects.

  The whole idea of having an abstraction works, but it is not an automatic process. IKE currently has no way to create the abstractions of objects on its own.

  The other two sub-requirements were that it should be able to reply to queries with remembered information and that, including the abstractions, several versions of an object should be possible to store. These requirements are both fulfilled, though the requirement about the abstraction is only partially fulfilled. The functionality of responding with abstractions first and then with detailed examples exists, but the mentioned ability to create the abstractions automatically does not.

7.1.2 Usability/Functionality

With almost every requirement fulfilled the next part to focus on is user experience, briefly mentioned in the third phase of prototyping. Obviously, with IKE's main view being only a Prolog command line, its usability is somewhat limited and needs to be explained. Regardless, it does what it is asked to do, just in a rather modest way. Below the current functionality and usability will be evaluated as it is, as well as what could be better and improved upon.

- *As it is now* – IKE is a command line that can reply to queries with its stored memories of the object that is asked for. Naturally this only includes objects stored in the memory. With only balls, clocks and buns implemented, this of course limits its use, but the memory can easily be expanded for in a future version.

  Should IKE be queried about an object it knows, it will first give the abstraction of it. If queried further, sub-types of the object, such as a golf ball or football in the case of balls will be delivered.

  As for usability, at critical junctures during the process of finding an object, when either going up or down in search depth, a message is delivered. Naturally, IKE will either reply with the result or, should the object not exists or no further version of the requested object exists this will instead be the result.

- *What could be better* – At this point, it is easy to see what parts of the program as a whole that could be improved on if given the opportunity to further study both programming and current technology more in-depth. However, when looking mainly at usability the text could be made more explanatory and there is currently not a lot of text being output should the object not exist. With proper study of user interfaces it could be made clearer without having to change any of the underlying functionality. In essence, what could currently be improved upon on IKE is how text is output and when, as well as the layout.

- *If it could be rebuilt* – Deeply invested scientists and programmers could likely create a better program with the same functionality or more, but even the current developers could most likely make a better program if given more time to put into studying both Prolog and user interfaces. One obvious point would be the core structure of the memory which could be changed to use a tree structure rather than a list.

7.1.3 Performance

Performance is a vital point to many systems, because as datasets get bigger even the most efficient programs will slow down and take longer to process the data. In a worst case scenario, a program may even fail to return when there is too much data to process.

However, when it comes to Prolog, performance is not always the main goal. This is because the Prolog language is mainly focused on functionality, rather than efficiency and speed of execution like lower level languages such as C. Below is an evaluation of IKE's effectiveness, usage scenarios and complexity, as well as whether or not IKE can even be considered an AI.

- *Effectiveness* – In its current implementation, IKE is not as efficient as it could be but it is effective enough for its relatively small size. As explained above the memory is a simple list with a depth of three, consisting of object types, sub-types and instances of specific sub-types. Depending on the query, different depths will of course be explored. As objects in the memory increases however the efficiency will start to decrease.

- *Scenarios* – As the depth or the number of objects increases, execution will naturally take longer. In the best case scenario, IKE is queried about one of the three abstractions ball, clock or bun. In that case IKE only has to make three assertions at most. However, when queried about an object which is not an abstraction and to which the type is not immediately apparent, IKE will have to go through each list of objects in order until the object is found. This means that if the object is not found at all, every object in IKE's memory has been traversed resulting in the longest execution time possible, the worst case scenario.

- *Success* – Above, IKE is explained to be not as efficient as it could be. However, the developers still consider IKE a success, as all of the six requirements are met and the functionality requested of IKE is implemented. There is still much to improve, but the current version of IKE is complete.

7.2 Tests

IKE is a success as far as the developers are involved. However, it is possible to both underestimate and overestimate a piece of work. As such, it is imperative that IKE is tested by external operators as well, operators who have been informed to some extent of IKE and its functionality and purpose. This has been written about previously in the chapter on the fourth phase of prototyping.

7.2.1 Users

For proper testing of IKE, it was decided to have two test cases apart from the first reveal of the model with the employer. The two test cases were performed by students at KTH in Sweden at different points in time. Next is a list of the people who got to test IKE, the employer and the two external testers.

- *Employer* – The first member of the public to experience IKE was the employer who wanted IKE developed. This was done quite early in development before any actual programming was done on the implementation.

- *External tester 1* – The first external tester is a friend who has been a part of the process of writing the thesis by proofreading it several times.

As such the tester is aware of the system, its purpose and how it has been created. The idea behind choosing this tester to test the program is mainly to find any bugs or errors missed during development.

- *External tester 2* – The second external tester was chosen mainly for the special quality of not being a programmer, not being aware of the development of IKE or even knowing about the concept of AI-Complete. As such the tester can give an objective view of the system and what can be expected of it without any subjective preconceptions. The idea is to find out how someone uninvolved will react to the system and what could prove difficult to understand.

7.2.2 Preparation

For the three different test users, three different test cases were required. This is because each the test was carried out at a different point in development. Following this is an explanation of what version of the system was tested, how it was distributed and with what information it was delivered to the testers.

- *What systems* – As mentioned, IKE was shown to the three testers at different points in development. The employer who was first to be exposed to IKE would be shown the model form of IKE. On the other hand, the other two receive the implemented version of IKE to garner valuable feedback from hands-on experience.

- *How distribute* – The second thing to prepare before each test is in which way it should be delivered to the test person. This also includes how the test should be administered, but more on that later. For the employer, IKE is distributed in the form of sketches at the time of the test. The second tester receives the file ready to be run together with some information on how to do so in order to do this at their leisure. The third test is carried out in a similar fashion, but with IKE installed on a computer brought to the test and handed to the tester.

- *What information* – Each test subject received information on how to test IKE. Apart from necessary information for testing, they were also given background information on IKE as a system, its purpose and functionality. They were also briefed on the idea of abstractions and instances of objects.

- *How* – The first test was performed on KTH's Kista campus early in April with the subject in a conference room. The feedback from this test changed IKE as some new ideas were explored. This is explained in more detail below.

  The second test was as described above conducted on the tester's spare time. The code was emailed out together with the developers' contact information in case of questions. A brief readme was supplied to show how the program should be used. The tester was given free rein to test the program as wanted, returning with critique a few days later.

  The third and final test however was done with the person and the two developers and was somewhat different. The test itself was done at the home of one of the developers, with everyone present should any questions arise. No real explanation on how the program works or its purpose was given, except the bare minimum in order to operate it.

7.2.3 Results

Contained here is what information and further ideas were received as feedback from the testers. First is the employer's commentary on the model, followed by the second tester and rounded up with the third.

- *Employer* – The employer has had a lot of influence over the development of IKE as a whole and this feedback is very valuable. A lot of what was discussed was about the nature of feelings and the senses and how they are bound together. A lot of ideas on how to actually implement the model in Prolog were also discussed. Additionally, the idea of being able to search by feelings and not only by object was also discussed but eventually dismissed.

In fact, the main thing taken away from this test was also dismissed in the end. This was the idea to use a database instead of a regular text file, but it was deemed that this could potentially be implemented in the future as it was not of too much importance.

- *External tester 1* – Critique was given two or so days after the system was handed over for testing. Most of the critique was regarding features that were left out, which is valid out of a usability viewpoint. Regarding the implemented functionality there were no complaints. The main point raised was to increase the functionality, particularly the ability to do use the information given, subjects that have been discussed in the delimitations of the thesis.

- *External tester 2* – The second tester was mainly concerned about the usability. The lacking user interface was deemed confusing, which is understandable coming from someone with minimal programming knowledge and who has never used a terminal. Regardless, it is valuable critique as it raises the point that the actual usability of the system in its current state is rather limited. No additional features were requested and no bugs were discovered. The system was thus cleared for release, despite the confusion.

7.3 Review

With both the thoughts of the developers and the testers gathered, a final review of the product can be compiled. While there are certainly some shortcomings, these are not entirely negative and most of them are things to improve and some missing features. Detailed below is first a review of the model, then of the implementation and lastly a review of the system known as IKE as a whole.

7.3.1 Model

Above, most of the focus is on the implementation, mainly because it is easier to test and find faults in, but also because the positives and negatives are easier to determine. The model on the other hand is developed at a very high level, more akin to a blueprint of the system detailing how it would work.

Regardless, the model is developed in accordance with the requirements, meeting all of them and was endorsed by the employer during the first public showing before moving on to implementing the system. As such, the model can be considered satisfactory.

7.3.2 Implementation

As for the implementation, this is the version of IKE that can actually be interacted with and is where the flaws begin to show. That said, the implementation does have the required features to the extent allowed by the delimitations. It is executable and no fatal bugs have been discovered. The problems and flaws begin to show where memory and execution times are concerned. This is mainly due to the previously mentioned list structure of the memory rather than a tree structure.

7.3.3 System

When it comes to evaluating the system as a whole, it becomes increasingly difficult to objectively pass judgment on the system, as its developers. Regardless, going by the fact that all the requirements have been met, it has to be considered at least acceptable, despite its flaws.

8 Conclusions

The final chapter of this report summarises the project and tells what goals have been reached and how IKE relates to the previously mentioned related work in the field of AI and AI-Complete. Afterwards there is a discussion about the work that has gone into writing this report. Finally, the last paragraph of the report is about the future work and what remains before IKE can be considered an AI-Complete system.

8.1 Summary

This thesis presents the construction and findings during the development of IKE, a system approaching AI-Completeness by merging feelings and the five human senses into one large system. IKE has come a long way from being an idea without even a name to actually being a system that can be interacted with in order to get information. This part of the report covers what has been done, more information regarding the data study, how IKE fits into society and research in general, why IKE was created as well as the answers that were reached meanwhile.

8.1.1 What has been done

At the start of the project the need for an extensive data study in order to gather more information about the subject of AI and AI-Completeness quickly became apparent. This paragraph will focus on explaining how the data study went and what results came from it. Following this is a short summary of the work done on both the model and the system before the purpose and the results of the entire project are explored.

- *Data Study* – This concerned the lack of information that was vital to the project, both in terms of work methods to use and more information on the research topic. As mentioned in the first chapter of the report, a well-known database was used to find papers and books containing relevant information on subjects related to AI and AI-Complete. Information regarding prototyping and other potentially applicable system development methods was also researched. The information was gathered and studied in accordance with the chosen methodologies discussed in chapter 1.5 and were applied to the project as evident from chapter 4 and 5.

- *Model Work* – After the data study, work on constructing IKE began. At first, sketches of the model were drawn and various different features for the finished product were considered. At this point in time, how to implement these features and how they would work in the implementation was not the main focus but it was kept in mind to avoid the problems later on. Once the first prototype model of the system was developed it was shown to the employer and a few changes were discussed and then applied and implemented to better fit the vision of the product.

- *System Work* – After the revised model was finished, work began on implementing the system. This was where the finer details of the model were worked out. A working system was quickly completed, because despite the model's limited detail it was developed with how to implement the features in mind. The system was evaluated twice, once near completion and once more after going through a few changes proposed during the first trial run. When the model was to be reviewed it was shown to the employer, but the implementation was instead shown to people who were not already too invested in the development of the system. As such two unrelated students were chosen as they have some general knowledge of development while not being directly involved in the development of this particular system. This was done in order to allow the two testers to give objective feedback.

8.1.2 Related Work

A lot of work has been done, but it is also important to know why. At the start of the thesis, the reasoning behind creating IKE was explained to be the presenting of a prototype for the researchers of AI-Complete systems and also finding out what is still missing.

The answers found during this project are implicitly mentioned throughout the thesis. In the second chapter the results of the data study are written as a background to the system. Several related works are brought up in which various scientists are working on different aspects of AI-Complete problems. However, most of these scientists classify their research as AI in general, which could serve as a pointer to how far away from AI-Completeness the scientists really are.

Just finding information on the subject was rather difficult, even though many of the papers published as general AI problems couple be considered AI-Complete problems. In fact, the very definition of AI-Complete from the 90's label it as tasks that are impossible to solve, which seems to still hold true. In the end, scientists are still very far from realising a truly AI-Complete system.

This is where IKE comes in to bring awareness and hopefully push more researchers to work harder towards finding a solution to the AI-Complete problems. An important part here is therefore to try and see what others could stand to gain from the development of IKE. The main point to explore is how IKE fits into the world of science and what it can contribute to current research.

The idea of giving a robot or AI feelings has been explored by one of the previously mentioned scientists working on creating the phenomenal type of consciousness. However, this scientist focused more on creating a way to let the AI naturally experience feelings. This was done through a sensorimotor approach, where sensors were placed all over the robot and the data was studied in many different ways. The idea was to interpret what sensations could be felt, how and why. IKE is on the other side of the same coin in the sense that the idea with IKE is to create a way to store and handle feelings in relation to objects. By combining these two it could be possible to let an AI experience feelings and be able to remember them, something which would be a great step towards an AI-Complete system.

Another related piece of work is the research carried out by a scientist working on how to integrate feelings, who is also the only one to actually call this research AI-Complete. This was to explore how to handle all the feelings in a natural way and how to combine this with the five human senses. This is done by storing all of the information together in one central nervous system on a object by object basis. IKE can in a sense be seen as a successor to as well as an attempt at an implementation of this research as a lot of the ideas behind IKE are borrowed heavily from this paper, most notably how the senses can be described digitally.

What IKE now brings to the table is a system that is executable, interactable and which actually works with the ideas of several researchers implemented. Researchers can now use these ideas to further their own research. The core ideas IKE explores are as mentioned before how the senses and feelings can be stored in a single memory. That way the senses, feelings and memories are linked is similar to the way a human might think of them.

8.1.3 IKE, AI-Completeness and other researchers

While the answer to the question of how far away from AI-Completeness researchers are today is that it is still very far away, the fact remains that this answer was confirmed. The idea was to present the findings and a system for researchers to experiment on and be inspired by, which is what has been done seeing as the system has been made and scientists can now take inspiration from it for their own research.

It is now up to them to gather more information and be inspired, such that we may soon see the day when a robot with an AI can greet us in a manner indistinguishable from a human. That day is however still far away as one other finding was an approximation about how far away researchers truly are. As mentioned briefly above, researchers today are very far from creating an AI-Complete system. Of course, researchers are not as far away as they were back in the 90's when the term was coined to indicate a problem not solvable by technology at the time.

Technology has advanced and ideas not even imaginable back then are already implemented in today's world, but a robot with an AI like in the world of science fiction is still a tricky subject. A core problem here is that no researcher seems to even want to use the term AI-Complete, most of them are aware of it but no one is brave enough to address the problem. This is possibly because while there is a vast amount of information on the subject of AI-Completeness in books about AI there is a meagre amount of actual research claiming to be about AI-Completeness. That is where IKE comes in, as it has been thought of as an AI-Complete project from the beginning.

8.2 Discussions

Before further delving into what is left to do in the future, this is a good chance to discuss the project as a whole and focus more on the actual project rather than the resulting product. This discussion will cover what we as writers and developers felt went well and what we would have done differently.

To start off the discussion we will go through what has gone well before going on to explore what could have been done differently and finishing up with an evaluation of the methods chosen.

While there have been some doubts as to the relative quality of IKE in its current state, most notably concerning the memory structure, we as developers cannot help but be proud of it. It is a project in which we have started from scratch and developed everything from the ground up, unlike anything else during our studies at KTH. All the information had to be gathered and the architecture, structure and code all had to be created by hand. We did however receive a lot of help and guidance from different parties, although much of this help was towards the writing of this thesis.

Still, it may not be the best. As previously mentioned in the evaluation of IKE as a system, there are certain points where the system could be improved. While we are proud of the program we have created, we must admit that it could have been much more. It is easy to look back at development and see different solutions that might have led to a better end product.

An example of such a solution would be the output and how interactions with IKE take place. If we had had more experience with the language Prolog and how to combine it with other languages, we could have developed a user interface in for example Python, a versatile and easy to use imperative language, while using Prolog as a backend to handle the logic. Even memory handling could have been done in Python to add more features. As mentioned before, IKE currently only reads the memory file as a list which has to be formatted in a particular way for it to work. This is certainly not the best choice and we are aware of this. At the same time, it should pointed out that implementing a highly efficient memory handling algorithm was not part of the goal with IKE. It was simply a necessity for IKE to even work in the first place.

The other big thing was the memory structure. As it stands now it is a simple nested list with tuples at the deepest level. With its current small size, it is relatively easy to handle both in terms of computation speed and in terms of manually editing the memory. The main problem with any list is that once it grows larger, reaching hundreds of thousands of entries, the system will require a substantially longer time to search through each object to see whether or not it is what was asked for.

It also becomes harder to edit the list and without accidentally breaking it, since manually editing the memory bypasses "illegal entries" and IKE is not built to check that its memory is not corrupt.

A much better idea from the point of computational efficiency would be to use a balanced tree structure of leaves and branches where the amount of lookups necessary is theoretically halved with each check.

While a list becomes increasingly difficult to manually edit once it grows large enough, this is even truer for a tree, because you also have to take all parent and child nodes into account and whether any changes will result in an unbalanced tree structure.

Another point that we would have liked to focus more on is the detail of the model. Had it been a more complicated system than IKE is now, the level of detail on the model would have to be higher to sufficiently relay the functionality of the system. This can be in part attributed to our inexperience with developing models, which in hindsight means we should have looked for even more information on the development of models.

As for the methods, we will remember what we have learnt for future projects. Before this thesis, most of the methods described in chapter 3 were unknown to us and a lot of research went into learning about methods. In previous assignments, we have mostly worked without any methods to guide us resulting in unstructured work and sometimes even more work than necessary.

Furthermore, in the future we will already have the research on the methods done and we can simply choose a suitable method and start working. With that said, the most important method to evaluate is prototyping. We are very happy with this choice of work method. Both of us authors feel that agile methods may be great, but a lot of extra work go into just ensuring the agile work method. This may work well for a large team working on a big project, but for a smaller project like this it is more important to focus on producing a quality prototype. This is what prototyping has helped us achieve.

8.3 Future work

Now on to what is left before IKE can reach AI-Completeness. There is naturally a lot left, such as granting IKE a personality, but this part will focus more on things that could still be done with current technology. First is a brief look on the delimitations made in the initial chapter, followed by what is left for IKE and finishing up with what is actually attainable.

8.3.1 Delimitations

The delimitations made were focused mainly on IKE's functionality. That is, functionality that would be a nice addition, but which does not directly contribute to IKE's purpose. This could be functionality such as a better memory implementation which is editable directly through "talking" with IKE or actual sensors for each of the senses. There is also the lack of a visual interface which has been mentioned several times already.

8.3.2 What is left

It may be easier to go through what is not left than what is left, as there is still a lot missing and not everything is known yet.

An example of a missing piece is personality. Properly developing and maintaining a personality for a system like this would require not only AI researchers and programmers, but also psychologists. Included in having a personality could be the ability to act on or react to the feelings it has bound to objects. It could be to be moved to tears from something sad, or jumping with joy from something happy. More important, the ability to assign feelings on its own could come from having a personality.

Potentially, two different IKEs could develop entirely different personalities depending on how they have been made to interact with their surroundings. Hopefully, a personality would help IKE reach the empathy that was one of the main reasons for giving IKE feelings in the first place.

Another big part that remains are the sensors and how to naturally classify data. For example, it would need a tongue to taste things and then translate this to data in an objective way. Alternatively, as humans experience things subjectively, perhaps IKE should as well.

The other senses also require different sensors in order to instantly identify what IKE sees, hears or touches.

Apart from IKE's senses, it also requires a body in order to properly be able to experience its surroundings, or it would have no need to implement all senses together.

8.3.3 What can be further developed

Out of the potential future additions listed above, only a few are even possible with current technology. The user interface is relatively easy to implement. Sensors do exist to cover sight, hearing and touch, but they and the technology powering them are not yet sufficiently advanced. For example, object character recognition, OCR has existed for a long time, but it is used mostly for text recognition and even then it is not always dependable.

For IKE, the objects recognized could number in the thousands and the problem stems from having to identify new objects. Technology such as this is however not restricted to IKE, but should an affordable solution show itself there will be many applications that can make use of it. The sensors for hearing, microphones have also existed for a very long time and are by now highly sophisticated and can easily record and store most types of sounds. The problem here lies more in the software and processing the information gathered. This goes for both sight and hearing.

As far as the code goes, IKE's memory could be given numerous improvements, for example being turned into a proper database with proper protection against tampering.

Needless to say, the future for AI is bright, but not yet very clear. As research progresses it is constantly getting clearer and eventually the day might come where IKE can greet you on the street and tell you about a bakery which has delicious cinnamon buns.

References

[1] B. J. Copeland, "Colossus: its origins and originators," *IEEE Ann. Hist. Comput.*, vol. 26, no. 4, pp. 38–45, Oct. 2004.

[2] J. Schmidhuber, "2006: Celebrating 75 Years of AI - History and Outlook: The Next 25 Years," in *50 Years of Artificial Intelligence*, M. Lungarella, F. Iida, J. Bongard, and R. Pfeifer, Eds. Springer Berlin Heidelberg, 2007, pp. 29–41 [Online]. Available: http://link.springer.com.focus.lib.kth.se/chapter/10.1007/978-3-540-77296-5_4. [Accessed: 26-Feb-2015]

[3] R. Cordeschi, "Ai turns fifty: revisiting its origins," *Appl. Artif. Intell.*, vol. 21, no. 4–5, pp. 259–279, Apr. 2007.

[4] S. Kubrik, *2001: A Space Odyssey*. Metro-Goldwyn-Mayer, 1968.

[5] A. Håkansson, "AIC–An AI-system for Combination of Senses," *Procedia Comput. Sci.*, vol. 22, pp. 40–49, 2013.

[6] H. Lipson, "Curious and Creative Machines," in *50 Years of Artificial Intelligence*, M. Lungarella, F. Iida, J. Bongard, and R. Pfeifer, Eds. Springer Berlin Heidelberg, 2007, pp. 315–319 [Online]. Available: http://link.springer.com.focus.lib.kth.se/chapter/10.1007/978-3-540-77296-5_28. [Accessed: 27-Feb-2015]

[7] E. S. Raymond, "AI-complete," *Jargon File*. 23-Mar-1991 [Online]. Available: http://catb.org/esr/jargon/oldversions/jarg282.txt. [Accessed: 19-Feb-2015]

[8] R. V. Yampolskiy, "Turing Test as a Defining Feature of AI-Completeness," *Artif. Intell. Evol. Comput. Metaheuristics-- Footsteps Alan Turing Xin-She Yang Ed*, pp. 3–17, 2013.

[9] Z. J. McDowell, "The machine question: Critical perspectives on AI, robots, and ethics," *New Media Soc.*, vol. 16, no. 6, pp. 1041–1043, Sep. 2014.

[10] P. Z. Shah, "Pakistan Says U.S. Drone Kills 13," *The New York Times*, 19-Jun-2009 [Online]. Available: http://www.nytimes.com/2009/06/19/world/asia/19pstan.html. [Accessed: 24-Mar-2015]

[11] D. Jon-Erik, "Sustainable Development: Introduction, definitions and perspectives," Royal Instiitute of Technology, Kista, 10-Aug-2014.

[12] A. Håkansson, "Portal of Research Methods and Methodologies for Research Projects and Degree Projects," presented at the The 2013 World Congress in Computer Science, Computer Engineering, and Applied Computing WORLDCOMP 2013; Las Vegas, Nevada, USA, 22-25 July, 2013, pp. 67–73 [Online]. Available: http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A677684&dswid=8266. [Accessed: 26-Feb-2015]

[13] "Elsevier Science Acquires Engineering Information." [Online]. Available: http://newsbreaks.infotoday.com/NewsBreaks/Elsevier-Science-Acquires-Engineering-Information-18017.asp. [Accessed: 30-Apr-2015]

[14] Engineering Village, "Compendex," *Engineering Village*. [Online]. Available: http://www.engineeringvillage.com/

[15] S. M. Shieber, "Lessons from a Restricted Turing Test," *ArXivcmp-Lg9404002*, Apr. 1994 [Online]. Available: http://arxiv.org/abs/cmp-lg/9404002. [Accessed: 02-Mar-2015]

[16] A. M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proc. Lond. Math. Soc.*, vol. s2–42, no. 1, pp. 230–265, Jan. 1937.
[17] D. Proudfoot, "Anthropomorphism and AI: Turing's much misunderstood imitation game," *Artif. Intell.*, vol. 175, no. 5–6, pp. 950–957, Apr. 2011.
[18] J. B. Pollack, "Mindless Intelligence," *IEEE Intell. Syst.*, vol. 21, no. 3, pp. 50–56, Jan. 2006.
[19] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Pearson Education, 2003.
[20] R. Trappl, "AI: Introduction, paradigms, applications (including CBR), impacts, visions," in *Advanced Topics in Artificial Intelligence*, V. Mřrík, O. Štěpánková, and R. Trappl, Eds. Springer Berlin Heidelberg, 1992, pp. 1–24 [Online]. Available: http://link.springer.com.focus.lib.kth.se/chapter/10.1007/3-540-55681-8_27. [Accessed: 26-Feb-2015]
[21] A. S. da Silva, X. Gao, and P. Andreae, "Wallace: Incorporating Search into Chatting," in *PRICAI 2014: Trends in Artificial Intelligence*, D.-N. Pham and S.-B. Park, Eds. Springer International Publishing, 2014, pp. 842–848 [Online]. Available: http://link.springer.com.focus.lib.kth.se/chapter/10.1007/978-3-319-13560-1_68. [Accessed: 27-Feb-2015]
[22] S. Sinkevicius, A. Lipnickas, and K. Rimkus, "Automatic amber gemstones identification by color and shape visual properties," *Eng. Appl. Artif. Intell.*, vol. 37, pp. 258–267, Jan. 2015.
[23] G. Angelini, M. Ernandes, and M. Gori, "Webcrow: A Web-Based Crosswords Solver," in *Intelligent Technologies for Interactive Entertainment*, M. Maybury, O. Stock, and W. Wahlster, Eds. Springer Berlin Heidelberg, 2005, pp. 295–298 [Online]. Available: http://link.springer.com.focus.lib.kth.se/chapter/10.1007/11590323_37. [Accessed: 26-Feb-2015]
[24] J. Cameron, *The Terminator*. Orion Pictures, 1984.
[25] S. Putchala and N. Agarwal, "Machine vision: an aid in reverse Turing test," *AI Soc.*, vol. 26, no. 1, pp. 95–101, Nov. 2009.
[26] R. G. Feyen, "Bridging the Gap: Exploring Interactions Between Digital Human Models and Cognitive Models," in *Digital Human Modeling*, V. G. Duffy, Ed. Springer Berlin Heidelberg, 2007, pp. 382–391 [Online]. Available: http://link.springer.com.focus.lib.kth.se/chapter/10.1007/978-3-540-73321-8_44. [Accessed: 26-Feb-2015]
[27] D. W. Carruth, M. D. Thomas, B. Robbins, and A. Morais, "Integrating Perception, Cognition and Action for Digital Human Modeling," in *Digital Human Modeling*, V. G. Duffy, Ed. Springer Berlin Heidelberg, 2007, pp. 333–342 [Online]. Available: http://link.springer.com.focus.lib.kth.se/chapter/10.1007/978-3-540-73321-8_39. [Accessed: 26-Feb-2015]
[28] S. M. Potter, "What Can AI Get from Neuroscience?," in *50 Years of Artificial Intelligence*, M. Lungarella, F. Iida, J. Bongard, and R. Pfeifer, Eds. Springer Berlin Heidelberg, 2007, pp. 174–185 [Online]. Available: http://link.springer.com.focus.lib.kth.se/chapter/10.1007/978-3-540-77296-5_17. [Accessed: 27-Feb-2015]

[29] J. K. O'Regan, "How to Build Consciousness into a Robot: The Sensorimotor Approach," in *50 Years of Artificial Intelligence*, M. Lungarella, F. Iida, J. Bongard, and R. Pfeifer, Eds. Springer Berlin Heidelberg, 2007, pp. 332–346 [Online]. Available: http://link.springer.com.focus.lib.kth.se/chapter/10.1007/978-3-540-77296-5_30. [Accessed: 27-Feb-2015]

[30] J. Arnowitz, M. Arent, and N. Berger, *Effective Prototyping for Software Makers*. Burlington, MA, USA: Morgan Kaufmann, 2006 [Online]. Available: http://site.ebrary.com.focus.lib.kth.se/lib/kth/detail.action?docID=10160310. [Accessed: 28-Feb-2015]

[31] G. Suganya and S. A. S. A. Mary, "Progression towards Agility: A comprehensive survey," in *2010 International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2010, pp. 1–5.

[32] J. Shore and Chromatic, *The Art of Agile Development*. O'Reilly Media, Inc., 2007 [Online]. Available: http://proquest.safaribooksonline.com.focus.lib.kth.se/9780596527679. [Accessed: 28-Feb-2015]

TRITA TRITA-ICT-EX-2015:54