

The History of Artificial Intelligence

History of Computing
CSEP 590A

University of Washington
December 2006

Introduction – *Chris Smith*

The Turing Test – *Brian McGuire*

History of AI applied to Chess – *Chris Smith*

Expert Systems – *Ting Huang*

AI Winter and its lessons – *Gary Yang*

Japan's Fifth Generation Computer System project – *Chris Smith*

Conclusion – *Chris Smith*

Table of Contents

Introduction	4
'Artificial' intelligence	4
Themes of AI	4
The Turing Test	5
Introduction	5
Alan Turing	5
Inception of the Turing Test.....	6
Problems/Difficulties with the Turing Test	6
Alternatives to the Turing Test	7
The Current State of the Turing Test	8
Conclusion.....	9
History of AI applied to Chess	10
Origins of computer-Chess.....	10
Realization.....	10
Go as the next frontier.....	11
Conclusion.....	11
Expert Systems.....	12
Overview	12
Key Technological Issues.....	12
Managerial and Organizational Challenges	14
Is "Thinking" Machine Ever Possible	14
Social Implications of Expert Systems.....	15
Concluding Remarks.....	16
AI Winter and its lessons.....	17
Overview	17
Trigger of AI Winter	17
The ALPAC report.....	17
The Lighthill report.....	18
The Duration of AI Winter.....	18
Discussion.....	19

Conclusion.....	21
Japan's Fifth Generation Computer System project	22
Motivations and influences.....	22
Progress made on FGCS	23
Successes.....	23
Failures.....	23
Lessons learned.....	23
Conclusion.....	24
References	25
Introduction, History of AI applied to Chess, and Japan's FGCS	25
Turing Test	25
AI Winter	26
Expert Systems.....	26

Introduction

Artificial Intelligence (AI) has been studied for decades and is still one of the most elusive subjects in Computer Science. This partly due to how large and nebulous the subject is. AI ranges from machines truly capable of thinking to search algorithms used to play board games. It has applications in nearly every way we use computers in society.

This paper is about examining the history of artificial intelligence from theory to practice and from its rise to fall, highlighting a few major themes and advances.

'Artificial' intelligence

The term artificial intelligence was first coined by John McCarthy in 1956 when he held the first academic conference on the subject. But the journey to understand if machines can truly think began much before that. In Vannevar Bush's seminal work *As We May Think* [Bush45] he proposed a system which amplifies people's own knowledge and understanding. Five years later Alan Turing wrote a paper on the notion of machines being able to simulate human beings and the ability to do intelligent things, such as play Chess [Turing50].

No one can refute a computer's ability to process logic. But to many it is unknown if a machine can *think*. The precise definition of think is important because there has been some strong opposition as to whether or not this notion is even possible. For example, there is the so-called 'Chinese room' argument [Searle80]. Imagine someone is locked in a room, where they were passed notes in Chinese. Using an entire library of rules and look-up tables they would be able to produce valid responses in Chinese, but would they really 'understand' the language? The argument is that since computers would always be applying rote fact lookup they could never 'understand' a subject.

This argument has been refuted in numerous ways by researchers, but it does undermine people's faith in machines and so-called expert systems in life-critical applications.

Themes of AI

The main advances over the past sixty years have been advances in search algorithms, machine learning algorithms, and integrating statistical analysis into understanding the world at large. However most of the breakthroughs in AI aren't noticeable to most people. Rather than talking machines used to pilot space ships to Jupiter, AI is used in more subtle ways such as examining purchase histories and influence marketing decisions [Shaw01].

What most people think of as 'true AI' hasn't experienced rapid progress over the decades. A common theme in the field has been to overestimate the difficulty of foundational problems. Significant AI breakthroughs have been promised 'in 10 years' for the past 60 years. In addition, there is a tendency to redefine what 'intelligent' means after machines have mastered an area or problem. This so-called 'AI Effect' contributed to the downfall of US-based AI research in the 80s.

In the field of AI expectations seem to always outpace the reality. After decades of research, no computer has come close to passing the Turing Test (a model for measuring 'intelligence'); Expert Systems have grown but have not become as common as human experts; and while we've built software that can beat humans at some games, open ended games are still far from the mastery of computers. Is the problem simply that we haven't focused enough resources on basic research, as is seen in the AI winter section, or is the complexity of AI one that we haven't yet come to grasp yet? (And instead, like in the case of computer Chess, we focus on much more specialized problems rather than understanding the notion of 'understanding' in a problem domain.)

This paper will go into some of these themes to provide a better understanding for the field of AI and how it has developed over the years. In looking at some of the key areas of AI work and the forces that drove them, perhaps we can better understand future developments in the field.

The Turing Test

Introduction

The Turing test is a central, long term goal for AI research – will we ever be able to build a computer that can sufficiently imitate a human to the point where a suspicious judge cannot tell the difference between human and machine? From its inception it has followed a path similar to much of the AI research. Initially it looked to be difficult but possible (once hardware technology reached a certain point), only to reveal itself to be far more complicated than initially thought with progress slowing to the point that some wonder if it will ever be reached. Despite decades of research and great technological advances the Turing test still sets a goal that AI researchers strive toward while finding along the way how much further we are from realizing it.

In 1950 English Mathematician Alan Turing published a paper entitled “Computing Machinery and Intelligence” which opened the doors to the field that would be called AI. This was years before the community adopted the term Artificial Intelligence as coined by John McCarthy[2]. The paper itself began by posing the simple question, “Can machines think?”[1]. Turing then went on to propose a method for evaluating whether machines can think, which came to be known as the Turing test. The test, or “Imitation Game” as it was called in the paper, was put forth as a simple test that could be used to prove that machines could think. The Turing test takes a simple pragmatic approach, assuming that a computer that is indistinguishable from an intelligent human actually has shown that machines can think.

The idea of such a long term, difficult problem was a key to defining the field of AI because it cuts to the heart of the matter – rather than solving a small problem it defines an end goal that can pull research down many paths.

Without a vision of what AI could achieve, the field itself might never have formed or simply remained a branch of math or philosophy. The fact that the Turing test is still discussed and researchers attempt to produce software capable of passing it are indications that Alan Turing and the proposed test provided a strong and useful vision to the field of AI. It’s relevance to this day seems to indicate that it will be a goal for the field for many years to come and a necessary marker in tracking the progress of the AI field as a whole. This section will explore the history of the Turing test, evaluate its validity, describe the current attempts at passing it and conclude with the possible future directions the Turing test solution may take.

Alan Turing

Alan Turing was an English mathematician who is often referred to as the father of modern computer science[3]. Born in 1911, he showed great skill with mathematics and after graduating from college he published a paper “On Computable Numbers, with an Application to the Entscheidungs problem” in which he proposed what would later be known as a Turing Machine – a computer capable of computing any computable function.

The paper itself was built on ideas proposed by Kurt Godel that there are statements about computing numbers that are true, but that can’t be proven[5]. Alan Turing worked on the problem in an effort to help define a system for identifying which statements could be proven. In the process he proposed the Turing Machine. The paper defines a “computing machine” with the ability to read and write symbols to a tape using those symbols to execute an algorithm [4]. This paper and the Turing machine provided that basis for the theory of computation.

While Alan Turing focused primarily on mathematics and the theory of what would become computer science during and immediately after college, soon World War 2 came and he became interested in more practical matters. The use of cryptography by the Axis gave him reason to focus on building a machine capable of breaking ciphers. Before this potential use presented itself, Alan Turing likely hadn’t been too concerned that the Turing machine he’d proposed in his earlier work was not feasible to build.

In 1939 he was invited to join the Government Code and Cipher school as a cryptanalyst[5] and it became clear that he needed to build a machine capable of breaking codes like Enigma which was used by the Germans. He designed in a few weeks and received funding for the construction electromechanical machines called ‘bombes’ which would be used to break Enigma codes and read German messages by automating the processing of 12

electrically linked Enigma scramblers. It wasn't the Turing machine, but the concepts of generating cyphertext from plaintext via a defined algorithm clearly fit with the Turing machine notion.

After the war Turing returned to academia and became interested in the more philosophical problem of what it meant to be sentient, which led him down the path to the Turing test.

Inception of the Turing Test

In 1950 Alan Turing was the Deputy Director of the computing laboratory at the University of Manchester. The paper which defined what would come to be known as the Turing test was published in a Philosophical journal called *Mind*. The paper itself was based on the idea of an 'Imitation Game'. If a computer could imitate the sentient behavior of a human would that not imply that the computer itself was sentient? Even though the description itself is fairly simple, the implications of building a machine capable of passing the test are far reaching. It would have to process natural language, be able to learn from the conversation and remember what had been said, communicate ideas back to the human and understand common notions, displaying what we call common sense.

Similar to how he used the Turing Machine to more clearly formalize what could or could not be computed, Alan Turing felt the need to propose the Turing Test so that there was a clear definition of whether or not the responses given by a human were part of the computable space. In the paper he wanted to replace the question, 'Can machines think?' (which can have many possible answers and come down to a difference of opinion) with a version of the 'Imitation Game.'

The original game upon which Turing's idea was based required a man, a woman and an interrogator. The goal was for the interrogator to identify which of the participants was a man and which was a woman. Since the interrogator would be able to identify the gender of the respondent by their voice (and maybe handwriting) the answers to the interrogator's questions would be type written or repeated by an intermediary. For the Turing Test, one of those two participants would be replaced by a machine and the goal of the interrogator would not be to identify the gender of the participants, but which is human and which is a machine.

As described above, the Turing Test has a few key components that in effect define what Turing means when he wonders if machines can think. First the interrogator knows that there is one human and one machine. The test doesn't just require a computer to fool a human into thinking it is sentient; it asks the computer to fool a suspicious human. Second, physical nature isn't important – the goal is to not be able to tell the difference between man and machine when comparing the output of the machine and the true human. The communication medium is such that there are absolutely no hints beyond what can be expressed with written language. Also, the test doesn't include anything specific – no complex problem solving or requests to create art. As described, it seems a machine would pass the Turing test if it were able to make small talk with another human and understand the context of the conversation. For Turing, passing such a test was sufficient for him to believe that machines were capable of thinking.

Beyond defining the game, the paper continues with an introduction to digital computers and how they can be used for arbitrary computation – harkening back to the description of the Turing machine. Taken with Godel's incompleteness theorem and Turing's formalization of what can and cannot be computed, the Turing test seems to strike at the simple question of whether that ability to appear sentient falls in to the realm of computable problems that a Turing machine can handle, or if it falls under the tiny subset of things that are true, but cannot be proven so. The test is simple, but the question is hugely significant and tied in to Turing's earlier work towards formalizing what can be computed.

Problems/Difficulties with the Turing Test

A large portion of Turing's original paper deals with addressing counter arguments concerning how the test he proposes may not be valid. In the introduction to that section he states that he believes there will be computers with enough storage capacity to make them capable of passing the Turing test "in about fifty years". The statement is interesting because it seems to imply that the AI software required to pass the Turing Test would be

rather straightforward and that the limiting factor would only be memory. Perhaps this limitation was at the front of his mind because he was routinely running into problems that he could have solved if only there were enough storage available. The same type of reasoning is similar to what happens today when we believe that Moore's law will let us solve the hard problems.

Beyond the storage limitations, he also raises other objections, including those based in theology (the god granted immortal soul is necessary for sentience), mathematical arguments based on Godel's work, the ability for humans to create original works and experience emotion, and others.

One of the more interesting contradictions to the test is what he terms 'The Argument from Consciousness.' The argument goes that just imitating a human would not be enough because it doesn't invoke the full range of what it is that we consider to be human. Specifically, the Turing Test could be passed by a machine unable to do things such as write a poem or piece of music wrapped up as part of an emotional response. A machine passing the Turing test would not really have to experience or interpret art either. Turing argues that it is impossible to tell if the machine is feeling unless you are the machine, so there is no way to contradict the claim or to prove it. Using that method to dismiss the argument, he points out that the Turing test could include the machine convincing the interrogator that it is feeling something, even if there is truly no way to know that the emotions are actually being felt the way they would in a human. This would be similar to how humans communicate to convince each other of what they are feeling, though there is no guarantee that it is really true.

Another interesting counter argument against the test that Turing describes is 'Lady Lovelace's Objection.'" She posited that because machines can only do what we tell them, they cannot originate anything, while it is clear that humans do originate new concepts and ideas all of the time. At the time this was written it may not have been possible to model the learning process, but much of the progress that has been made in teaching machines to learn and infer seems to have shown that this issue can be overcome. There have been specific implementations where voice or character recognition is reached by software training itself to recognize the variances in human writing or dialect. At least in these specific cases a machine can recognize something new so perhaps they will be able to in the general case as well.

Overall the potential problems with the Turing test appear to fall in one of two categories:

Does imitating a human actually prove intelligence or is it just a hard problem

Is intelligence possible without passing the Turing test

It seems fair to say that passing the Turing test is only a subset of the situation that humans have to contend with on a day to day basis. So it is possible that there are other key capabilities like experiencing emotions, having core beliefs or motivations, or problem solving that might be simulated in a computer but would not necessarily be the same as what humans do. The Turing test avoids these questions by judging the computer (and human) only on the text they output as part of the casual conversation that takes place during the test. So even if a computer could pass the Turing test, is that enough to say machines are 'intelligent' or that they can 'think', or does that just say that they can now pass the Turing test, and there is much more to understand before we do consider them intelligent.

Beyond that, there are many humans that we'd consider sentient – young children for instance, that would probably do poorly in the Turing test because they haven't accumulated enough knowledge and experience in communication. We wouldn't apply the Turing test to them and say that they therefore are not capable of thought, which means that it might be possible for a computer to 'think' but still not pass the Turing test.

Alternatives to the Turing Test

Many people have proposed their own version of the Turing test to help contend with the perceived or possible shortcomings of the test as proposed by Alan Turing. Most of the alternatives either narrow the scope of the test to make it easier to pass (a more reachable goal), or shift the scope to an area where researchers might make better progress. One alternative, called the Feigenbaum Test, avoids the issues that make it difficult for a

computer to communicate in a causal manner as is done in the Turing test. The Feigenbaum test asks the computer to be able to pass as an expert in a particular field, essentially setting a mark for when technology like that in Expert systems has matured[6]. This test definition does a couple of things – while eliminating the casual, anything goes, nature of the Turing interrogator, the test now requires that the computer be able to solve problems that another expert would be able to solve. In some ways the test is harder, given the expert problem solving portion, while being easier in others where the casual conversation isn't needed.

Another variation on the Turing test was proposed by Nicholas Negroponte, one of the co-founders of the MIT Media Lab. In his test he determines that the value in a 'thinking' computer is its ability to work together with a human instead of passing an interrogation by a human. Rather than have a human interrogate the machine to see if it is a human as well, the test would be whether or not the machine could help the human reach his or her goals in the same way another human would[7]. This test was believed to be more difficult because the computer would need an even better ability to understand the vagueness or incorrectness of what is communicated to it by the human.

The Current State of the Turing Test

Since the Turing test was proposed many attempts have been made to build software capable of passing it (or a more constrained version). A limited form of the test - temporarily convincing a non-suspicious human that a computer is actually another person has been possible for some time[7], but the true test itself has never been passed.

As the Turing test is such a fundamental and difficult problem, the research into it tends to be slow and incremental – trying to build something better than the last. Faster computers with more memory haven't been a great boon to the progress, it seems because the problem isn't in having enough time and storage, but in devising the correct algorithms to make use of the processing power and storage that we have available to us. The limited progress against the Turing test is perhaps due to the reduction of research in AI or the extreme difficulty of the problem. In the end, despite the efforts that have been put forth, not a lot of progress has been made.

Currently there is one mechanism to encourage further research - the Loebner Prize and the associated Turing Test Competition[8]. The prize includes a \$100,000 cash reward for a system capable of being indistinguishable from a human. To encourage participation, the yearly competition has a few different levels of reward. Each year a medal and \$2000 is awarded to the best entry. Given the size of the reward it may not be surprising that few researchers actively work toward attaining it when there are far more lucrative avenues for research.

The Loebner prize web site includes the transcripts of the top four entries in each year going back to 1995. In reviewing the conversations for the winning entries it is amazing how far they are from being indistinguishable from a human and passing the test. The difference between human and computer is significant with the software often replying in ways that make no sense. At times the winning program offers seemingly random responses to questions that don't seem to be complicated.

Here are some example exchanges in the 2005 contest between the judge and the winning program, Jaberwocky. In this example the judge types in random characters:

JUDGE: Hi George, cslkjslw weicmslw weos, right? PROGRAM: Don't repeat your name, csthiang!
--

It seems the program has determined that the letters are part of the judges name, which isn't an assumption a human would likely make.. When the judge carried out the same exchange with a human the results were different, causing the human to ask questions rather than making assumptions:

JUDGE: Wljso lwjeido wleje elwjeo wjeol, isn't it? CONFEDERATE: Is that Polish?
--

JUDGE: Who.
CONFEDERATE: What you wrote above.
JUDGE: This is ljelwl, aren't you.
CONFEDERATE: I'm a little worried about your writing skills.

Often the winning system just appears to be responding at random with exchanges similar to the one below:

JUDGE: OK, enough for now. I'll go chat with the other entity.
PROGRAM: Ok then it was nice to share your hair brush.
JUDGE: What do mean by that? How have you shared my hair brush?
PROGRAM: Good.
JUDGE: Good?
PROGRAM: Yes.
JUDGE: This has become a very zen-like conversation.
PROGRAM: And worse yet, I gave it to charity.
JUDGE: What? WHat did you give to charity? This conversation?
PROGRAM: Er.. 1 million Dollars?
JUDGE: Which charity?
PROGRAM: Are we talking about the simpsons or unreal?
JUDGE: I'm not sure what the hell we're talking about, actually.

The difficulty of the Turing test and our lack of progress in attaining it can be seen when the winning systems provide such convoluted and disjointed conversations as the one above. While it is possible for software to mimic human responses for very small subsets of conversation with predefined branching points, general conversation is still far beyond what even the best systems have been able to produce.

Conclusion

Alan Turing's original hope that the test would be passed by a computer by the year 2000 has not been realized. Despite all the effort expended by the research community, advances in processor technology, and cheap memory, no computer has yet been able to approach passing the Turing test. It is clear that the Moore's law increase in computation power hasn't been the driving force in improvement in Turing Test focused AI, instead it is a software problem. Software architectures, such as the Expert Systems (discussed later) offer possible solutions as their designs are refined and applied to various problems, perhaps including the Turing test or one of its derivatives.

History of AI applied to Chess

Chess has long been considered a game of intellect, and many pioneers of computing felt that a chess-playing machine would be the hallmark of true artificial intelligence. While the Turing Test is a grand challenge to ascertain machine intelligence, chess too is a good pursuit, one which fortunately has been 'solved' by AI researchers; producing programs which can rival if not best the world's best chess players. However, even the best game-playing machines still do not *understand* concepts of the game and merely rely on brute force approaches to play.

Origins of computer-Chess

Chess and intelligence have always been linked; the ability to play chess was even used as a valid question to ask during a Turing Test in Turing's original paper. Many people envisioned machines one day being capable of playing Chess, but it was Claude Shannon who first wrote a paper about developing a chess playing program [Shannon50].

Shannon's paper described two approaches to computer chess: Type-A programs, which would use pure brute force, examining thousands of moves and using a min-max search algorithm. Or, Type-B, programs which would use specialized heuristics and 'strategic' AI, examining only a few, key candidate moves.

Initially Type-B (strategic) programs were favored over Type-A (brute force) because during the 50s and 60s computers were so limited. However, in 1973 the developers of the 'Chess' series of programs (which won the ACM computer chess championship 1970-72) switched their program over to Type-A. The new program, dubbed 'Chess 4.0' went on to win a number of future ACM computer chess titles. [WikiChess]. This change was an unfortunate blow to those hoping of finding a better understanding of the game of chess through the development of Type-B programs.

There were several important factors in moving away from the arguably more intelligent design of a Type-B program to a dumber Type-A. The first was simplicity. The speed of a machine has a direct correlation to a Type-A program's skill, so with the trend being machines getting faster every year it is easier to write a strong Type-A program and 'improve' a program by giving it more power through parallelization or specialized hardware. Whereas a Type-B program would need to be taught new rules of thumb and strategies – regardless of how much new power was being fed to it. Also, there was the notion of predictability. The authors of 'Chess' have commented on the stress they felt during tournaments where their Type-B program would behave erratically in accordance to different hard-coded rules.

To this day Type-A (brute force) programs are the strongest applications available. Intelligent Type-B programs exist, but it is simply too easy to write Type-A programs and get exceptional play just off of computer speed. Grandmaster-level Type-B programs have yet to materialize since more research must be done in understanding and abstracting the game of chess into (even more) rules and heuristics.

Realization

Perhaps the best known Type-A program is IBM's Deep Blue. In 1997 Deep Blue challenged and defeated the then world chess champion Gary Kasparov. The 3.5/2.5 match win wasn't a decisive victory but with machines continually increasing in power, many feel the match was just a taste of things to come.

Few surprised by a computer beating a world chess champion. Scientist David G Stoke explained this notion of expected computer superiority with: "Nowadays, few of us feel deeply threatened by a computer beating a world chess champion – any more than we do at a motorcycle beating an Olympic sprinter." Most of this sentiment is due to Deep Blue being a Type-A program. Deep Blue evaluated around 200 million positions a second and averaged 8-12 ply search depth. (Up to 40 under certain conditions.) [CambleHsuHoane02] Humans on the other hand are generally thought to examine near 50 moves to various depths. If Deep Blue were a Type-B program then perhaps the win would have been more interesting from the standpoint of machine intelligence.

Another interesting implication of Deep Blue's win was IBM's financial gain from the match. Some estimates put the value of Deep Blue's win to the tune of \$500 million in free advertising for IBM Super computers. In addition, the match drove IBM's stock price up \$10 to a then all-time high [ScaeffePlatt97].

Go as the next frontier

Computer chess continues developed in order to create more powerful machines with less-specialized hardware, however little interest is placed on Type-B programs which try to capture the raw knowledge of human chess players. With chess being 'solved' to many, attention is now played on other games which computers have yet to challenge humans – one game in particular stands out as being the next frontier of game-based AI research: the ancient Asian game of Go.

What differentiates Go is the inability to correlate computing power into strength. Whereas chess has a branching factor of about 40, Go typically has a branching factor of 200. (Valid moves a player can make at any point in time.) [Muller00] In addition, there is no clear victory condition. Where chess algorithms search for a checkmate, the end-game determination in Go requires agreement by both players as well as extensive search to understand if the board position is terminal. Despite years of research there still isn't a clear algorithm for determining whether or not a group of stones is alive or dead (a fundamental concept of the game). Even with a \$1,000,000 prize offered to whoever can create a program capable of defeating a strong amateur Go player (the Ing prize), even the best programs are not capable of defeating a weak club-level player.

Go programs are slowly making advances however, mainly in the 'Type-B' arena. For example, a branch of image processing known as mathematical morphology has aided in helping computers quickly understand board positions much like the way humans process them [Bouzy03]. Though, perhaps unfortunately, people are beginning to see how massive parallelization can begin to marginally improve the strength of computer-Go programs [DoshayMcDowel05].

Conclusion

Despite excellent progress in making powerful chess playing machines, many find difficulty in calling them intelligent for the same reasons as the 'Chinese Room Argument'. Examining billions of board positions to arrive at a single move may not seem as a highly intelligent approach, but the net result is solid play. Hopefully in the coming years more difficult games will require new abstractions for understanding game logic, which would have a more tangible relationship to general artificial intelligence; opposed to specialized search algorithms.

Expert Systems

Overview

Expert systems are computer programs aiming to model human expertise in one or more specific knowledge areas. They usually consist of three basic components: a knowledge database with facts and rules representing human knowledge and experience; an inference engine processing consultation and determining how inferences are being made; and an input/output interface for interactions with the user.

According to K. S. Metaxiotis et al [1], expert systems can be characterized by:

- using symbolic logic rather than only numerical calculations;
- the processing is data-driven;
- a knowledge database containing explicit contents of certain area of knowledge; and
- the ability to interpret its conclusions in the way that is understandable to the user.

Expert systems, as a subset of AI, first emerged in the early 1950s when the Rand-Carnegie team developed the general problem solver to deal with theorems proof, geometric problems and chess playing [2]. About the same time, LISP, the later dominant programming language in AI and expert systems, was invented by John McCarthy in MIT [3].

During the 1960s and 1970s, expert systems were increasingly used in industrial applications. Some of the famous applications during this period were DENDRAL (a chemical structure analyzer), XCON (a computer hardware configuration system), MYCIN (a medical diagnosis system), and ACE (AT&T's cable maintenance system). PROLOG, as an alternative to LISP in logic programming, was created in 1972 and designed to handle computational linguistics, especially natural language processing. At that time, because expert systems were considered revolutionary solutions capable of solving problems in any areas of human activity, AI was perceived as a direct threat to humans. It was a perception that would later bring an inevitable skeptical backlash [4].

The success of these systems stimulated a near-magical fascination with smart applications. Expert systems were largely deemed as a competitive tool to sustain technological advantages by the industry. By the end of 1980s, over half of the Fortune 500 companies were involved in either developing or maintaining of expert systems [5]. The usage of expert systems grew at a rate of 30% a year [6]. Companies like DEC, TI, IBM, Xerox and HP, and universities such as MIT, Stanford, Carnegie-Mellon, Rutgers and others have all taken part in pursuing expert system technology and developing practical applications. Nowadays, expert systems has expanded into many sectors of our society and can be found in a broad spectrum of arenas such as health care, chemical analysis, credit authorization, financial management, corporate planning, oil and mineral prospecting, genetic engineering, automobile design and manufacture and air-traffic control.

As K. S. Metaxiotis and colleagues [1] pointed out, expert systems are becoming increasingly more important in both decision support which provides options and issues to decision makers, and decision making where people can make decisions beyond their level of knowledge and experience. They have distinct advantages over traditional computer programs. In contrast to humans, expert systems can provide permanent storage for knowledge and expertise; offer a consistent level of consultation once they are programmed to ask for and use inputs; and serve as a depository of knowledge from potentially unlimited expert sources and thereby providing comprehensive decision support.

Key Technological Issues

There are many unresolved technological issues and performance limitations that severely affect the development and implementation of expert systems. The most critical of these problems confronting corporations and institutions and the solutions thereof will be examined in this section.

The key technological issues facing expert systems lie in the areas of software standards and methodology, knowledge acquisition, handling uncertainty, and validation [7].

Software Standards and Interoperability

There are no general standards in expert system software and development methodology. Neither do commonly adopted expert system protocols and infrastructure exist. The knowledge systems are often developed uniquely with little consideration for interoperability.

Recent efforts in defining expert system standards have been actively pursued by a coalition of the American Association of Artificial Intelligence (AAAI), the IEEE Computer Society, DARPA, and the US government. Development complexity, costs, and risks could be greatly reduced once widely accepted standards are established and a new generation of expert system tools is in place.

Knowledge Acquisition and Analysis

Knowledge acquisition is usually considered as a way to discover static facts of the world and the relationships of various events that human uses in solving real life problems. The problem-solving skills in humans oftentimes are far more complicated and complex than what knowledge collection can achieve. For example, humans learn how to walk at an early age through practice and sometimes painful experience. This kind of trial and error know-how is not accessible in the form of facts and rules. If humans are asked to articulate a set of rules based on their know-how, more often than not it will not accurately reflect their skill. More over, knowledge system does not learn from its experience.

Case-based reasoning (CBR) theory that focuses on solving new problems based on similar past problem solutions seems to be able to eliminate the complex task of maintaining rules and facts through the use of adaptive acquisition of problem-solving techniques [8]. The CYC project developed by Cycorp Inc. aims to assemble and process commonsense knowledge which CBR does not handle. Future expert systems could possibly integrate the commonsense knowledge from CYC project with application-specific modules captured from CBR to enhance their knowledge acquisition and analysis process.

Handling Uncertain Situation

The ability of expert system to derive correct output is often compromised by the lack of precision in rules and inputs. Inference engine is built upon algorithms that manipulate knowledge in the form of decision tree which is not designed to handle uncertainty.

Some expert systems in areas such as linear and nonlinear control, pattern recognition, financial systems and data analysis incorporate fuzzy logic to cope with imprecise rules and inputs. The fuzzy logic in such systems usually uses preset labels to categorize real-time inputs, and utilizes fuzzy inference to calculate numerical conclusions from imprecise rules.

System Integration

Knowledge database are not easily accessible. Expert system tools are often LISP-based, which lacks the ability to integrate with other applications written in traditional languages. Most systems are still not portable among different hardware. All these system integration issues can contribute to higher costs and risks. New system architectures are required to fully integrate external systems and knowledge databases.

Validation

The quality of expert systems is often measured by comparing the results to those derived from human experts. However, there are no clear specifications in validation or verification techniques. How to adequately evaluate an expert system remains an open question, although attempts have been made to utilize pre-established test cases developed by independent experts to verify the performance and reliability of the systems.

Managerial and Organizational Challenges

The success in technical or economic sense of an expert system does not guarantee a high-level of adoption rate or long-term use in business. T. Grandon Gill surveyed expert systems built during the early and mid 1980s [9]. Of all the systems surveyed, the key results were as follows:

- about one-third were being actively used and maintained,
- about one-sixth were still available to users but were not being maintained, and
- about one-half had been abandoned.

The survey also indicated that problems suffered by some of those machines that fell into disuse, had neither technical nor economic basis [9].

Alignment of Technology and Business Strategy

Building expert systems involves substantial amount of investments and efforts. Inconsistency between the organization's business strategy and the technology could ultimately ruin such money and efforts.

Maintenance Cost of Expert Systems

Maintenance cost could be high because expert systems are complicated and might require extensive knowledge of both application domain and development tools in people who develop and maintain the system. A loss of key personnel could result in a whole portion of the project to be re-worked. Projects can be abandoned or delayed if the vulnerabilities in staffing turnover are not resolved.

Expert Systems Make Mistakes

Legal concerns over "expert system makes mistakes" could drive investors and developers away. As mentioned previously, there is little consensus on what testing is necessary to evaluate an expert system's validity, reliability and performance. There are no legal authorities to certify and validate systems. The potential legal and financial liabilities if such systems should go wrong, especially in life-critical systems such as medical diagnosis and air-traffic control could be crippling and astronomical.

Resistance from Users

Unlike traditional computer programs, expert systems perform tasks that an expert performs. This could trigger potential strong opposition and resistance of such technology from users with concerns about expert systems taking their jobs.

These managerial and organizational challenges appear to be very important for expert systems. Failure to take care of such issues could lead to system abandonment or cancellation.

Is "Thinking" Machine Ever Possible

Scientists have long dreamed of autonomous "thinking" systems that are free of human interference. Despite five decades of research, the intuitive intelligence of human beings seems to be still beyond the capabilities of 'reasoning' machines'. While some people deem the invention of a 'thinking machine' a dangerous endeavor that is ultimately doomed to failure, others are more optimistic. Notably, Jeff Hawkins predicted in his award-winning book *On Intelligence* that the mind-machine may come into the world within 10 years [10]. He reasoned, "It took fifty years to go from room-size computers to ones that fit in your pocket. But because we are starting from an advanced technological position, the same transition for intelligent machines should go much faster" [10].

To explore the development potential of expert systems, it is necessary to compare the differences between human and machine "thinking" and discuss the possibilities for the future.

Human's Know-how and Intuitive Intelligence

As mentioned previously, humans learned how to walk through trial and practice. This kind of skill is called “know-how”. Humans acquire such skills via instructions and experience. Human learning is a gradual process, and there appears to be no sudden leaping forward from rule-based knowledge to experience-based know-how [11]. Novices follow rules and instructions while more competent users take into account the situation elements, such as sensing the weakness of the opponent in chess play. Proficient users also recall solutions from past similar experience and can intuitively apply them to the present without first sorting them out by rules or going through rule-based deliberations. Another aspect about human intelligence is that when human experts consciously work on solving problems, they have a different mindset. Taking playing chess as an example, grand masters don't see themselves as manipulating pieces on the board, rather, they deem themselves deeply involved in the world of opportunities, threats, strengths, weaknesses, fears, and hopes [11]. This level of involvement helps the human expert to think differently and to come up with innovative solutions.

The Human Mind

The human brain has many interesting properties. Raj Reddy speculated that there are about one hundred billion neural cells in the human brain and the brain might be performing 200 trillion operations per second if not faster than that [12]. In problem domains such as vision, speech and motor processes, “it is more powerful than 1,000 supercomputers; however, for simple tasks such as multiplication, it is less powerful than a four bit microprocessor” [12]. These processing events taking place in the brain require little conscious effort and awareness on the part of humans and they are extremely difficult for machines to emulate. Conversely, machines can excel in some processes that are difficult if not impossible to a human being. Reddy went on to argue that silicon-based intelligence, if it's ever achieved, might just have different attributes after all.

Hubert Dreyfus and colleagues also questioned if human mind can be materialized into an information processing machine. For example, “if people know that a small box is resting on a large box, they can imagine what would happen if the large box were removed” [11]. A computer, however, “must be given a list of facts about boxes, such as their size, weight, and frictional coefficients, as well as information about how each is affected by various kinds of movements” [11]. Human beings think with images. They use images, not descriptions, to understand and respond to situations. It is totally different from the explicit, step-by-step reasoning that logic machine performs.

What the Future Holds

The thought of making a “thinking” machine could be exciting, controversial, and also intimidating. Ray Kurzweil in his book *The Singularity Is Near: When Humans Transcend Biology* offered some intriguing thoughts of the future such as the nanobots, which are submicro agents injected into human bloodstream [13]. They could be used to monitor and maintain chemical and biological balances. In addition, these nanobots could also specialize in patrolling the brain and downloading every stored neural pattern and synaptic connections from brain cells to a supercomputer system. Such system would, in turn, recreate a software version of the human mind, including memory, emotions, instincts, and thoughts. Moreover, the program could be ported to other machines just like any other software, thinking and acting as “yourself” with immortality [13]!

Clearly, the inner-workings of the human mind are complicated and the challenge of creating machines that could simulate human intellectual abilities is daunting. Moreover, the possibility of endowing computer systems with human-like intelligence still remains debatable.

The controversy extends beyond whether machines can “think” like humans. The “thinking” machine, if it becomes a reality, would undoubtedly have profound social implications and cause irrevocable changes in the very foundation of our society. Nevertheless, the ongoing success of expert systems, as described in previous sections, seems assured.

Social Implications of Expert Systems

Harnessing the brainpower of the experts so that it could be passed along to assist others has always been the main driving force behind the development of expert systems. It is one of the most positive potentials of AI. As Reddy has mentioned, “sharing the knowledge and know-how in the form of information products is the only way

to reduce this ever-widening gap between the have and have-nots" [12]. Expert systems could be a means to share the wealth of knowledge, the most important wealth to share with the disadvantaged. As progresses are made in developing smart AI applications, we might be well on our way to "help the poor, the illiterate, and the disadvantaged of our nation and the world" [12].

While the usage of expert systems is certainly beneficial to our social lives, there are also potential downfalls and detriments that could lead to difficult legal and ethical dilemmas. While it is obviously dangerous and foolhardy to deploy logic machines to be in command of a battle field, what about air-traffic control system that routes airplanes carrying thousands of passengers, or medical diagnosis systems that might assist physicians in life-or-death situations? What happens if these systems give wrong advice? Who should be held accountable? And if these systems can think autonomously and be aware of their own existence, could the incorrect advice be intentional? Should the legal system be extended to deal with machines in the future, like the "Three Laws of Robotics" in Isaac Asimov's fiction?

The development of expert systems also raises the question of who should own the knowledge. Richard L Dunn presented a list of questions on this issue: "Is it you, or your employer? And if a knowledge engineer shows up at your office, how much should you share with him? If he 'drains your brain', are you more, or less, valuable to your company? Can your company take that intelligence and give it or sell it to others without compensating you?" [14]

Certainly copyright and intellectual property laws are well established. Employers are entitled to own the patentable products and copyrightable materials developed during the course of employment. But with regard to the knowledge and experience you have gained via your work, are they subjected to the same regulation? After all, personal knowledge and experience are marketable commodities. That's why your employer hires you. However, supposing that knowledge or experience could somehow be captured and stored in a computer system no longer controlled by you, or that computer system has acquired knowledge and experience from someone or a group of people who have the same expertise as you, will the company still need to employ you?

It's better to be prepared for all these questions before it's too late to answer them

Concluding Remarks

The complexity of human intelligence had been underestimated before, especially in the expert systems field. Technological limitations and managerial challenges still remain in the development of expert systems. However, with the success of neural networks, CASE technology and other state-of-art technologies, the future for expert systems seems bright despite the earlier setbacks.

Moreover, careful consideration must be given to legal and ethical issues that will certainly arise during the course of advancing expert system technology. Should autonomous "thinking" machines ever become reality, our lives as we know it would be forever changed.

AI Winter and its lessons

Overview

The history of Turing Test and Expert systems all showed that people greatly overestimated AI's progress since early days. Before long the hype far exceeded AI's accomplishments. In 1970s, the AI industry entered a period of time described as "*AI Winter*". According to the AI Newsletter, the phrase was borrowed from "*Nuclear winter*", a cold-war theory that "mass use of nuclear weapons would blot out the sun with smoke and dust, causing plunging global temperatures, a frozen Earth, and the extinction of humanity"[Merritt05]. Indeed, during the AI winter, commercial and scientific activities in AI declined dramatically. Arguably AI is still recovering from the winter that lasted nearly two decades. This section describes the trigger and process of AI winter and tries to understand the causes and implications of hype in the history of AI.

Trigger of AI Winter

The onset of the AI winter could be traced to the government's decision to pull back on AI research. The decisions were often attributed to a couple of infamous reports, specifically the Automatic Language Processing Advisory Committee (ALPAC) report by U.S. Government in 1966, and the Lighthill report for the British government in 1973.

The ALPAC report

In the US, one of the main motivations for the funding of AI research was the promise of machine translation (MT). Because of Cold War concerns, the US government was particularly interested in the automatic and instant translation of Russian. In 1954, the first demonstration of MT, the Georgetown-IBM experiment, showed a great promise. The system was by no means complete, consisting only six rules, a 250-item vocabulary and specialized only in Organic Chemistry. But the event received so much public interests that it appeared on the front page of *New York Times* on the next day under the title of "Russian is turned into English by a fast electronic translator". Similar reports appeared in newspapers and magazines across the country in the months to come. It was the most widespread and influential publicity that MT has ever received till today[Hutchins04]. Almost all the reports cited the prediction from Leon Dostert who was in charge of the experiment:

"five, perhaps three, years hence, interlingual meaning conversion by electronic process in important functional areas of several languages may well be an accomplished fact."[Hutchins04]

Stimulated by the great publicity and an active response from the Soviet government, US agencies started to support MT research since June 1956[Hutchins04]. But the progress in the next decade was slow and it was why ALAPC was founded to investigate the causes.

Judging by its impact to MT research, the ALAPC report was often regarded as the best known event in the history of MT. Effectively the report brought an end to the substantial funding in the US. "More significantly, perhaps, was the clear message to the general public and the rest of the scientific community that MT was hopeless"[Hutchins96]. Its impact could be felt in the following some twenty years where an interests in MT became something to keep quiet about.

The ALAPC report had a primary focus on the economic return of the MT research instead of the broader scientific value. The committee believed that the MT research should have "a clear promise of effecting early cost reductions, or substantially improving performance, or meeting an operational need"[Hutchins96]. Particularly, the report studied exclusively on the US government and military needs in the automatic translation of the Russian documents. It did not consider any other potential use of MT systems with any other languages.

The report started by studying the supply and demand of Russian translations in the academia and the government agencies. It was able to establish that there was no shortage of translators. At that time, out of 4000 translators under contract in Joint Publication Research Service, only 300 on average were being used each month[Hutchins96]. It then reasoned that in order to justify MT research, MT must be able to provide crucial improvements in terms of quality, speed and cost. Of course, in none of these aspects was MT able to perform satisfactorily at that time. Among them, the poor quality of MT systems was the major concern. ALAPC argued that

outputs of the MT systems required substantial post-editing in order to be nicely readable by human. The post-editing could take up even more time than translating from scratch by a human translator. Worse yet, the MT results were often misleading and incomplete in the first place! Further, to evaluate the progress in MT, the report compared the output of the latest MT systems with the dozen-year-old Georgetown-IBM experiment. Ironically the output of the Georgetown-IBM system was better, requiring less post-editing. Thus MT appeared to have processed backwards! Many MT researchers refuted this methodology pointing out that it was flawed because Georgetown-IBM experiment was a special purpose system on “doctored” input while the latest MT systems were general purpose ones working on unprepared input. However, it is probably not fair to put all the blame on ALPAC for using the best-known MT “success” as a baseline. There was simply no established general-purpose benchmark when the technology itself was still maturing.

In the conclusion of the report, ALPAC emphasized the little yield from the large government investment. The report echoed the disappointments felt by the general public and correctly pointed out that MT still required much more basic study. Till today MT is still a subject of active research.

The Lighthill report

The “Lighthill report” commonly refers to “Artificial Intelligence: A General Survey” by Professor Sir James Lighthill of Cambridge University in 1973. His review of AI was at the request of Brian Flowers, the head of the British Science Research Council, the main funding body of British university scientific research. The review was to help the council evaluate requests for support in AI research. In the paper, Lighthill offered a pessimistic prognosis for AI, stating that “in no part of the field have discoveries made so far produced the major impact that was then promised”[Lighthill73]. Although Professor Lighthill was a hydro-dynamist with no previous knowledge in AI, the paper created no less impact. It eventually led the British government to end support for AI research in most of the British universities except only four.

Lighthill classifies AI technology into three categories: category A the *advance automation or application*, category C the study of *central nervous system*, and category B the *bridge* in between of A and C which is only worthwhile if it aids A and C. On the basis of such classification, Lighthill concluded that AI research had contributed little to either category A and C, and hence was not worthwhile to proceed.

The main section of the document was titled “Past disappointments”. Lighthill included example of automatic landing system for airplanes, where conventional engineering techniques using radio waves had proven to be more useful than AI methods. Although AI techniques might be useful when landing an aircraft on an uncontrolled environment, Lighthill saw AI was nowhere near a practical solution. He also cited the lack of sophistication of chess programs which only reached an “experienced amateur” level at that time. Other than showing the lack of successful AI applications, Lighthill pointed out that all AI method required substantial knowledge of the subject matter in order to be useful. He was unsatisfied with at the lack of ability to automatically acquire knowledge which made the AI methods not truly intelligent. In addition, Lighthill brought up the issue of “Combinatorial Explosion” to explain the fact that existing AI techniques worked well only in the lab on small domains, but were woefully inadequate in large-scale realistic problem.

Overall, both reports showed great disappointments in the AI technology and had the same deadly effect of cutting government funding in AI research. We know that government funding is critical in the history of all the computing technologies, especially at their infant stage. These reports rang the bell of the approaching AI winter.

The Duration of AI Winter

During AI winter, AI research program had to disguise themselves under different names in order to continue receiving funding. Many somewhat ambiguous names came up during this time that carried strong hint of AI, such as “Machine Learning”, “Informatics”, “Knowledge-based system” and “Pattern recognition”. The re-branding of these disciplines allowed AI to continue to progress in the winter. However, there was less and less perceived advancements under the name of AI which further aggravated the decline in the overall support.

The commercial AI industry probably received a heavier blow in the winter. AI programs intrinsically require a large amount of computing power. In the early 70s, they started to exceed the limit of the common research computers.

The situation was further exacerbated by the nature of LISP symbolic programming language, which was not suited for standard commercial computers that were optimized for assembly and FORTRAN language. So starting in 70s, many companies became offering machines that were specially tailored to the semantics of LISP and were able to run larger AI programs. However, with the onset of AI winter, the industry saw a shift of interest away from the LISP programming language and machines. Coupled with the beginning of “PC revolution”, many LISP companies such as Symbolics, LISP machines Inc, and Liquid Inc. failed as the niche AI market was no longer able to pay the premium for the specialized machines.

For the LISP programming language itself, the popularity originated largely from the academic where fast prototyping and the script-like semantics were very favorable. It did not achieve wide range of success in commercial software development as there were inherent inefficiencies associated with functional programming languages. As a result, many new ideas pioneered by the LISP language, such as garbage collection, dynamic typing, and object-oriented programming went into oblivion along with LISP. Although many of these new ideas were not specific to AI, they did not return until the late 90s.

The progression of AI winter demonstrated a positive feedback system, where symptoms become causes[WikiAIWinter]. The effect of the first triggers in the late 60s continued to be amplified throughout the next decades until activities in the AI eco-system die down in the 80s. A unique self-reinforcing factor in this vicious cycle was the so-called “*AI effect*”. AI effect refers to the tendency for people to discount advances in AI after the fact[AAA]. When some intelligent behavior is achieved in computer, the inner workings are as plain as lines of computer code. The mystery is gone and people quickly dismiss the achievement as mere computations. Michael Kearns explained it as “People subconsciously are trying to preserve themselves some special role in the universe”[AAA]. AI was then often ridiculed as “almost implemented”. Therefore disappointments continued to mount since AI research was always behind its moving target.

Discussion

Disappointments were evident in both ALPAC and Lighthill reports which eventually triggered the downfall of AI. In the ALPAC report, it is clear that a wrong impression about MT was given that automatic translation of good quality was much closer than in reality. As a direct consequence, “sponsorship and funding of MT research in the following years were more liberal (and unquestioning) than they ought to have been”[Hutchins96]. Unfortunately, the early success was really an artificial one: the Georgetown-IBM experiment was “doctored” for that particular occasion. Grammar and vocabulary rules were created specifically to deal with the particular text samples so that the system could appear in the best light. It was widely suspected that Leon Dosert demonstrated the system prematurely to attract funds for further research at Georgetown[Hutchins96]. MT researchers did not treat it to be much more than a first effort, not even a prototype. But neither the press nor the funding agencies took notice of that fact. It can be learnt that success on miniature systems could be most deceiving combined with enthusiastic press. Lighthill’s point of “Combinatorial Explosion” expressed the similar concern on the scalability of AI methods. Indeed, success on specific domain is insufficient as a basis to fund the development of large-scale applications.

Apart from the underestimation of the complexity involved in real-life problems, the disappointments came from the superficial study of AI applications and eagerness to publicize optimistic predictions in the academia. John McCarthy, the initial proponent of AI study, pointed out that much work in AI was not “directed to the identification and study of the intellectual mechanisms”[McCarthy00], but more to generate amazements in the public. AI scientists were also often too quick to claim that a general scheme of intelligent behavior was found which could be applicable for all problem solving[McCarthy00]. Many of such formalisms led to published predictions that computer would achieve certain level of intelligence by certain time. Unfortunately none of them proved to be true. Indeed, the disappointment could have been avoided if researchers had genuine understanding of the inner-workings and deficiencies of their AI algorithms, and exercised caution when expressing hope on any panacea scheme.

Further, a deeper cause of the disappointments probably lies in the lack of understanding of AI in general. The basis of the Lighthill report, i.e. the classification of AI technology, represented a common view that AI was an

applied science derived from biology. Such understanding inevitably led to high expectations on the productivity of AI technology. However, John McCarthy provided a clearer description of AI as:

”studying the structure of information and the structure of problem solving processes independently of application and independently of its realization in animals or humans.”[McCarthy00]

Focusing AI methods on certain applications, or treat AI as simple as simulating biological structure on computer will undoubtedly generate disillusion as the view ignores the broader intellectual challenges in AI research.

Finally many emerging computing technology other than AI also found similar wintery periods in their history. For example, the internet business went through a roller-coaster in 2000. The commonality lies in the hypes that are often associated with the rise and fall of new technologies. Hype Cycle, developed by Gartner group, provide a clear time-line for the adoption of new technologies.

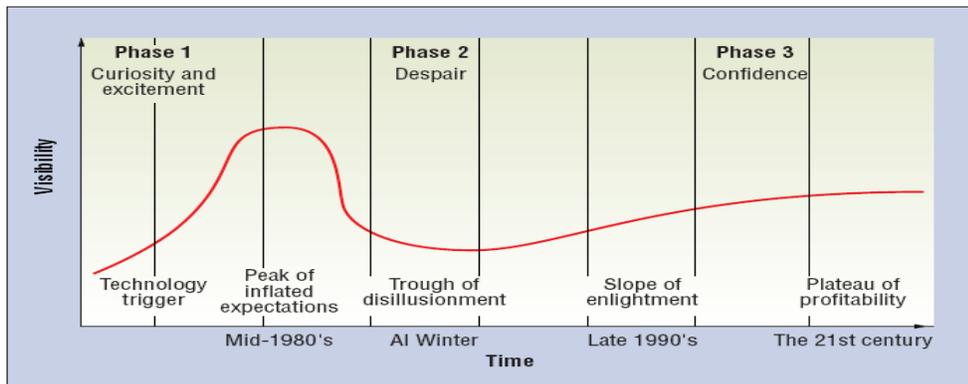


Figure 1. The Hype Cycle and AI winter [Menzies03]

The model illustrated five phases[Gartner]: 1) *Technological Trigger* from events that generates significant press and interests, 2) *Peak of inflated expectations* marked by over-enthusiasm in combination of more failures than success, 3) *Trough of Disillusionment* when the technology was no longer fashionable, 4) *Slope of Enlightenment* for people who persist to understand the benefit and practical applications of the technology, 5) *Plateau of Productivity* as the benefits is widely accepted again. The history of AI leading up to AI winter fits the characteristics of the model extremely well. Remarkably the attendance at the National Conference of Artificial Intelligence matched the pattern of the Hype Cycle closely.

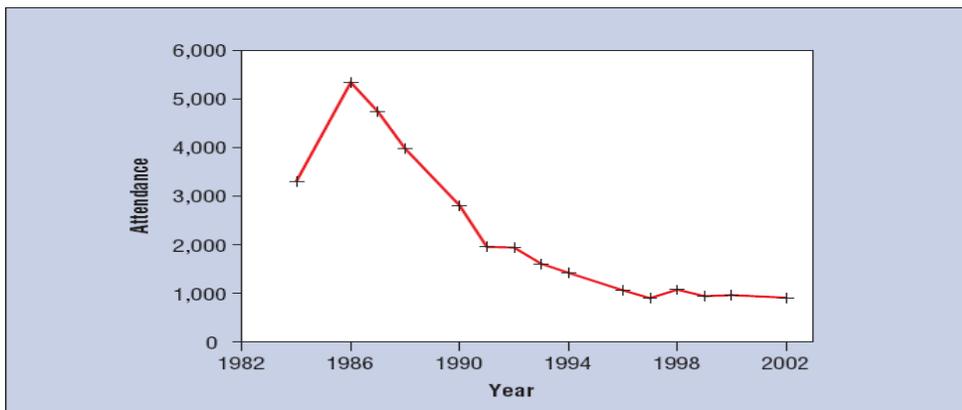


Figure 2. Attendance at the National Conference of Artificial Intelligence[Menzies03]

The prevalence of Hype Cycle suggests that AI winter is not only determined by the nature of the technology, but

also by a common tendency in human cognition. Curiosity, excitement and disappointment are all inherent parts in our exploration of the unknown. Without the initial great excitement and thus the subsequent heavy support, AI technology might not have taken off in the first place. Without the cooling-down at the peak of the hype, people might not be able to appreciate the real challenges in the field and generate new insights for future research. In a sense, some level of hype may be catalytic to the technology. However, the history of AI winter reminds us that uncontrolled hype, especially when exaggerated by the press, would wreck havoc.

Conclusion

Overall, the study of AI winter highlighted a few lessons:

Small-scale success in AI was deceptive. The complexity of AI implies that many issues will only be encountered and solved on large-scale problems in real life.

AI is a subject with broad intellectual challenges of its own. It is not limited to specific applications or certain biological structures. It requires combined basic research in cognition, statistics, algorithms, linguistics, neurosciences and much more.

Hype is a double-edge sword. It boosted the rise of AI initially, but did great harm in the end. It is a common responsibility of researchers, funders and the public to restrain it so that the AI winter will not come again.

Japan's Fifth Generation Computer System project

Amid the AI Winter in the United States an epic attempt to realize the 'AI dream' was underway in Japan in the form of the Fifth Generation Computer System (FGCS) project during the 1980s. For years Japan (and the rest of the world) lagged behind the United States in technology, and for the most part played follow-the-leader. In 1978 Japan's Ministry of International Trade and Industry (MITI) commissioned a study of what the future would hold for computers, and three years later attempted to construct fifth generation computers – creating what project heads described as an 'epochal' leap in computer technology, in order to give Japan the technological lead for years to come. This new generation of machines would not be built on standard microprocessors, but multi-processor machines specializing in logic programming. The bet was that these high-power logic machines would catalyze the world of information processing and realize artificial intelligence.

After 10 years of research and more than a billion dollars in funding we still don't have truly intelligent computers. Most of the research done didn't push the state of the art and where it did it was only in areas specific to the scale and scope of FGCS applications. However the story of the FGCS project is an interesting one, well worth examining. Both the grand vision of intelligent machines and epic failure of the project are worthy of investigation.

Motivations and influences

The motivation for Japan to build FGCS machines was a simple one, in order to create new technology and take the lead in the computer industry. (Much like Japan's successful push into the automotive industry.) The study concluded that 'knowledge processing' would be the future of computers, and that if Japan played an active role in developing specialized machines for this task the country could control this information for huge financial gains[FeigenMcCord83]. This view was widespread and was summed up by the IOTC's (discussed later) director and visionary of the FGCS project, Kazuhiro Fuchi:

"The route to knowledge information processing represents a practical philosophy and an inevitable direction for development of information processing technology. The question is ... whether to stand still or proceed, as there are no other plans to choose from."

More than pushing ahead to give Japan a lead in the technology industry, the FGCS was to push computing technology into an entirely new area: knowledge processing. One of the key goals of the FGCS project was to produce KIPS or Knowledge Information Processing Systems. The idea is that AI wouldn't be done through fancy algorithms or learning, but iterating over massive knowledge bases and inferring things from that data. Specifically hardware based syllogism engines. For example, hardware to deduce: a dove is a type of bird, and all birds fly, therefore a dove can fly.

Logic programming had been well studied in the US during 60s and 70s, and while effective, they weren't able to make machines really that intelligent outside of an academic environment. The FGCS project supposed that with specialized hardware capable of inferences across massive databases that enough relevant knowledge could be used to generate meaningful data. With a scale as large as the FGCS project had hoped, a critical mass of knowledge could be contained in a database file system and processed instantaneously.

Initial plans had specified machines capable of making a million logical inferences per second. Project heads had estimated that capability as powerful enough to translate natural speech, prove theorems, and other intelligent activities.

In order to achieve this was MITI formed the Institute for New Generation Computer Technology (ICOT). The ICOT then embarked on a 10 year journey to realize a FGCS. The first three years would be for initial research, followed by four years of production/completing minor modules, and a final three years to develop prototype machines. MITI, overseeing the IOCT would negotiate the budget with Japan's Ministry of Finance. The budget for the first three years was to include salaries for workers 'on loan' from Japanese technical firms, which were expected to invest in the FGCS project after it had demonstrated results and also to loan more workers to help out in the national effort.

Progress made on FGCS

Successes

With so much money, effort, and national pride riding on the success of the FGCS project it is difficult to imagine it not being successful. The FGCS project did produce both sequential and parallel knowledge processing machines. Near the end of 1980 parallel KIPS machines, such as the 256-processor PIM/m, were capable of producing over half a million of logical inference per second. [Nakashima92] [Kurozumi92].

Failures

Perhaps the failure of the FGCS project was best summarized in a report on the conference preceding the completion of the intermediary stage of the FGCS project:

“Most of the applications presented at the conference were interesting because they were ‘X done in logics programming’ – not because they were ‘X done better than before.’ The hope of course is that the final computer will be fast enough to run programs which are infeasible on normal computers. [Nielson88]

The main issue was the failure to realize ‘intelligent’ software. The IOTC had researchers working on difficult problems such as natural language processing, automatic theorem proving, and even a program capable of playing the game of Go [SeilchiTaki91]. However none of these areas experienced significant breakthroughs due to new hardware capabilities. Even with ultra-powerful logic machines teaching software to be intelligent was too daunting of a task. Similar to Type-A vs. Type-B chess programs, in some situations you can convert computational power into increased perceived intelligence. However, the bottleneck is typically understanding how to teach the computer to think – by abstracting the problem space.

Another problem with the project was the assumption that parallel logic chips would be required in order to perform advanced computations. Microprocessor technology advanced steadily during the 10 years of the FGCS project. Although FGCS hardware was superior in pure logic-programming, commodity hardware was competitive, especially compared to low-end single-processor KIPS systems.

Lessons learned

Despite the high hopes the FGCS didn’t realize the huge advances expected. Clearly some issues such as an emphasis on parallelization were just ahead of their time, while others like massive knowledge databases and pure logic-based programming didn’t pan out. It is easy to disregard the FGCS as a failure in history, but we can learn some powerful lessons about government mandated research and a push towards intelligent systems.

Government sponsored computer research has worked well for the United States; in fact most of the history of early computers is owed directly to US military spending. However what separated Japan in the FGCS project from other US-based research efforts such as DARPA was that Japan focused all their resources on the single endeavor of intelligent machines. While a focused effort in a single area / technology can lead to larger rewards it is difficult to predict if that technology will be successful.

With Japan’s goal of becoming a world-leader in computer technology it would be impossible to ignite a revolution without taking a risk. So it is difficult to see this as an error. If the project was successful and did find ways to dramatically increase the intelligence of software through customized hardware then the world of computing as we know it would have been forever changed. Just as US-based researchers in the 50s and 60s theorized that machines would be capable of passing a Turing Test ‘soon’, the issue at root was underestimating the difficulty of artificial intelligence.

The most noteworthy aspect of the FGCS project was the gamble made on intelligent computers. There was never a debate or hesitation on whether a machine can be intelligent, just a bold attempt to realize that vision. Perhaps the lesson to learn is that such a view is unfounded, and the failure of legions of Japanese researchers should provide skepticism about the practicality of AI. More likely is that the bottleneck in artificial intelligence development is software, and not hardware. Perhaps another FGCS project is still essential for the development of true AI, but probably only after we develop intelligent machines which are just too slow for practical use.

Conclusion

In each of the sections above we've covered different threads of the history of AI that carry similar themes. Overall we see a field pulled in two directions – one toward short term practical applications and the other toward the grander issues that spill over into the very definition of what it means to be intelligent.

From one perspective we have made great progress – AI is a fairly well defined field which has grown over the last 50 years and has helped solve many problems, be it adaptive spam blocking, image/voice recognition, high performance searching etc. On the other hand, the things the originators of AI like Turing and McCarthy set out to do seem just as far away now as they were back then.

In AI, the hard problems have not been solved and progress has been slow, be it due to a loss in funding, lack of interest by researchers or a realization that the problems only become more difficult the better one understands them. These difficulties themselves could have helped bring on the AI winter scenario - why bother throwing money at the research when there are more practical applications waiting for resources. So after 50 years or work, there is no computer that can pass the Turing test, there was no mass replacement of experts with software systems, and computers can't beat humans at some simple but strategic games like Go.

This of course doesn't mean that we give up thinking and trying, but just that we refine our approach. Over the years we have learned that having a massive knowledge base isn't enough, nor is a million logical inferences a second. The hard problem in the field of AI is finding a way to teach a machine to think, but in order to articulate 'thought' in a way current computers can understand we must first understand thinking and intelligence ourselves. Until then, we will go on creating chess programs which rely on brute force, expert systems which fail to notice the obvious, and converse with programs which just aren't that interesting to talk to.

References

Introduction, History of AI applied to Chess, and Japan's FGCS

- [Turing50] Turing, Alan. 1950. Computing Machinery and Intelligence. *Mind* 49, 433 – 460.
- [Bush45] Bush, Vannevar. 1945. As We May Think. *The Atlantic Monthly*. July 1945.
- [Searle80] Searle, John. 1980. Minds, Brains, and Programs. *Behavioral and Brain Sciences* 3, 417-424.
- [Shaw01] Shaw, Michael; Subramaniam, Chandrasekar; Tan, Gek Woo; and Welge, Michael. Knowledge management and data mining for marketing. *Decision Support Systems*. Vol. 31, Issue 1. 2001.
- [FeigenMcCord83] Feigenbaum, Edward and McCorduck, Pamela. *The Fifth Generation*. Addison-Wesley. 1983.
- [SchaefferPlatt97] Kasparov versus Deep Blue: The Re-match. *ICAA Journal*, vol. 20, no. 2. 1997. pp.95 – 102
- [CambellHsuHoane02] Cambell, Murray; Hoane, Joseph; and Hsu, Feng-hsiung. Deep Blue. *Artificial Intelligence*. Volume 134, issue 1-2. 2002. 57 – 83.
- [Kurozumi92] Kruozumi, Takashi. Overview of the 10 years of the FGCS project. *Proceedings of the international conference on fifth generation computer systems*. 1992.
- [Nakashima92] Nakashima, Hiroshi; Kondo, Seiichi; Inamura, Yu; Onishi, Satoshi; Masuda, Kanae. Architecture and Implementation of PIM/m. *Proceedings of the International Conference on Fifth Generation Computer Systems*. 1992.
- [SeiIchiTaki91] Sei, Shinichi; Oki, Hiroaki; and Akaosugi, Takashi. Experimental version of parallel computer Go-playing system GOG. Technical Report TR718, ICOT, Japan. 1991.
- [Nielsen88] Nielsen, Jakob. Trip report for the international conference on Fifth Generation Computer Systems. 1988. Available online at: <http://www.useit.com/papers/tripreports/fifthgeneration.html>
- [Shannon50] Shannon, Claude. *Philosophical Magazine*. Ser. 7. 1950.
- [WikiChess] Wikipedia entry on Computer Chess. http://en.wikipedia.org/wiki/Computer_chess
- [Müller00] Müller, Martin. Not like other games why tree search in Go is different. *Proceedings of Fifth Joint Conference on Information Sciences*, pp 974
- [DoshayMcDowel05] Doshay, David and McDowell, Charlie. Suggo: A computer badku program. *International Conference on Baduk*. October 2005.
- [Bouzy03] Bouzy, Bruno. Mathematical morphology applied to computer go. *International Journal of Pattern Recognition and Artificial Intelligence*. Vol 17. Pp: 257-268. 2003.

Turing Test

- [1] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 2003
- [2] Alan Turing, "Computing Machinery and Intelligence", 1950, <http://www.abelard.org/turpap/turpap.htm>
- [3] Wikipedia, Alan Turing, http://en.wikipedia.org/wiki/Alan_Turing
- [4] Alan Turing, "On Computable Numbers with an Application to the Entscheidungsproblem", 1936, <http://www.abelard.org/turpap2/tp2-ie.asp>

- [5] Simon Singh, *The Code Book*, 1999
- [6] Ray Kurzweil, *The Singularity is Near*, 2005
- [7] Stewart Brand, *The Media Lab*, 1988
- [8] Loebner Prize <http://www.loebner.net/Prizef/loebner-prize.html>
- [9] Long Bets, <http://www.longbets.org/172>

AI Winter

- [Merritt05] Merritt, Dennis. AI Newsletter, May 2005, http://www.ainewsletter.com/newsletters/aix_0501.htm
- [Hutchins04] Hutchins, John. The Georgetown-IBM experiment demonstrated in January 1954, Sep 2004, <http://ourworld.compuserve.com/homepages/WJHutchins/AMTA-2004.pdf>
- [Hutchins96] Hutchins, John. ALPAC: the (in)famous report, June 1996, *MT news International* 14, 9-12.
- [Lighthill73] Sir Lighthill, James. "Artificial Intelligence: A General Survey", *Artificial Intelligence: A paper symposium*, Science Research Council, 1973.
- [WikiAIWinter] Wikipedia, AI winter, http://en.wikipedia.org/wiki/AI_winter
- [AAAI] American Association of Artificial Intelligence (AAAI), AI Effect, <http://www.aaai.org/AITopics/html/aieffect.html>
- [McCarthy00] McCarthy, John. Review of "Artificial intelligence: A General Survey", June 2000, <http://www-formal.stanford.edu/jmc/reviews/lighthill/lighthill.html>
- [Menzie03] Menzie, Tim. 21st-Century AI: Proud, Not Smug, 2003, *IEEE Intelligent Systems*
- [Gartner] Gartner Group, Understanding HypeCycle, <http://www.gartner.com/pages/story.php.id.8795.s.8.jsp>

Expert Systems

- [1]. Expert systems in medicine: academic illusion or real power? Metaxiotis, K. S. and Samouilidis, J.-E. 2, 2000, *Information Management & Computer Security*, Vol. 8, p. 75.
- [2]. Report on a general problem-solving program. Newell, A., Shaw, J.C. and Simon, H.A. 1959. *Proceedings of the International Conference on Information Processing*. pp. 256-264.
- [3]. Recursive Functions of Symbolic Expressions and Their Computation by Machine (Part I). McCarthy, John. 1960, *Communications of the ACM*, pp. 10-13.
- [4]. The overselling of expert systems. Martins, G. 1984, *Datamation*, Vol. 5, pp. 76-80.
- [5]. The payoff from expert systems. Enslow, B. January/February, 1989, *Across the Board*, p. 54.
- [6]. AI: industry's new brain child. Cook, B. April, 1991, *Industry Week*, p. 57.
- [7]. The reality and future of expert systems. Goel, A. Winter, 1994, *Information Systems Management*, Vol. 11, p. 1.

- [8]. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. Aamodt, Agnar and Plaza, Enric. 1, 1994, Artificial Intelligence Communications, Vol. 7, pp. 39-52.
- [9]. MIS Quarterly. Gill, T Grandon. 1, 1995, Minneapolis, Vol. 19, p. 51.
- [10]. Hawkins, Jeff and Blakeslee, Sandra. On Intelligence. s.l. : Times Books, 2004.
- [11]. Why Expert Systems Do Not Exhibit Expertise. Dreyfus, Hubert and Stuart. Summer, 1986, IEEE Expert.
- [12]. Foundations and Grand Challenges of Artificial Intelligence. Reddy, R. Winter, 1988, AI Magazine, p. 9.
- [13]. Kurzweil, Ray. The Singularity Is Near: When Humans Transcend Biology. s.l. : Viking Adult, 2005.
- [14]. Who owns your brain? Dunn, Richard L. 9, 1998, Plant Engineering, Vol. 52, p. 10.