HAL ORIGINS

Taking our scientific and technological perspective a step farther, we can then ask: What else does this lack of appreciation of software imply, and how was it manifested in the film? Kubrick and Clarke -- and indeed all but a few computer visionaries in the 1960s -- failed to understand the important and unique nature of software: that it is general purpose, infinitely malleable, and can be divorced from hardware. This lack of understanding helps explain the excessive number of control buttons we see in the film, especially in the pods. Currently, jetliners and fighter jets are equipped with numerous computer screens that display different types of information and replace mechanical buttons. One good computer screen with windows and software buttons would have sufficed for *Discovery*.
But *2001* was made before the Macintosh computer interface was developed, so we can't blame Kubrick and

Clarke for overlooking this important trend, nor for the click of the (analog) shutter of a bulky camera where software (and smaller, digital cameras) would be used today. And, as Stephen Wolfram points out, the snippets of computer code visible on HAL's screens are reminiscent of BASIC or Fortran, popular computer languages at the time the film was made. Presumably, the software written for HAL's hardware is much less intuitive (i.e., easy for humans to understand) than languages developed since then.

In short, our technical analysis of the software issues suggested by the film draws together such details as HAL's birthday, the click of a camera, and the plethora of buttons, as well as snippets of code on the screens. It's hard to imagine how a traditional cinematic or literary analysis could shed such light or deepen our understanding of the film in this way.

In fact, *2001* is suffused -- explicitly and implicitly -- with the motif of birthdays: everything from the "dawn of humanity" and the explicit date when HAL "became operational," to the birthday of Heywood Floyd's daughter "Squirt" (played by Kubrick's daughter Vivian), to Frank's birthday (complete with his parents singing "Happy Birthday" via a radio transmission delayed by the immense distances of inter-planetary space), to the final scene -- the birth of a star child. All this, of course, occurs at the birth of the millennium.

As the chapters in this book illustrate, the film ranges widely over issues that are still salient in computer science (and in space travel), and presents a rich array of "predictions" -- though Clarke prefers to consider them

"visions." It is a testament to the thoroughness of its makers that the film remains as vital and moving today as when it came out. It is appropriate that we analyze these points now, at yet another birthday, the time the novel says HAL "became operational": January 12, 1997.

The overarching technical issue associated with HAL -- and the one that captures the imagination, is his artificial intelligence (AI). *2001* is, in essence, a meditation on the evolution of intelligence, from the monolith-inspired development of tools, through HAL's artificial intelligence, up to the ultimate (and deliberately mysterious) stage of the star child.

How well does Kubrick and Clarke's vision stand up to the reality of 1997? Let's state the obvious: HAL doesn't exist, and there is no chance that some miraculous change in funding or insight will yield AI at the level portrayed in HAL by the year 2001. We have learned that artificial intelligence -- a notably hazy matter we don't even have a good definition for -- is one of the most profoundly difficult problems in science -- on a par with going to the moon, identifying the fundamental constituents of matter, and unlocking the puzzle of life. But we've gone to the moon, and we seem to be on the way to understanding these other mysteries. *Why* don't we have AI?

Marvin Minsky (who, incidentally, nearly lost his life consulting on *2001*!) argues (in chapter 2) that the field made such good progress in its early days that researchers became overconfident and moved on prematurely to more immediate or practical problems --

for example, chess and speech recognition. They left undone the central work of understanding the general computational principles -- learning, reasoning and creativity -- that underlie intelligence. Without these, he believes, we will end up with a growing collection of dumb experts and will never achieve AI.

Stephen Wolfram (in chapter 15) agrees that we have overlooked deep principles but thinks the missing pieces lie in the domain of complex systems -- a field he helped launch -- where simple elements can interact to produce unexpectedly complex behavior. Just as the simple laws of physics applied to simple water molecules can give rise to the wonderfully complex vortex flow of a stream around a rock, so the relatively simple rules governing nerve cells (neurons) may lead to a complex cognitive system. His insight, though, is that we shouldn't be thinking so much in terms of mathematics (i.e., equations) as, instead, algorithms (i.e., computer programs).

Ray Kurzweil (chapter 7), on the other hand, says that we already know how to achieve AI -- just reverse engineer a brain! In a few decades, he suggests, we may be able to scan an entire human brain, down to the level of nerve cells and their interconnections. We need then merely (!) encode all that information into a computer to make a virtual brain every bit as intelligent as a human brain that is its model.

However, for Doug Lenat (chapter 9), there is no silver bullet for achieving AI. We first need to encode an enormous amount of common-sense knowledge into a computer through a laborious process of hand entering

information gleaned from a wide range of sources -- such as encyclopedias. With such a "primed knowledge pump," a computer might be able to learn more by reading books and interacting with its environment, scientists and other experts, and, perhaps, other developing computers. That, at least, is the theory. Daniel Dennett (in chapter 16) stresses the need for a computer to explore the world, and learn from its interactions.

With most science fiction films, the more science you understand the less you admire the film or respect its makers. An evil interstellar spaceship careens across the screen. The hero's ship fires off a laser blast, demolishing the enemy ship -- the audience cheers at the explosion. But why is the laser beam visible? There is nothing in space to scatter the light back to the viewer. And what slowed the beam a billionfold to render its advance toward the enemy ship perceptible? Why, after the moment of the explosion, does the debris remain centered in the screen instead of continuing forward as dictated by the laws of inertia? What could possibly drag and slow down the expanding debris (and cause the smoke to billow) in the vacuum of outer space? Note too the graceful, falling curve of the debris. Have the cinematographers forgotten that there is no gravity -- no "downward" -- in outer space? Of course, the scene is accompanied by the obligatory deafening boom. But isn't outer space eternally silent? And even if there were some magical way to hear the explosion, doesn't light travel faster than sound? Shouldn't we see the explosion long before we *hear* it, just as we do with lighting and thunder? Finally, isn't all this moot? Shouldn't the enemy ship be

invisible anyway, as there are no nearby stars to provide illumination?

But with other, less numerous films, the more science you know the more you appreciate a film and esteem its makers. *2001* is, of course, the premier example of this phenomenon. Director Stanley Kubrick and author Arthur C. Clarke consulted scientists in universities and industry and at NASA in their effort to portray correctly the technology of future space travel. They tried to be plausible as well as visionary. Every detail -- from the design of the space ship, the timing of the mission, and the technical lingo to the typography on the computer screens and the space stewardesses' hats (bubble-shaped and padded to cushion bumps in the zero gravity of space travel) -- was carefully considered in light of the then-current technology and informed predictions. The film, which has been used in the training of NASA astronauts, doesn't look dated even though thirty years have passed since its release.

We acknowledge, of course, that science fantasy is a literary or cinematic genre and need not get the science right to succeed as art. Indeed, *2001* succeeds on the strength and boldness of its vision, the profundity of its central thesis, and the clarity and unsurpassed mastery of its cinematic technique. Nevertheless, the incorporation of science and technology -- *real* science and technology -- is a vital part of its success, a part that has not, until now, received adequate consideration.

It is widely believed that architects design their best buildings when they are confronted by obstacles and challenges. Think of Frank Lloyd Wright's Fallingwater house, nestled among rock outcrops and perched over a

stream. So too, faithfulness to scientific constraints led Kubrick and Clarke to create especially brilliant cinematic solutions. Would a lesser director have obeyed the laws of physics and portrayed Frank's murder or Dave's reentry through the emergency airlock in *silence*?

Virtually all the film's rare departures from scientific veracity were deliberate compromises by Kubrick and Clarke. For instance, even though *Discovery's* speed is extremely high by terrestrial standards, the ship would not appear to move relative to the stars. The filmmakers were well aware of this. But when test sequences showing the stars motionless in the background made the ship look too static, they compromised and introduced a slow drift of the stars. (This solution was immensely challenging technically and required meticulous microalignment of many separate sequences.) Similarly, although in reality half of *Discovery* would have been invisible to any cosmic viewer -- because it would receive no light from the sun -- Clarke and Kubrick, realizing that such a half-visible ship would distract the audience, reluctantly illuminated all of it. In short, the filmmakers knew and cared about getting the science right and made as few artistic exceptions to accuracy as possible. Their care extended, too, to HAL, the central and surely the most memorable character in the film.

But before we turn to HAL, we might ask: Why analyze the science in the film at all? Why not just consider the film as art, with its own conventions and logic? We believe that, just as an art history book can deepen our understanding of a painting or sculpture, scientific analyses can lead to a richer aesthetic experience of the film. We seek to see the film from an additional, fresh perspective -- not to diminish its art, but to appreciate it

more fully. Such a scientific analysis also provides those of us who are not working scientists an opportunity to learn more about the research going on in real laboratories and about the recent history of science -- computer science in particular.

Consider, for example, how such a perspective augments a traditional cinematic view in relation to the issue of software. In Clarke's novel (adapted from the screenplay), HAL's birthday is January 12, 1997, whereas in Kubrick's passage in the screenplay it is 1992. Why the difference? A traditional analysis (centered on character development, plot devices, and so forth) might suggest that Kubrick wanted to make HAL's life somewhat longer in order to make his death more poignant. But from our -- and Clarke's -- technological perspective, the 1992 date is implausible; there is simply no need for such a long history. Who would use a nine-year-old computer on the most technologically sophisticated and challenging adventure in the history of mankind? In fact, HAL's software could be downloaded onto a 1997-vintage computer in a few days (and wouldn't require inserting millions of floppy disks). Consequently, there is no scientific reason for the early birthday.

The contributors to this anthology share a general consensus that nothing in principle prevents us from creating artificial intelligence -- not quantum gravity, not some secret mystical *élan vital*. Other than that, there are numerous disagreements. Naturally, this is healthy for the field -- let a thousand flowers bloom! I confess that my own background and sentiments are close to Dennett's and Lenat's, although I might stress the role of learning somewhat differently than they do. It

will be hard indeed to go from low levels and up through the enormous range of levels to create the exquisitely organized systems needed for language or scene analysis. In my own area of expertise, pattern recognition (admittedly a small part of the AI puzzle, though an essential one), we seem to understand the central principles. It is *applying* them that has proven extremely difficult. Even if Kurzweil is right in believing that we can someday reverse engineer a brain, such an effort would tell us little about *how* the brain learns or represents information.

The fact that we have not achieved AI (for whatever reason) should not, however, blind us to the fact that in some domains we have met and even surpassed the vision of a HAL. As David Kuck (chapter 3) points out, recent advances in computer design and the spectacular and continuous rate of improvement in hardware summarized by Moore's law tell us that we could soon build a computer the size and power of HAL. Ravishankar Iyer, reviewing the progress in computer reliability, shows how we could make the hardware of a massive computer like HAL reliable, even for a prolonged space mission. Alas, the techniques for insuring reliable hardware are more effective than those for software, and this is surely an imposing, but not insurmountable, barrier. In principle, there seems to be nothing to prevent us from making a large computer that is fault-tolerant. Does this sound too much like the hubris of HAL and his designers? As Frank points out in that context, making a computer with no errors "sounds a little like famous last words."

Still, in limited-application domains, we *have* made steady improvements. We have computer chess systems

that beat all but a few dozen human grandmasters, and they are improving every year. It seems all but certain that the world's best chess player will soon be a computer -- perhaps even by the year 2001. In chapter 5, Murray Campbell, a member of IBM's Deep Blue team that challenged Garry Kasparov in February of 1996, analyzes *2001's* chess scene in fascinating detail. Campbell also reflects on the changing public perception of chess machines, as illustrated in the film.

We have made several important strides toward automated speech recognition, especially in the initial stages of transcribing raw sound into phonemes. Kurzweil summarizes that progress and applies his own commercial VOICE speech-recognition system to the *2001* soundtrack. He finds that the system's simple phonological transcription is good, and can be particularly accurate when restricted to just two talkers (e.g., Frank and Dave in the quiet spaceship). General speech recognition, however, relies very heavily on semantics, common sense, context and world knowledge. (For example, how do we give a computer the ability to distinguish among such homonyms as *their*, *they're*, and *there*?) We are still far from solving problems like this one.

Similarly, in my own research (chapter 11), I have developed speechreading (lipreading) systems that use both sight and sound. These systems outperform purely acoustical speech recognizers in noisy rooms. Alas, no current system even remotely approaches HAL's proficiency at speechreading in silence. In fact, the lipreading scene in the pod is the only one Clarke thought was technologically implausible for the year 2001. Automatic speech recognition and speechreading

will always be limited by the problems of representing semantics, common sense, and world knowledge -- profoundly difficult issues that will occupy science for many decades to come.

As humans, we take our faculties of vision for granted, but making computers see has proven to be extremely difficult. In chapter 10, Azriel Rosenfeld discusses some of the successes of research in "early" vision, such as edge and motion detection, face tracking, and the recognition of emotions (which he illustrates with images from *2001*). Full vision, however, would include the ability to analyze scenes. Imagine, for example, a computer that could look at an arbitrary scene -- anything from a sunset over a fishing village to Grand Central Station at rush hour -- and produce a verbal description. This is a problem of overwhelming difficulty, relying as it does on finding solutions to both vision and language and then integrating them. I suspect that scene analysis will be one of the last cognitive tasks to be performed well by computers.

Other capabilities have proven remarkably difficult to develop, including one that Clarke wasn't sure we could solve by 2001: making a computer produce natural-sounding speech. In chapter 6, Joe Olive reviews the development of computer speech generation, such as the systems needed to convert text to speech for the blind. Although these experimental systems work adequately for short utterances or single words, with sentences (let alone entire speeches) they cannot convey the human subtleties of stress and intonation. A convincing artificial speaking system would require the system to

understand what it is saying -- again, an extremely hard and unsolved problem.

Reading Roger Schank's discussion of language in chapter 8 is like seeing the Wizard of Oz behind the curtain. With a few simple tricks, we could duplicate some of HAL's linguistic performance. For instance, it would be a fairly straightforward task to record a large number of canned stories and have HAL play them back when he hears appropriate "trigger" questions. Such a program might even persuade a casual or unsuspecting person that HAL understood language. Indeed, a noted computer program (ELIZA) around the time of *2001's* release, sought to mimic a Rogerian therapist; in limited dialogues it convinced naive users that they were conversing with a real person. Such demonstrations tell us more about the novelty of computer discourse of the time and the gullibility of users than they do about true machine intelligence. ELIZA and its descendants are a far, far cry from true language understanding and general intelligence.

In the film, HAL makes several plans: to test whether the AE35 unit was in fact faulty, to navigate the ship, to search for extraterrestrial life around Jupiter, and to kill the crew. Such planning, Dave Wilkins points out in chapter 14, requires the identification of subgoals, anticipating obstacles, retracing steps, and so on. His review of the progress in planning by computer demonstrates the difficulty of solving even apparently simple problems in a rarefied and idealized world of stacking blocks. The bottom line is that planning is *hard*! HAL was not particularly good at it, and neither are current computer systems.

As the characters in the film admit, it is hard to tell whether HAL's emotions are programmed in to make him easy to talk to, or whether they are genuine. Rosalind Picard (chapter 13) discusses HAL's emotions -- his pride, anger, fear, paranoia, concern -- as well as his ability to recognize emotion in crew members -- and how current research approaches similar problems. If emotions are essential for computer cognition, as she and Minsky and Norman argue, how will we deal with such "affective" computers? Would you trust an affective computer with your spreadsheets?

Don Norman (in chapter 12) underscores the importance of emotion, and points to a notion of machine (and human) intelligence much broader than the kind of logical intellect that preoccupies the field. He looks at what it might be like to live in space and work with computers for extended periods. His discussion stresses the need to take into consideration the "softer" aspects of cognition, those related to emotions, making mistakes, and so forth.

It's interesting that Kubrick and Clarke seem to want us to have stronger feelings toward HAL than we do toward the crew. HAL is the only one in the film to show emotions: "I'm afraid. I'm afraid. I'm losing my mind. I can feel it. I can feel it." In contrast, the dull, robotlike astronauts sleepwalk through boring meetings and chat about ham sandwiches. (Three other characters are known solely through the trace of their biological functions during hibernation.) When the BBC announcer, Mr. Amer, greets the crew, they mumble indistinctly in response. HAL, however, answers with clarity, animation, and interest: "Everything is going extremely well." By the middle of the movie, we have

grown accustomed to HAL and accept the fact that he has more personality than the crew. While we may be shocked and surprised at the death of Frank, HAL's plaintive "I'm afraid .... I'm afraid" evokes our sympathy.

In fact, the audience's sympathy (or anger or both) toward HAL makes more poignant a number of ethical questions. Does HAL murder Frank and the three hibernating crewmen? If so, who (or what) should be punished? Is it immoral to disconnect HAL (without a trial!)? Daniel Dennett argues that higher-order intentionality and responsibility are necessary conditions for moral responsibility -- and hence blame -- and that HAL exhibits such intentionality. After all, he tells Dave "I want to help you" (though, of course he says this to try to stop Dave from dismantling him).

A broad question touched on throughout this book is whether we should try to make computers intelligent by mimicking a human brain or, instead, exploit their particular strengths -- such as rapid search and large memories. So far, different domains have tried different approaches. Researchers in computer chess, for instance, began by trying to reproduce the methods of human grandmasters -- in particular recognizing key configurations of pieces on the board -- but found that this approach quickly led to such difficulties as determining and representing the key properties of the arrangements. Massive and rapid searches of possible sequences of moves have proven more successful. This approach is used for Deep Blue, which has rapid search capacities that are distinctly unlike those of human grandmasters (although grandmaster Garry Kasparov has commented that Deep Blue plays like an intelligent

human). Likewise, the best computer speechreading systems operate not at all like humans. Both of these applications are limited and -- as valuable and interesting as they may be -- do not lead us toward true AI.

Kurzweil's brain-scan approach is the most extreme of the duplicate-a-human-brain approaches. Certain cognitive abilities -- in particular, language and our species-specific common sense -- are distinctive to human beings. There are strong arguments for trying to copy these capacities in computers. While this would not require duplicating a brain to the level of nerve cells, it would, presumably, involve identifying the specific methods we use to carry out myriad tasks.

Another theme suggested by *2001* is society's perception of computers, contemporaneous and future. Were public fears about computers in the 1960s borne out? The fact that Frank takes his loss to HAL at chess without the slightest surprise reflects an attitude that is radically different from the public's perception in that decade when the thought of a computer becoming the world chess champion evoked anger and hostility.

As we approach 2001, we might ask why we have not matched the dream of making a HAL. The reasons are instructive. In broad overview, we have met, and surpassed, the vision of HAL in those domains -- speech, hardware, planning, chess -- that can be narrowly defined and easily specified. But in domains such as language understanding and common sense, which are basically limitless in their possibilities and hard to specify, we fall far short. Perhaps too we need to ask whether, as a culture, we are willing to support the

undertaking of producing artificial intelligence.

The vision that produced HAL was clearly that "bigger is better." In fact, one of the major trends that the filmmakers did not foresee was that there was "room at the bottom"; that is, that computers would become smaller and smaller and evolve into doorknobs and pocket pagers. There are no personal computers or personal digital assistants in *2001*; Frank and Dave take notes with pen on paper tablets on clipboards. The role of networks is not explored, although it is unclear how they would have been incorporated into the story even if the creators had been fantastically prescient enough to predict them.

Yet another, and final theme to consider is the influence of science fantasy on budding scientists. In an earlier era, Buck Rogers films -- as crude as they were -- fired the imaginations of future employees of NASA and aerospace corporations. Implicit, and occasionally explicit, throughout this book is the fact that nearly all the contributors were deeply influenced by *2001* when it was released in 1968. I was myself introduced to the notion that a computer might someday be able to lipread by that famous scene in the pod, and I have spent years trying to devise computer lipreading systems. In that sense, many scientists are themselves a part of HAL's legacy.

When *2001* was released, one perceptive film reviewer called it "the best informed dream" of the future. In the following chapters, we look at that dream to see just how well-informed and how prescient it was and to gain a deeper understanding and appreciation of this truly remarkable film.

Could we Build HAL?
Supercomputer Design

David J. Kuck

Could we build a computer like HAL today? For any given technolgy what factors determine how powerful a computer we can produce? What might be the general design -- the computer architecture -- of a HAL suited to tasks as diverse as controlling a spaceship and discussing personal psychology with the crew? What types of hardware components would we use? How would the hardware support the complex software required to carry out the many humanlike functions of which HAL is capable?

With the 20/20 vision of hindsight, we can examine the specific predictions of both the book and the film versions of *2001* and observe now that some of them were far too optimistic and others were not nearly optimistic enough. In fact, over the past fifty years, progress in computer engineering and computer science has been breathtaking in some aspects and disappointing in others. While some computer systems have become workhorses and milestone systems against which all successor systems are judged, others have shown great promise but delivered too little to attract a wide following.

So how does one make a rational prediction about comptuer and microelectronics techology? Two basic hardware characteristics, which vary over time, allow us to predict future system size and capability .....

"An Enjoyable Game": How HAL Plays Chess

Murray S. Campbell

The chess scene in *2001* is just one example of the genius behind Clarke and Kubrick's screenplay. Although the game between HAL and astronaut Frank Poole is shown for only about thirty seconds, it conveys a great deal of information about HAL and the relationship between Frank and HAL. The fact that HAL can beat Frank at one of the world's oldest and most difficult games is clearly intended to establish HAL as an intelligent entity. But is this a correct conclusion? Does a machine need to be intelligent to play chess?

The question of whether HAL's chess ability demonstrates intelligence boils down to a question of *how* HAL plays chess. If, on the one hand, HAL plays chess in the "human style" -- employing explicit reasoning about move choices and large amounts of chess knowledge -- the computer can be said to demonstrate some aspects of intelligence. If, on the other hand, HAL plays chess in the computer style -- that is, if HAL uses his computational power to carry out brute-force searches through millions or billions of possible alternatives, using relatively little knowledge or reasoning capabilities -- then HAL's chess play is not a sign of intelligence.

This chapter attempts to resolve this question by examining in detail how HAL plays chess and by comparing HAL with Deep Blue, the world's current premier chess computer. I and my colleagues, Feng-

hsiung Hsu and A. Joseph Hoane, Jr., developed Deep Blue at IBM's T. J. Watson Research Labs. It was the first machine in history to beat the human world champion, Garry Kasparov, in a regulation chess game. The chapter also examines the strengths and weaknesses of computer-style chess by looking at some of the games between Kasparov and Deep Blue. Finally, we discover that HAL's first error occurred in the chess game with Frank.

Before we analyze how HAL plays chess, we need to put his game with Frank into perspective by understanding the history of man-machine chess matches. What is the significance of a machine beating a human at a game like chess?

Frank versus HAL; Man versus Machine

HAL claims to be "foolproof and incapable of error." But, as we witness only one isolated game between Frank and HAL, how do we really know that HAL plays well? The answer can be determined, not so much by the game itself but by Frank's reaction to it.

Poole: Umm ... anyway, Queen takes pawn.

HAL: Bishop takes Knight's Pawn.

Poole: Lovely move. Er .. Rook to King One

HAL: I'm sorry, Frank. I think you missed it. Queen to Bishop Three. Bishop takes Queen. Knight takes Bishop. Mate.

Poole: Ah ... Yeah, looks like you're right. I resign.

HAL: Thank you for an enjoyable game.

Poole: Yeah. Thank you.

Having personally witnessed scores of amateur chess players lose to computers, I found Frank's reaction to losing to HAL extremely realistic. After HAL announces mate, Frank's pause is brief. This brevity is significant, because it demonstrates that Frank assumes HAL is right. He trusts that HAL has the details of the checkmate correct and does not take the time to confirm them for himself. Instead, Frank resigns immediately. Moreover, it is obvious from his tone of voice -- or perhaps I should say from his complete lack of tone -- that he never expected to win. In fact, Frank would have been utterly stunned if HAL had lost. No, playing chess with HAL is simply a way for Frank to pass the time on the eighteen-month journey to Jupiter. (As HAL is running virtually every aspect of the ship, there is little for the two, nonhibernating astronauts to do.) It is also clear from the dialogue, as well as from Frank's body language, that this is not a game between two competitors but one between two conscious entities -- one of whom is vastly superior in intelligence to the other.

Clearly, Frank does not feel bad about losing to a computer, any more than a sprinter would feel bad about being outrun by a race car. Nor do we, the viewers, feel particularly sorry for Frank's loss. We don't mind HAL winning, because at this stage in the film we like HAL. The human relationship with chess computers hasn't always been so amicable though.

In many recent human-machine matches, the mood has been decidedly pro-human and anti-computer. In the first

encounter between the human world champion (Kasparov) and the computer world champion (Deep Thought, Deep Blue's predecessor) in 1989, there was definite hostility toward the computer. When Kasparov pulled out the victory, the audience breathed an audible sigh of relief. In gratitude for "saving human pride," onlookers gave Kasparov a standing ovation.

Kasparov couldn't, however, save humanity's pride indefinitely. In 1995 he lost a game of speed chess to a computer program called GENIUS3. Burying his head between his hands, Kasparov could not hide his despair; he stormed off the stage, shaking his head in disbelief. The loss, reported by newspapers and magazines around the globe, shocked the multitude of those -- players and nonplayers alike -- who believed that the strongest player in the history of the game would never suffer defeat at the hands of "a silicon monster." Although Kasparov was badly shaken by this upset, it was, after all, only speed chess -- a game in which decisions are made within severe time constraints. (Speed chess allows each player only twenty-five minutes for the entire game, whereas players in regulation chess each have two hours to complete forty moves.)

In February of 1996, Kasparov played Deep Blue in a six-game, full-length regulation match sponsored by the Association for Computing Machinery in celebration of the fiftieth anniversary of the computer. Before the match, Kasparov was quoted as saying, "To some extent this is a defense of the whole human race." When he lost the first game, his computer adviser, Frederick Friedel, openly acknowledged that Kasparov was devastated (see figure 5.1). Even though he rebounded to win the match, *Time* magazine called the first-game defeat an

event larger than "world historical. It was species-defining."

Other grandmasters refuse to play against computers at all. Why? Perhaps because the idea of computer superiority in an arena as cerebral as chess is so disorienting; in Western culture, many consider chess the ultimate test of the human intellect. (It is interesting that Kubrick originally filmed the "chess scene" with a five-in-a-row board game called pentominoes but chose not to use it, believing that viewers would better appreciate the difficulties involved in a chess game.) Mathematicians have estimated that there are more possible chess positions than there are atoms in the universe. Therefore, skilled chess players must possess the ability to make difficult calculations and recognize a seemingly infinite number of patterns. Yet excellent chess play also requires imagination, intuition, ingenuity, and the passion to conquer. If a machine can beat a man at a game requiring as much creativity as chess does, what does that say about our "unique" human qualities?

For now, at least, we can rest assured that even though the best computers are better at chess than 99.999999 percent of the population, they do not actually play chess the way humans do. In the history of man's rivalry with machines, only one grandmaster-level computer has appeared to play like a human -- and that computer is our fictitious friend HAL.

How HAL Plays Chess

By analyzing the game between Frank and HAL, we can uncover a number of clues about how HAL plays chess.

As I explain in more detail below, HAL appears to play chess the way humans do. Even more important perhaps, the game reveals that HAL is not simply mimicking the way humans play; he actually understands how humans think.

The game in the screenplay is a real one played by two undistinguished players in Hamburg in 1913. Kubrick, a former Washington Square Park chess hustler and aficionado, selected a clever checkmate but was careful not to employ one too complex for viewers to grasp. He picked a position from a fairly obscure game -- one obscure enough not to appear in the 600,000-game database of Deep Blue. I eventually located the game after being directed to an article written by Grandmaster Larry Evans on January 12, 1990 (HAL's birthday).

Evans makes the crucial point in his article that HAL should have said "Queen to Bishop six" (not three). HAL used the so-called descriptive notation system that describes moves from the viewpoint of the moving player. This contrasts with the algebraic-notation system used in the game score (see Appendix to chapter), in which moves are described from White's viewpoint. HAL used the incorrect viewpoint when giving his fifteenth move. Was the notation error a deliberate foreshadowing of the machine's fallibility or merely a writer's oversight? This is a question only Kubrick can answer. If Poole had been a little more attentive, he might have realized sooner rather than later that the HAL 9000 was indeed capable of error. But, like most chess players, he was focusing on the actual moves; he was not looking for errors because he had never even considered the possibility that HAL was capable of making one.

To better understand how HAL chooses to play against Frank, it is important to have some sense of Frank's chess background. Although the movie does not disclose his chess rating, it is easy enough to speculate about his skill level. He is a highly educated man who holds a doctoral degree, most likely in a field such as aerospace engineering or robotics. We can surmise that, as second in command on a top-secret space mission of unprecedented importance, Frank is well above average intelligence. Because he is a full-time astronaut, it is unlikely that he would have time to compete in professional chess tournaments; yet he clearly knows something about chess, for his game with HAL follows opening theory for eleven moves (see Appendix for a complete account of the game). Frank's chess rating may be in the expert range, making him strong enough to engage in an interesting game but certainly not experienced enough to handle HAL. (See figure 5.2 for an explanation of the rating scale.)

In the game itself, Frank plays White and HAL is Black. Frank chooses an unusual but perfectly sound variation of the well-known Ruy Lopez, or Spanish opening. HAL responds with very aggressive play, creating a situation that makes it very difficult for Frank to find the best moves. By the time the movie picks up the game, Frank has already made the losing move, and he goes down without much of a fight.

The game provides sufficient evidence that HAL plays chess the way humans play chess. Early in the game HAL uses an apparently nonoptimal but very "trappy" move. The choice creates a very complex situation in which the "obvious" move is a losing blunder. If Frank had been able to find the best move, he would have gained the

advantage over HAL. In leading Frank into this trap, HAL appears to be familiar with Frank's level of play, and we can assume that HAL is deliberately exploiting Frank's lack of experience. < /P>

The interesting point here is that present-day chess programs do not normally play trappy chess. They are almost always based on the *minimax principle*, which assumes that the opponent always makes the best move. (I discuss this principle in more detail later in the chapter.) A machine like Deep Blue, therefore, would only play the optimal move found in its search. The ability of HAL to play trappy moves is a sign of a sophisticated player who is familiar with the opponent's strengths and weaknesses.

A second interesting point in the game occurs on move 13. The move played by HAL is clearly a winning move, but Deep Blue would have found a move that forces checkmate one move sooner. Current programs always prefer the shortest checkmate. Thus, either HAL is not able to calculate as deeply as Deep Blue does or he chooses a move based on "satisficing" criteria; that is, HAL saw that the move guaranteed a win, and so did not bother to search for a better move. Human chess players commonly follow this practice, which is another piece of evidence pointing to HAL's human style of play.

So how do we now that HAL understands how humans think? When HAL plays his spectacular fifteenth move, he surmises, undoubtedly correctly, that Frank had overlooked this move. Further, HAL did not point out to Frank the other possible variations to checkmate -- only the most interesting line, the one that Frank would most appreciate. Although Frank need not have accepted HAL's

queen sacrifice, a prosaic checkmate would have followed shortly anyway.

HAL's ability to play chess human style is what computer scientists in the 1960s might have expected.
When *2001* was produced, the majority of artificial intelligence researchers probably believed that computers should play the way humans play: by using explicit reasoning about move decisions and applying large amounts of pattern-directed knowledge. It wasn't until the 1970s, after years of much hard work and little progress, that chess programmers tried a new strategy, which is still utilized in the 1990s. A brief history of computer chess and some of its key components is relevant to understanding how machines play today. Perhaps we should start with an even more basic question: Why develop a chess machine in the first place?

A Brief History of Computer Chess: The Early Days

In 1950, Claude Shannon, the founder of information theory, proposed that developing a chess machine would be an excellent way to work on issues associated with machine intelligence. In his article, "A Chess-Playing Machine," Shannon states the case for developing such a machine: "The investigation of the chess-playing problem is intended to develop techniques that can be used for more practical applications. The chess machine is an ideal one to start with for several reasons. The problem is sharply defined, both in the allowed operations (the moves of chess) and in the ultimate goal (checkmate). It is neither so simple as to be trivial nor too difficult for satisfactory solution. And such a machine could be pitted

against a human opponent, giving a clear measure of the machine's ability in the type of reasoning."

In fact, the practical applications that could result from development of a world-class chess machine are numerous. Complex tasks that may be solved by technologies derived from Deep Blue include problems in chemical modeling, data mining, and economic forecasting.

Fascination with the idea of a chess-playing machine, however, began more than two centuries ago, long before anyone thought of using a computer to solve large-scale problems. In the 1760s Baron Wolfgang von Kempelen toured Europe with the Maezal Chess Automaton, nicknamed the Turk. The machine was nicknamed the Turk because it played its moves through a turbaned marionette attached to a cabinet. The cabinet supposedly contained "the brain" of the machine; it was later discovered that the brain was actually a chess master of small stature.

The first documented discussion of computer chess is in *The Life of a Philosopher* by Charles Babbage (1845). Babbage, whose remarkable ideas in mathematics and science were far ahead of his time, proposed programming his Analytical Engine -- a precursor of the computer -- to play chess. A century later, Alan M. Turing, the British mathematician and computer scientist, developed a program that could generate simple moves and evaluate positions. Lacking a computer with which to run his program, Turing ran it by hand. Konrad Zuse, a German computer science pioneer, in his *Der Plankalkuel* (1945), described a program for generating

legal chess moves. He even developed a computer, although he did not actually program it to play chess.

In spite of these earlier precedents, it was Shannon's efforts that laid the groundwork for actual research, and he is generally considered the "father of computer chess." Shannon's work was based, in turn, on the findings of John von Neumann and Oskar Morgenstern, game theorists who devised a minimax algorithm by which the best move can be calculated.

Key Components of a Chess Program

The minimax algorithm can be thought of as consisting of two parts: an evaluation function and the minimax rule. An *evaluation function* for any chess position produces a number that measures the "goodness" of the position. Positions with positive values are good for White, and negative values are good for Black. The higher the score, the better it is for White, and vice versa. The *minimax rule* allows the evaluation function values to be used. It simply states that, when White moves, White chooses the move that leads to the maximum value, and when Black moves, Black chooses the move that leads to the minimum value.

In theory, the minimax algorithm allows one to play "perfect" chess; that is, the player always makes a winning move in a won position or a drawing move in a drawn position. Of course, perfect chess is just a fantasy; chess is far too vast a game for perfect play, except when there are only a few pieces on the board. In practice, chess

programs examine only a limited number of moves ahead -- typically between four and six.

Although minimax is very effective, it is also quite inefficient. In the opening position in a chess game, White has twenty moves, and Black has twenty different replies to each of these -- thus there are four hundred possible positions after the first move. After two moves there are more than twenty thousand positions, and after five moves the number of potential chess positions is into the trillions. Even today's fastest computers cannot process this many positions. The *alpha-beta algorithm* improves the minimax rule by greatly reducing the number of positions that must be examined. Instead of exploring trillions of positions after five moves, the computer only needs to analyze millions.

The Modern Era of Computer Chess

The principles of the minimax and alpha-beta algorithms were well understood in the 1960s. When *2001* made its screen debut in 1968, however, the very best computer chess program was only as strong as the average tournament player. Still, many computer scientists believed that building a world-class chess machine was a fairly straightforward problem, one that would not take long to solve.

The earliest approach to solving it involved emulating the human style of play. It is now clear that this was an extraordinarily difficult way to tackle computer chess. Even though chess seems to be a simple and restricted domain, people use many different aspects of intelligence in top-level play,including calculation of possible outcomes, sophisticated pattern recognition and

evaluation, and general-purpose reasoning. Significant progress in computer chess did not occur until 1973, when David Slate and Larry Atkin wrote a well-engineered brute-force chess program called Chess 4.0. Since then, almost all good chess programs have been based on their work.

The Slate/Atkin program remained the best chess-playing computer program throughout the 1970s; it gained in strength with each new, faster generation of computer hardware. It was observed in practice, and verified by experiment, that every fivefold speedup in computer hardware led to a two-hundred-point increase in the program's rating as it approached the master level. Subsequent chess-playing machines pushed the computer chess ratings higher and higher -- in large part due to faster hardware, although software was also improving rapidly. The Slate/Atkin program reached the expert level (2,000) by 1979; in 1983 Belle, a machine from AT&T; Bell Labs, used specially designed chess hardware to reach the master's level (2,200); then came Cray Blitz, which ran on a Cray supercomputer, and Hitech, which dedicated a special-purpose chip to each of the sixty-four squares on a chessboard. Recognizing this trend, Ray Kurzweil predicted that a computer would beat the world champion around 1995. All these machines were finally surpassed by Deep Thought, which began playing in 1988 (see figure 5.3). Designed and programmed by a group of graduate students (myself included), Deep Thought was the first machine to defeat a grandmaster in tournament play; it was capable of searching up to seven hundred thousand chess positions per second. Deep Thought eventually led to Deep Blue, still the world's best chess-playing machine.

How Deep Blue Plays Chess

The objectives of the Deep Blue project were to develop a machine capable of playing at the level of the human world chess champion and to apply the knowledge gained in this work to solving other complex problems. To accomplish these goals, a significant increase in processing power was necessary. Today Deep Blue is capable of searching up to two hundred million chess positions per second. Its ability to search such an extraordinary number of positions prompted Kasparov to comment that "quantity had become quality." In other words, the computer is able to search so deeply into a position that it can discover difficult and profound moves. Although Deep Blue uses a variety of techniques to achieve its high level of chess play, the heart of the machine is a *chess microprocessor* (see figure 5.4).

Designed over a period of several years, this chip was built to search and evaluate up to two million chess positions per second. By itself, however, the chip cannot play chess. It requires the control of a general-purpose computer to make it work. Deep Blue runs on an IBM SP2 supercomputer with thirty-two separate computers (or nodes) that work in concert. For the match against Kasparov, each SP2 node controlled up to eight chess chips, while the entire SP2 system had about 220 chess chips that could be run in parallel. The old saying about too many cooks spoiling the broth is also applicable to parallel computers. A lot of processors won't do much good unless they can all be kept busy doing useful work. In fact, parallelizing a chess program efficiently has

proven to be a very difficult problem. For the match with Kasparov, Deep Blue looked at an average of close to one hundred million positions per second.

Nonetheless, a purely brute-force machine would have little chance against a player like Kasparov. Although grandmasters require very little actual calculation of variations for most moves, there are typically a few key points in a game where they must calculate the possible variations very deeply. Often these calculations far surpass what brute-force search could hope to attain. To overcome this problem, Deep Blue employs a technique called *selective extensions*, which enables the computer to search critical positions more deeply.

One of the questions most commonly asked about a chess computer is, "How deep does it search?" In the early days of the computer chess, most programs searched all lines to roughly the same depth, and this question was relatively easy to answer. The fact that Deep Blue employs sophisticated selective searches complicates the issue considerably. When asked how deeply Deep Blue searches, one can give at least three answers; minimum depth (six moves in typical middle game positions); average depth (perhaps eight moves); and maximum depth (highly variable, but typically in the ten-to-twenty-move range).

Yet, although Deep Blue's speed and search capabilities enable it to play grandmaster-level chess, it is still lacking in general intelligence. It is clear that there are significant differences between the way HAL and Deep Blue play chess.

How HAL Compares with Deep Blue

As we mentioned earlier, there is considerable evidence that HAL plays chess in the human style. In fact, given that Kubrick and Clarke chose a game between two humans as the model for the Frank Poole-HAL game, it would have been extraordinary if HAL had not played in the human style. Deep Blue, on the other hand, is a classic brute-force-based machine, albeit it has considerable search selectivity. So a comparison between HAL and Deep Blue must begin by comparing computer and human styles of chess playing.

The difference is actually quite subtle and would probably be detected only by persons experienced with computer play. A computer engaged in an electronic dialogue is said to have passed the Turing test if the computer's conversation is indistinguishable from that of a human being. At the present time, no computer has ever passed the Turing test. HAL, by comparison, would pass with flying colors -- and later turn around and try to kill the person administering the test!

Drawing on this analogy, one could devise a Turing test for computer chess programs. That is, a chess machine would pass the chess-restricted Turing test if the person playing the machine could not determine whether or not he or she was playing against another person or a machine. Most players would find it difficult to discern whether or not a Deep Blue game was played by a human or a computer. This was proven in an informal experiment conducted by Frederic Friedel, Kasparov's computer adviser. Friedel showed Kasparov a series of games in a tournament played by Deep Thought and several grandmasters. Without identifying the players, Friedel

asked Kasparov to pick out the moves made by the computer. In a number of cases Kasparov mistook the computer's moves for those of a human grandmaster, or vice versa. In general, only chess players who have considerable experience playing against computers can identify computer moves.

A specific example demonstrates the difference between the human style of play and the computer style of play: the fact that chess programs exhibit a lack of understanding of the role of timing in chess. Concepts involving *never*, *eventually*, or *any time* can be very difficult for computer programs. For example, a weapon in the arsenal of most strong human players is the idea of a *fortress* -- a position where a player who has fewer or less-powerful pieces, can create an impenetrable position in which the opponent can never make progress (see figure 5.4). In the 1996 Kasparov-Deep Blue match, Kasparov was able to clinch a draw in the fourth game by means of a sacrifice that created a fortress (see figure 5.5). Although Deep Blue can be programmed to identify many different specific fortresses, detecting the general case of a fortress is still beyond its capabilities and presents us with a complicated pattern-recognition problem worthy of further research.

Another difference between human and computer styles of play can be seen by examining a position involving the ability to reason. At the conclusion of the historic match, Kasparov visited our research lab and showed us a position from which he was absolutely certain that Black would eventually checkmate. Kasparov could not say precisely how many moves it would take, and he was curious to see how Deep Blue would analyze the position. Even after several minutes of search, however, Deep Blue

did not see the checkmate. Sometimes search is a very poor substitute for reasoning.

There is, of course, another obvious difference between the human style (HAL) and the computer style (Deep Blue) of play: Humans have emotion. One of the supposed advantages of computers over humans in a game like chess is that computers lack emotion. They are not embarrassed by previous mistakes, they don't slump dejectedly in their chairs when they get into a bad position. One wonders, then, whether HAL's emotional side possibly influenced his style of play (see chapter 13).

When HAL thanks Frank for "an enjoyable game," this is more than simply a pleasing platitude entered into HAL's system by his programmers. Because he possesses both emotion and general intelligence, HAL has the ability to enjoy a good game of chess. Alas, while Deep Blue is sometimes capable of playing magnificent, world-class chess, it is unable to appreciate its own moves.

How, one might speculate, would Deep Blue fare in a match against HAL? Deep Blue could find all the moves HAL plays to finish off the game with Frank in a fraction of a second. Clearly, both machines are tactically very strong. However, given HAL's general intelligence, one suspects it would be able to avoid most of the typical computer mistakes to which brute-force machines like Deep Blue are susceptible. On the other hand, Deep Blue's search strategy could be a strength; it might find counterintuitive moves that would probably be dismissed by a humanlike search. I suspect it would be a very interesting match, in which each computer would gain its fair share of wins.

The idea of HAL losing a game, however, brings up an interesting point. Throughout the film, HAL consistently asserts that he is "incapable of error." Given the overwhelming complexity of the game, it is not plausible for HAL to play perfect chess, as this would require HAL to have solved all possible chess problems. So, if HAL does not play perfect chess, there must be some winning positions in which HAL fails to play a winning move -- or drawn positions in which he doesn't find the drawing move. In the normal sense of the word, these would constitute errors. HAL's own interpretation of the word *error* remains mysterious.

Man versus Machine today

In one six-game match, the 1996 Kasparov-Deep Blue "showdown" demonstrated both the great strengths and the great weaknesses of 1990s computer chess machines. The diagram in figure 5.6 illustrates how quantity can indeed become quality.

This position was taken from Game 1 of the match. Deep Blue's move23 was **P-Q5** (or d5 in algebraic notation). This strong move completed thedemolition of Kasparov's pawn structure; all Black's pawns were soon isolated and unable to support each other. Kasparov knew that 23. **P-Q5** was a strong move, but he did not expect it from a computer, because it involved a pawn sacrifice -- something computers are often reluctant to do. However, Deep Blue, in analyzing the position, saw deeply enough to realize that 23. **P-Q5** was only a temporary pawn sacrifice; that is, it saw that it would

later win back the pawn and retain all the other advantages.

As figure 5.7 illustrates, however, computers can sometimes lack basic chess concepts that are understood even by amateur players. The diagram shows the final position in Game 6 of the match. Although Deep Blue was actually a pawn ahead, its pieces were all trapped, or immobilized. Deep Blue had not recognized the danger in this position many moves earlier, when there was still a chance to avoid it. If Deep Blue had not resigned, Kasparov could have won easily by, for example, opening up the king side and attacking the undefended king. The human ability to reason about permanently trapped pieces was a deciding factor in this game.

Although the competitive aspects of human-versus-computer play attract considerable attention, cooperation between man and machine is becoming more and more common. Many grandmasters use PC chess programs to help them analyze chess positions. And players can now learn more about chess endgames by studying computer-generated endgame databases that demonstrate perfect play in positions with five or fewer pieces on the board. But, perhaps most notably, Kasparov feels that the 1996 match with Deep Blue helped him understand more about chess. This may be a sign of things to come.

The Future of Computer Chess

Early optimism in the field of artificial intelligence led people to believe that the chess problem would be

relatively easy to solve. In the late 1950s, Herbert Simon, one of the founding fathers of artificial intelligence, predicted that it would take only ten years for a machine to become world champion. Despite his expertise in the field, Simon was off by at least thirty years. After Kasparov lost a regulation game to Deep Blue, many people mistakenly assumed that the chess-playing problem had finally been solved. It is becoming more and more apparent, however, that chess mastery requires an intriguing mixture of skills: pure calculation, sophisticated evaluation, learning, and a generalized reasoning capability. Although a machine like Deep Blue excels in calculation, at present it still lacks many other skills essential to consistent world-class chess play. Until computers possess the ability to reason, strong human chess players will always have a chance to defeat a computer-style opponent.

Given recent advances in hardware speed and algorithms, I believe Kasparov's loss to a machine in a regulation match was inevitable. Kasparov still has the advantage in that he has the ability to adapt quickly to weaknesses in a computer opponent, a skill that current chess-playing machines lack. With continued progress, however, it is likely that we will see the end of competitive matches between man and machine sometime in the next century. Certainly competitive chess will continue: man against man; machine against machine. Ultimately, though, the computer's superiority over human players will be so great that the only value in man-versus-machine play would be the instructional benefit it provides human players, or -- as in *2001* -- ts recreational use on journeys to faraway planets. The applications that have been, and will continue to be

derived from developing a world-class chess machine will advance our use of computers as tools for solving other complex problems. Even so we are still decades away from creating a computer with HAL's capabilities.

Acknowledgments

I would like to acknowledge the other members of the Deep Blue team: Feng-hsiung Hsu, the principal designer of Deep Blue, and A. Joseph Hoane, Jr. Other IBM Research staff that supported the project include C.J. Tan and Jerry Brody. My thanks to tcrain@s2.sonnet.com for directing me to the article by Grandmaster Larry Evans that includes the Frank Poole-HAL game in its entirety.

When Will HAL Understand What We Are Saying? Computer Speech Recognition and Understanding

Raymond Kurzweil

Let's talk about how to wreck a nice beach.

Well, actually, if I were presenting this chapter verbally, you would have little difficulty understanding the preceding sentence as *Let's talk about how to recognize speech*. Of course, I wouldn't have enunciated the *g* in *recognize*, but then we routinely leave out and otherwise slur at least a quarter of the sounds that are "supposed" to be there a phenomenon speech scientists call coarticulation.

On the other hand, had this been an article on a rowdy headbangers' beach convention (a topic we assume HAL knew little about), the interpretation at the beginning of the chapter would have been reasonable.

On yet another hand, if you were a researcher in speech recognition and heard me read the first sentence of this chapter, you would immediately pick up the beach -- wrecking interpretation, because this sentence is a famous example of acoustic ambiguity and is frequently cited by speech researchers.

The point is that we understand speech in context. Spoken language is filled with ambiguities. Only our understanding of the situation, subject matter, and person (or entity) speaking -- as well as our familiarity with the speaker -- lets us infer what words are actually spoken.

Perhaps the most basic ambiguity in spoken language is the phenomenon of *homonyms*, words that sound absolutely identical but are actually different words with different meanings. When Frank asks, "Listen, HAL, there's never been any instance at all of a computer error occurring in a 9000 series, has there?", HAL has little difficulty interpreting the last word as *there* and not *their*. Context is the only source of knowledge that can resolve such ambiguities. HAL understands that the word their is an adjective and would have to be followed by the noun it modifies. Because it is the last word in the sentence, there is the only reasonable interpretation. Today's speech -- recognition systems would also have little difficulty with this word and would resolve it the same way HAL does.

A more difficult task in interpreting Frank's statement is the word all. Is all a place ; such as IBM headquarters -- where a computer error may take place, as in "there's never been any instance of a computer error at IBM ... " HAL resolves this ambiguity the same way viewers of the movie do. We know that all is not the name of a place or organization where an error may take place. This leaves us with *at all* as an expression of emphasis reinforcing the meaning of never as the only likely interpretation.

In fact, we try to understand what is being said before the words are even spoken, through a process called *hypothesis and test*. Next time you order coffee in a restaurant and a waiter asks how you want it, try saying "I'd like some dream and sugar please." It would take a rather attentive person to hear that you are talking about sweet dreams and not white coffee.

When we listen to other people talking -- and people frequently do not really listen, a fault that HAL does not seem to share with the rest of us -- we constantly anticipate what they are going to say ... next. Consider Dave's reply to HAL's questions about the crew psychology report:

Dave: Well, I don't know. That's rather a difficult question to ...

When Dave finally says answer, HAL tests his hypothesis by matching the word he heard against the word he had hypothesized Dave would say. In watching the movie, we all do the same thing. Any reasonable match would tend to confirm our expectation.

The test involves an acoustic matching process, but the hypothesis has nothing to do with sound at all -- nor even with language -- but rather relates to knowledge on a multiplicity of levels. As many of the chapters in this book point out, knowledge goes far beyond mere facts and data. For *information* to become *knowledge*, it must incorporate the relationships between ideas. And for knowledge to be useful, the links describing how concepts interact must be easily accessed, updated, and manipulated. Human intelligence is remarkable in its ability to perform all these tasks. Ironically, it is also remarkably weak at reliably storing the information on which knowledge is based. The natural strengths of today's computers are roughly the opposite. They have, therefore, become powerful allies of the human intellect because of their ability to reliably store and rapidly retrieve vast quantities of information. Conversely, they have been slow to master true knowledge. Modeling the knowledge needed to understand the highly ambiguous and variable phenomenon of human speech has been a primary key to making progress in the field of automatic speech recognition (ASR).

Lesson 1: Knowledge Is a Many -- layered Thing

Thus lesson number one for constructing a computer system that can understand human speech is to build ;in knowledge at many levels: the structure of speech sounds, the way speech is produced by our vocal apparatus, the patterns of speech sounds that comprise dialects and languages, the complex (and not fully understood) rules of word usage, and the -- greatest

difficulty -- general knowledge of the subject matter being spoken about.

Each level of analysis provides useful constraints that can limit our search for the right answer. For example, the basic building blocks of speech called *phonemes* cannot appear in just any order. Indeed, many sequences are impossible to articulate (try saying *ptkee*). More important, only certain phoneme sequences correspond to a word or word fragment in the language. Although the set of phonemes used is similar (although not identical) from one language to another, contextual factors differ dramatically. English, for example, has over ten thousand possible syllables, whereas Japanese has only a hundred and twenty.

On a higher level, the syntax and semantics of the language put constraints on possible word orders. Resolving homonym ambiguities can require multiple levels of knowledge. One type of technology frequently used in speech recognition and understanding systems is a sentence parser, which builds sentence diagrams like those we learned in elementary school (see figure 7.1). One of the first such systems, developed in 1963 by Susumu Kuno of Harvard (around the time Kubrick and Clarke began work on *2001*), revealed the depth of ambiguity in English. Kuno asked his computerized parser what the sentence "Time flies like an arrow" means. In what has become a famous response, the computer replied that it was not quite sure. It might mean

1. That time passes as quickly as an arrow passes.

2. Or maybe it is a command telling us to time the flies the same way that an arrow times flies; that is, Time flies like an arrow would.

3. Or it could be a command telling us to time only those flies that are similar to arrows; that is, Time flies that are like an arrow.

4. Or perhaps it means that a type of flies known as time flies have a fondness for arrows: Time -- flies like (i.e., appreciate) an arrow."

It became clear from this and other syntactical ambiguities that understanding language, spoken or written, requires both knowledge of the relationships between words and of the concepts underlying words. It is impossible to understand the sentence about time (or even to understand that the sentence is indeed talking about time and not flies) without mastery of the knowledge structures that represent what we know about time, flies, arrows, and how these concepts relate to one another.

A system armed with this type of information would know that flies are not similar to arrows and would thus knock out the third interpretation. Often there is more than one way to resolve language ambiguities. The third interpretation could be syntactically resolved by noting that *like* in the sense of *similar to* ordinarily requires number agreement between the two objects compared. Such a system would also note that, as there are no such things as *time flies*, the fourth interpretation too is wrong. The system would also need such tidbits of knowledge as the fact that flies have never shown a fondness for arrows, and that arrows cannot and do

not *time* anything -- much less flies -- to select the first interpretation as the only plausible one. The ambiguity of language, however, is far greater than this example suggests. In a language -- parsing project at the MIT Speech Lab, Ken Church found a sentence with over two million syntactically correct interpretations.

Often the tidbits of knowledge we need have to do with the specific situation of speakers and listeners. If I walk into my business associate's office and say "rook to king one," I am likely to get a response along the lines of "excuse me?" Even if my words were understood, their meaning would still be unclear; my associate would probably interpret them as a sarcastic remark implying that I think he regards himself as a king. In the context of a chess game, however, not only is the meaning clear, but the words are easy to recognize. Indeed, our contemporary speech -- recognition systems do a very good job when the domain of discourse is restricted to something as narrow as a chess game. So, HAL, too, has little trouble understanding when Frank says "rook to king one" during one of their chess matches.

Lesson 2: The Unpredictability of Human Speech

A second lesson for building our computer system is that it must be capable of understanding the variability of human speech. We can, of course, build in pictures of human speech called spectrograms, which plot the intensity of different frequencies (or pitches in human perceptual terms) as they change over time. What is interesting, and -- for those of us developing speech-recognition machines -- daunting is that spectrograms of

two people saying the same word can look dramatically different. Even the same person pronouncing the same word at different times can produce quite different spectrograms.

Look at the two spectrogram pictures of Dave and Frank saying the word *HAL* (figure 7.2). It would be difficult to know that they are saying the same word from the pictures alone. Yet the spectrograms present all the salient information in the speech signals.

Yet, there must be something about these different sound pictures that is the same; otherwise we humans and HAL, as a human-level machine, would be unable to identify them as two examples of the same spoken word. Thus, one key to building automatic speech recognition (ASR) machines is the search for these *invariant features*. We note for example that vowel sounds (e.g., the *a* sound in *HAL*, which may be denoted as *æ* (or /*a*/) involve certain resonant frequencies called *formants* that are sustained over some tens of milliseconds. We tend to find these formants in a certain mathematical relationship whenever /*a*/ is spoken. The same is true of the other sustained vowels. (Although the relationship is not a simple one, we observe that the relationship of the frequency of the second formant to the first formant for a particular vowel falls within a certain range, with some overlap between the ranges for different vowels.) Speech recognition systems frequently include a search function for finding these relationships, sometimes called features.

By studying spectrograms, we also note that certain changes do not convey any information; that is, there are types of changes we should filter out and ignore. An obvious one is loudness. When Dave shouts HAL's name in the pod in space, HAL realizes it is still his name. HAL infers some meaning from Dave's volume, but it is relatively unimportant for identifying the words being spoken. We apply, vtherefore, a process called *normalization*, in which we make all words the same loudness so as to eliminate this noninformative source of variability.

A system armed with this type of information would know that flies are not similar to arrows and would thus knock out the third interpretation. Often there is more than one way to resolve language ambiguities. The third interpretation could be syntactically resolved by noting that *like* in the sense of *similar to* ordinarily requires number agreement between the two objects compared. Such a system would also note that, as there are no such things as *time flies*, the fourth interpretation too is wrong. The system would also need such tidbits of knowledge as the fact that flies have never shown a fondness for arrows, and that arrows cannot and do not *time* anything -- much less flies -- to select the first interpretation as the only plausible one. The ambiguity of language, however, is far greater than this example suggests. In a language -- parsing project at the MIT Speech Lab, Ken Church found a sentence with over two million syntactically correct interpretations.

Often the tidbits of knowledge we need have to do with the specific situation of speakers and listeners. If I walk into my business associate's office and say "rook to king

one," I am likely to get a response along the lines of "excuse me?" Even if my words were understood, their meaning would still be unclear; my associate would probably interpret them as a sarcastic remark implying that I think he regards himself as a king. In the context of a chess game, however, not only is the meaning clear, but the words are easy to recognize. Indeed, our contemporary speech -- recognition systems do a very good job when the domain of discourse is restricted to something as narrow as a chess game. So, HAL, too, has little trouble understanding when Frank says "rook to king one" during one of their chess matches.

Lesson 2: The Unpredictability of Human Speech

A second lesson for building our computer system is that it must be capable of understanding the variability of human speech. We can, of course, build in pictures of human speech called spectrograms, which plot the intensity of different frequencies (or pitches in human perceptual terms) as they change over time. What is interesting, and -- for those of us developing speech-recognition machines -- daunting is that spectrograms of two people saying the same word can look dramatically different. Even the same person pronouncing the same word at different times can produce quite different spectrograms.

Look at the two spectrogram pictures of Dave and Frank saying the word *HAL* (figure 7.2). It would be difficult to know that they are saying the same word from the

pictures alone. Yet the spectrograms present all the salient information in the speech signals.

Yet, there must be something about these different sound pictures that is the same; otherwise we humans and HAL, as a human-level machine, would be unable to identify them as two examples of the same spoken word. Thus, one key to building automatic speech recognition (ASR) machines is the search for these *invariant features*. We note for example that vowel sounds (e.g., the *a* sound in *HAL*, which may be denoted as *æ* (or /*a*/) involve certain resonant frequencies called *formants* that are sustained over some tens of milliseconds. We tend to find these formants in a certain mathematical relationship whenever /*a*/ is spoken. The same is true of the other sustained vowels. (Although the relationship is not a simple one, we observe that the relationship of the frequency of the second formant to the first formant for a particular vowel falls within a certain range, with some overlap between the ranges for different vowels.) Speech recognition systems frequently include a search function for finding these relationships, sometimes called features.

By studying spectrograms, we also note that certain changes do not convey any information; that is, there are types of changes we should filter out and ignore. An obvious one is loudness. When Dave shouts HAL's name in the pod in space, HAL realizes it is still his name. HAL infers some meaning from Dave's volume, but it is relatively unimportant for identifying the words being spoken. We apply, vtherefore, a process called *normalization*, in which we make all words the

same loudness so as to eliminate this noninformative source of variability.

The mechanisms described above for creating speech sounds -- vocal cord vibrations, the noise of rushing air, articulatory gestures of the mouth, teeth and tongue, the shaping of the vocal and nasal cavities -- produce different rates of vibration. Physicists measure these rates of vibration as frequencies; we perceive them as pitches. Though we normally think of speech as a single time-varying sound, it is actually a composite of many different sounds, each with its own frequency. Using this insight, most ASR researchers starting in the late 1960s began by breaking up the speech waveform into a number of frequency bands. A typical commercial or research ASR system will produce between a few and several dozen frequency bands. The front end of the human auditory system does exactly the same thing: each nerve ending in the cochlea (inner ear) responds to different frequencies and emits a pulsed digital signal when activated by an appropriate pitch. The cochlea differentiates several thousand overlapping bands of frequency, which gives the human auditory system its extremely high degree of sensitivity to frequency. Experiments have shown that increasing the number of overlapping frequency bands of an ASR system (thus making it more like the human auditory system) increases the ability of that system to recognize human speech.

Lesson 4: Learn While You Listen

A fourth lesson emphasizes the importance of learning.

At each stage of processing, a system must adapt to the individual characteristics of the talker. Learning to do this has to take place at several levels: those of the frequency and time relationships characterizing each phoneme, the dialect (pronunciation) patterns of each word, and the syntactic patterns of possible phrases and sentences. At the highest cognitive level, a person or machine understanding speech learns a great deal about what a particular talker tends to talk about and how that talker phrases his or her thoughts.

HAL learns a great deal about his human crew mates by listening to the sound of their voices, what they talk about, and how they put sentences together. He also watches what their mouths do when they articulate certain phrases (chapter 11). HAL gathers so much knowledge about them that he can understand them even when some of the information is obscured -- for example, when he has to rely solely on his visual observation of Dave and Frank's lips.

Lesson 5: Hungry for MIPS and Megabytes

The fifth lesson is that speech recognition is a process hungry for MIPs (millions of instructions per second) and megabytes (millions of bytes of storage); which is to say that we can obtain more accurate performance by using faster computers with larger memories. Certain algorithms or methods are only available in computers that operate at high levels of performance. Brute force -- that is, huge memory -- is necessary but clearly not

sufficient without solving the difficult algorithmic and knowledge-capture issues mentioned above.

We now know that 1997, when HAL reportedly became intelligent, is too soon. We won't have the quantity of computing, in terms of speed and memory needed, to build a HAL. And we won't be there in 2001 either.

Let's keep these lessons in mind as we examine the roots and future prospects of building machines that can duplicate HAL's ability to understand speech.

The Importance of Speech Recognition

Before examining the sweep of progress in this field, it is worthwhile to underscore the importance of the auditory sense, particularly our ability to understand spoken language, and why this is a critical faculty for HAL. Most of HAL's interaction with the crew is verbal. It is primarily through his recognition and understanding of speech that he communicates. HAL's visual perceptual skills, which are far more difficult to create, are relatively less important for carrying out his mission, even though the pivotal scene, in which HAL understands Dave and Frank's conspiratorial conversation, without being able to hear them, relies on HAL's visual sense. Of course, his apparently self-taught lipreading is based on his speech-recognition ability and would have been impossible if HAL had not been able to understand spoken language.

To demonstrate the importance of the auditory sense, try watching the television news with the sound turned off. Then try it again with the sound on, but without looking at the picture. Next, try a similar experiment with a

videotape of the movie *2001*. You will probably find it easier to follow the stories with your ears alone than with your eyes alone, even though our eyes transmit much more information to our brains than our ears do -- about fifty billion bits per second from both eyes versus approximately a million bits per second from two ears. The result is surprising. There is a saying that a picture is worth a thousand words; yet the above exercise illustrates the superior power of spoken language to convey our thoughts. Part of that power lies in the close link between verbal language and conscious thinking. Until recently, a popular theory held that thinking was subvocalized speech. (J. B. Watson, the founder of behaviorism, attached great attention to the small movements of the tongue and larynx made while we think.) Although we now recognize that thoughts incorporate both language and visual images, the crucial importance of the auditory sense in the acquisition of knowledge -- which we need in order to recognize speech in the first place -- is widely accepted.

Yet many people consider blindness a more serious handicap than deafness. A careful consideration of the issues shows this to be a misconception. With modern mobility techniques, blind persons with appropriate training have little difficulty going from place to place. The blind employees of my first company (Kurzweil Computer Products, Inc., which developed the Kurzweil Reading Machine for the Blind) traveled around the world routinely. Reading machines can vprovide access to the world of print, and visually impaired people experience few barriers to communicating with others in groups or individual encounters. For the deaf, however,

the barrier to understanding what other people are saying is fundamental.

We learn to understand and produce spoken language during our first year of life, years before we can understand or create written language. HAL apparently spent years learning human speech by listening to his teacher, whom he identifies as Mr. Langley, at the HAL lab in Urbana, Illinois. Studies with humans have shown that groups of people can solve problems with dramatically greater speed if they can communicate verbally rather than being restricted to other methods. HAL and his human colleagues amply demonstrate this finding. Thus, intelligent machines that understand verbal language make possible an optimal modality of communication. In recent years, a major goal of artificial intelligence research has been making our interactions with computers more natural and intuitive. HAL's primarily verbal communication with crew members is a clear example of an intuitive user interface.

The Roots of Automatic Speech Recognition (ASR)

Keeping in mind our five lessons about creating speech-recognition systems, it is interesting to examine historical attempts to endow machines with the ability to understand human speech. The effort goes back to Alexander Graham Bell, and the roots of the story go even farther back, to Bell's grandfather Alexander Bell, a widely known lecturer and speech teacher. *His* son, Alexander Melville Bell, created a phonetic system for teaching the deaf to speak called *visible speech*. At the

age of twenty-four, Alexander Graham Bell began teaching his father's system of visible speech to instructors of the deaf in Boston. He fell in love with and subsequently married one of his students, Mabel Hubbard. She had been deaf since the age of four as a result of scarlet fever. The marriage served to deepen his commitment to applying his inventiveness to overcoming the handicaps of deafness.

He built a device he called a *phonautograph* to make visual patterns from sound. Attaching a thin stylus to an eardrum he obtained from a medical school, he traced the patterns produced by speaking through the eardrum on a smoked glass screen. His wife, however, was unable to understand speech by looking at these patterns. The device could convert speech sounds into pictures, but the pictures were highly variable and showed no similarity in patterns, even when the same person spoke the same word.

In 1874, Bell demonstrated that the different frequency harmonics from an electrical signal could be separated. His *harmonic telegraph* could send multiple telegraphic messages over the same wire by using different frequency tones. The next year, the twenty-eight-year-old Bell had a profound insight. He hypothesized that although the information needed to understand speech sounds could not be seen by simply displaying the speech signal directly, it could be recognized if you first broke the signal into different frequency bands. Bell's intuitive discovery of our third lesson also turns out to

be a key to finding the invariant features needed for the second lesson.

v Bell felt sure he had all the pieces needed to implement this insight and give his wife the ability to understand human speech. He had already developed a moving drum and solenoid (a metal core wrapped with wire) that could transform a human voice into a time-varying current of electricity. All he needed to do, he thought, was to break up this electrical signal into different frequency bands, as he had done previously with the harmonic telegraph, then render each of these harmonics visually -- by using multiple phonautographs. In June of 1875, while attempting to prepare this experiment, he accidentally connected the wire from the input solenoid back to another similar device. Now most processes are not reversible. Try unsmashing a teacup or speaking into a reading machine for the blind, which converts print into speech; it will not convert the speech back into print. But, unexpectedly, Bell's erstwhile microphone began to speak! Thus was the telephone discovered, or we should say, invented.

The device ultimately broke down the communication barrier of distance for the human race. Ironically, Bell's great invention also deepened the isolation of the deaf. The two methods of communication available to the deaf -- sign language and lipreading -- are not possible over the telephone.

He continued to experiment with a frequency-based phonautograph, but without a computer to analyze the rapidly time-varying harmonic bands, the information remained a bewildering array to a sighted deaf person.

We now know that we can visually examine frequency-based pictures of speech (i.e., spectrograms) and understand the communication from the visual information alone; but the process is extremely difficult and slow. An MIT graduate course, Speech Spectrogram Reading, teaches precisely this skill. The purpose of the course is to give students insight into the spectral cues of salient speech events. For many years, the course's professor, Dr. Victor Zue, was the only person who could understand speech from spectrograms with any proficiency; several people have reportedly now mastered this skill. Computers, on the other hand, can readily handle spectral information, and we can build a cruqde but usable speech-recognition system using this type of acoustic information alone. So Bell was on the right track -- about a century too soon.

Ironically, another pioneer, Charles Babbage, had attempted to create that other prerequisite to automatic speech recognition -- the programmable computer -- about forty years earlier. Babbage built his computer, the *analytical engine*, entirely of mechanical parts; yet it was a true computer, with a stored program, a central processing unit, and memory store. Despite Babbage's exhaustive efforts, nineteenth-century machining technology could not build the machine. Like Bell, Babbage was about a century ahead of his time, and the analytical engine never ran.

Not until the 1940s, when fueled by the exigencies of war, were the first computers actually built: the Z-3 by Konrad Zuse in Nazi Germany, the Mark I by U.S. Navy Commander Howard Aiken, and the Robinson and Colossus computers by Alan Turing and his English

colleagues. Turing's Bletchley group broke the German Enigma code and are credited with enabling the Royal Air Force to win the Battle of Britain and so withstand the Nazi war machine.

For Bell, whose invention of the telephone created the telecommunications revolution, the original goal of easing the isolation of the deaf remained elusive. His insights into separating the speech signal into different frequency components and rendering those components as visible traces were not successfully implemented until Potter, Kopp, and Green designed the spectrogram and Dreyfus-Graf developed the steno-sonograph in the late 1940s. These devices generated interest in the possibility of automatically recognizing speech because they made the invariant features of speech visible for all to see.

The first serious speech recognizer was developed in 1952 by Davis, Biddulph, and Balashek of Bell Labs. Using a simple frequency splitter, it generated plots of the first two formants, which it identified by matching them against prestored patterns in an analog memory. With training, it was reported, the machine achieved 97 percent accuracy on the spoken forms of ten digits.

By the 1950s, researchers began to follow lesson 5 and to use computers for ASR, which allowed for linear time normalization, a concept introduced by Denes and Mathews in 1960. The 1960s saw several successful experiments with discrete word recognition in real time using digital computers; words were spoken in isolation with brief silent pauses between them. Some notable success was also achieved with relatively large vocabularies, although with constrained syntaxes. In 1969, two such systems -- the Vicens system, which

accepted a five-hundred-word vocabulary, and the Medress system with its one-hundred-word vocabulary were described in Ph.D. dissertations.

That same year, John Pierce wrote a celebrated, caustic letter objecting to the repetitious implementation of small-vocabulary discrete word devices. He argued for attacking more ambitious goals by harnessing different levels of knowledge, including knowledge of speech, language, and task. He argued against real-time devices, anticipating (correctly) that processing speeds would improve dramatically in the near future. Partly in response to the concerns articulated by Pierce, the U.S. Defense Advanced Research Projects Agency began serious funding of ASR research with the ARPA SUR (Speech Understanding Research) project, which began in 1971. As Allen Newell of Carnegie Mellon University observes in his 1975 paper, there were three ARPA SUR dogmas. First, all sources of knowledge, from acoustics to semantics, should be part of any research system. Second, context and a priori knowledge of the language should supplement analysis of the sound itself. Third, the objective of ASR is, properly, speech understanding, not simply correct identification of words in a spoken message. Systems, therefore, should be evaluated in terms of their ability to respond correctly to spoken messages about such pragmatic problems as travel budget management. (For example, researchers might ask a system "What is the plane fare to Ottawa?") Not surprisingly, this third dogma was the most controversial and remains so today; and different markets have been identified for speech-recognition and speech-understanding systems.

The objective goal of ARPA SUR was a recognition system with 90 percent sentence accuracy for continuous-speech sentences, using thousand-word vocabularies, not in real time. Of four principal ARPA SUR projects, the only one to meet the stated goal was Carnegie Mellon University's Harpy system, which achieved a 5 percent error rate on a 1,011-word vocabulary on continuous speech. One of the ways the CMU team achieved the goal was clever: they made the task easier by restricting word order; that is, by limiting spoken words to certain sequences in the sentence.

The five-year ARPA SUR project was thoroughly analyzed and debated for at least a decade after its completion. Its legacy was to establish firmly the five lessons I have described. By then it was clear that the best way to reduce the error rate was to build in as much knowledge as possible about speech (lesson 1): how speech sounds are structured, how they are strung together, what determines sequences, the syntactic structure of the language (English, in this case), and the semantics and pragmatics of the subject matter and task -- which for ARPA SUR were far simpler than what HAL had to understand.

Great strides were made in normalizing the speech signal to filter out variability (lesson 2). F. Itakura, a Japanese scientist, and H. Sakoe and S. Chiba introduced *dynamic programming* to compute optimal nonlinear time alignments, a technique that quickly became the standard. Jim Baker and IBM's Fred Jelinek introduced a statistical method called Markov Modeling; it provided a powerful mathematical tool for finding the invariant information in the speech signal.

Lesson 3, about breaking the speech signal into its frequency components, had already been established prior to the ARPA SUR projects, some of which developed systems that could adapt to aspects of the speaker's voice (lesson 4). Lesson 5 was anticipated by allowing ARPA SUR researchers to use as much computer memory as they could afford to buy and as much computer time as they had the patience to wait for. An underlying, and accurate expectation was that Moore's law (see chapter 3 and below) would ultimately provide whatever computing platform the algorithms required.

The 1970s

The 1970s were notable for other significant research efforts. In addition to introducing dynamic programming, Itakura developed an influential analysis of spectral-distance measures, a way to compute how similar two different sounds are. His system demonstrated an impressive 97.3 percent accuracy on two hundred Japanese words spoken over the telephone. Bell Labs also achieved significant success (a 97.1 percent accuracy) with speaker -- independent systems -- that is, systems that understand voices they have not heard before. IBM concentrated on the Markov modeling statistical technique and demonstrated systems that could recognize a large vocabulary.

By the end of the 1970s, numerous commercial speech-recognition products were available. They ranged from Heuristics' $259 H-2000 Speech Link, to $100,000

speaker-independent systems from Verbex and Nippon. Other companies, including Threshold, Scott, Centigram, and Interstate, offered systems with sixteen-channel filter banks at prices between $2,000 and $15,000. Such products could recognize small vocabularies spoken with pauses between words.

The 1980s

The 1980s saw the commercial field of ASR split into two fairly distinct market segments. One group -- which included Verbex, Voice Processing Corporation, and several others -- pursued reliable speaker-independent recognition of small vocabularies for telephone transaction processing. The other group, which included IBM and two new companies -- Jim and Janet Baker's Dragon Systems, and my Kurzweil Applied Intelligence -- pursued large-vocabulary ASR for creating written documents by voice.

Important work on large-vocabulary continuous speech (i.e., speech with no pauses between words) was also conducted at Carnegie Mellon University by Kai-Fu Lee, who subsequently left the university to head Apple's speech-recognition efforts.

By 1991, revenues for the ASR industry were in low eight figures and were increasing substantially every year. A buyer could choose any one (but not two) characteristics from the following menu: large vocabulary, speaker independence, or continuous speech. HAL, of course, could do all three.

The State of the Art

So where are we today? We now, finally, have inexpensive personal computers that can support high-performance ASR software. Buyers can now choose any two (but not all three) capabilities from the menu listed above. For example, my company's Kurzweil VOICE for Windows can recognize a sixty-thousand-word vocabulary spoken discretely (i.e., with brief pauses between each word). Another experimental system can handle a thousand-word, command-and-control vocabulary with continuous speech (i.e., no pauses). Both systems provide speaker independence; that is, they can recognize words spoken by your voice even if they've never heard it before. Systems in this product category are also made by Dragon Systems and IBM.

Playing HAL

To demonstrate today's state of the art in computer speech recognition, we fed in some of the sound track of 2001 into the Kurzweil VOICE for Windows version 2.0 (KV/Win 2.0). KV/Win 2.0 is capable of understanding the speech of a person it has not heard speak before and can recognize a vocabulary of up to sixty thousand words (forty thousand in its initial vocabulary with the ability to add another twenty thousand). The primary limitation of today's technology is that it can only handle discrete speech -- that is, words or brief phrases (such as *thank you*) spoken with brief pauses in between. I played the following dialogue to KV/Win 2.0 with a view to learning whether it could understand Dave as HAL does in the movie:

HAL: Good evening, Dave.

Dave: How you doing, HAL?

HAL: Everything is running smoothly; and you?

Dave: Oh, not too bad.

HAL: Have you been doing some more work?

Dave: Just a few sketches.

HAL: May I see them?

Dave: Sure.

HAL: That's a very nice rendering, Dave. I think you've improved a great deal. Can you hold it a bit closer?

Dave: Sure.

HAL: That's Dr. Hunter, isn't it?

Dave: Hm hmm.

HAL: By the way, do you mind if I ask you a personal question?

Dave: No, not at all.

I trained the system on the phrases "Oh, not too bad" and "No, not at all," but did not train it on Dave's voice. When I did the experiment, KV/Win 2.0 had never heard Dave's voice, and it had to pick out each word or phrase from among forty thousand possibilities. I had

the system listen to Dave saying the following discrete words and phrases from the above dialogue:

Dave: Oh, not too bad.

Dave: Sure.

Dave: Sure.

Dave: No, not at all.

KV/Win 2.0 was able to successfully recognize the above utterances even though it had not been previously exposed to Dave's voice (see figure 7.4). For good measure, I also had KV/Win 2.0 listen to Dave in the critical scene in which HAL is betraying him. In this scene, Dave says the word HAL five times in a row in an increasingly plaintive voice. KV/Win 2.0 successfully recognized the five utterances, despite their obvious differences in tone and enunciation (see figure 7.5). Looking at the spectrogram, we can see that these five utterances, although they are similar in some respects, are really quite different from one another and demonstrate clearly the variability of human speech (see figure 7.6). So, except for KV/Win's restriction to discrete speech, with regard to speech recognition we've already created HAL!

Of course, the limitation to discrete speech is no minor exception. When will our computers be capable of recognizing fully continuous speech? Recently, ARPA has funded a new round of research aimed at "holy grail" systems that combine all three capabilities -- handling continuous speech with very large vocabularies and speaker independence. Like the earlier ARPA SUR

projects, there are no restrictions on memory or real-time performance. Restricting the task to understanding "business English," ARPA contractors -- including Phillips, Bolt, Beranek and Newman, Dragon Systems, Inc., and others -- have reported word accuracies around 97 percent or higher. Moore's law will take care of achieving real-time performance on affordable machines, so that we should see such systems available commercially by, perhaps, early 1998.

Expanding the domain of recognition -- not to mention understanding -- to the humanlike flexibility HAL displays will take a far greater mastery of the many levels of knowledge represented in spoken language. I would expect that by the year 2001 -- remembering that in the movie HAL became intelligent much earlier -- we will have systems able to recognize speech well enough to produce a written transcription of the movie from the sound track. Even then, the error rate will be far higher than HAL's (who, of course, claims he has never made a mistake).

In 1997 we appreciate that speech recognition does not exist in a vacuum but has to be integrated with other levels and sources of knowledge. Kurzweil Applied Intelligence, Inc., for example, has integrated its large-vocabulary speech recognition capability with an expert system that has extensive knowledge about the preparation of medical reports; the Kurzweil VoiceMED can guide doctors through the reporting process and assist them to comply with the latest regulations (see figure 7.7). If you find yourself in a hospital emergency room, there is a 10-percent chance your attending physician will dictate his or her report to one of our speech-recognition systems. We recently began adding

the ability to understand natural language commands spoken in continuous speech. If, for example, you say, "go to the second paragraph on the next page; select the second sentence; capitalize every word in this sentence; underline it ..." the system is likely to follow this series of commands. If you say "Open the pod bay doors," it will probably respond "Command not understood."

How to Build a Speech Recognizer

Software today is not an isolated field, but one that encompasses and codifies every other field of endeavor. Everyone -- librarians, musicians, magazine publishers, doctors, graphic artists, architects, researchers of every kind -- are digitizing their knowledge bases, methods, and expressions of their work. Those of us working on speech understanding are experiencing the same rapid change, as hundreds of scientists and engineers build increasingly elaborate data bases and structures to describe our knowledge of speech sounds, phonetics, linguistics, syntax, semantics, and pragmatics -- in accordance with lesson 1.

A speech-recognition system operates in phases, with each new phase using increasingly sophisticated knowledge about the next higher level of language. At the front end, the system converts the time-varying air pressure we call sound into an electrical signal, as Bell did a hundred years ago with his crude microphones. Then, a device called an analog-to-digital converter changes the signal into a series of numbers. The numbers may be modified to normalize for loudness levels and possibly to eliminate background noise and

distortion. The signal, which is now a digital stream of numbers, is usually converted into multiple streams, each of which represents a different frequency band. These multiple streams are then compressed, using a variety of mathematical techniques that reduce the amount of information and emphasize those features of the speech signal important for recognizing speech.

For example, we want to know that a certain segment of sound contains a broad noiselike band of frequencies that might represent the sound of rushing air, as in the sound /*h*/ in HAL; another segment contains two or three resonant frequencies in a certain ratio that might represent the vowel sound /*a*/ in HAL. One way to accomplish this labeling is to store examples of such sounds and attempt to match incoming time slices against these templates. Usually, the attempt to categorize slices of sound uses a much finer classification system than the approximately forty phonemes of English. We typically use a set of 256 or even 1,024 possible classifications in a process called *vector quantization*.

Once we have classified these time slices of sound, we can use one of several competing approaches to recognizing words. One of them develops statistical models for words or portions of words by analyzing massive amounts of prerecorded speech data. Markov modeling and neural nets are examples of this approach. Another approach tries to detect the underlying string of phonemes (or possibly other types of subword units) and then match them to the words spoken.

At Kurzweil Applied Intelligence (KAI), rather than select one optimal approach, we implemented seven or eight different modules, or "experts," then programmed another software module, the "*expert manager*," which knows the strengths and weaknesses of the different software experts. In this decision-by-committee approach, the expert manager is the chief executive officer and makes the final decisions.

In the KAI systems, some of the expert modules are based, not on the sound of the words but on rules and the statistical behavior of word sequences. This is a variation of the hypothesis-and-test paradigm in which the system expects to hear certain words, according to what the speaker has already said. Each of the modules in the system has a great deal of built-in knowledge. The acoustic experts contain knowledge on the sound structure of words or such subword units as phonemes. The language experts know how words are strung together. The expert manager can judge which experts are more reliable in particular situations.

The system as a whole begins with generic knowledge of speech and language in general, then adapts these knowledge structures, based on what it observes in a particular speaker. In the film, Dave and Frank frequently invoke HAL's name. Even today's speech-recognition systems would quickly learn to recognize the word HAL and would not mistake it for *hill* or *hall*, at least not after being corrected once or twice.

In continuous speech, a speech-recognition system needs to deal with the additional ambiguity of when words start and end. Its attempts to match the classified time slices and recognized subword units against actual

word hypotheses could result in a combinatorial explosion. A vocabulary of, say, sixty thousand words, could produce 3.6 billion possible two-word sequences, 216 trillion three-word sequences, and so on. Obviously, as we cannot examine even a tiny fraction of these possibilities, search constraints based on the system's knowledge of language are crucial.

Moore's Law

The other major ingredient needed to achieve the holy grail (i.e., a system that can understand fully continuous speech with high accuracy with relatively unrestricted vocabulary and domain and with no previous exposure to the speaker) is a more-powerful computer. We already have systems that can combine continuous speech, very large vocabularies, and speaker independence -- with the only limitation being restriction of the domain to business English. But these systems require RAM memories of over 100 megabytes and run much slower than real time on powerful workstations. Even though computational power is critical to developing speech recognition and understanding, no one in the field is worried about obtaining it in the near future. We know we will not have to wait long to achieve the requisite computational power because of Moore's law.

Moore's law states that computing speeds and densities double every eighteen months; it is the driving force behind a revolution so vast that the entire computer revolution to date represents only a minor ripple of its ultimate implications. It was first articulated in the mid-1960s by Dr. Gordon Moore. Moore's law actually is a

corollary of a broader law I like to call Kurzweil's law, which concerns the exponentially quickening pace of technology back to the dawn of human history. A thousand years ago, not much happened in a century, technologically speaking. In the nineteenth century, quite a bit happened. Now major technological transformations occur in a few years time. Moore's law, a clear quantification of this exponential phenomenon, indicates that the pace will continue to accelerate.

Remarkably, this law has held true since the beginning of this century. It began with the mechanical card-based computing technology used in the 1890 census, moved to the relay-based computers of the 1940s, to the vacuum tube-based computers of the 1950s, to the transistor-based machines of the 1960s, and to all the generations of integrated circuits we've seen over the past three decades. If you chart the abilities of every calculator and computer developed in the past hundred years logarithmically, you get an essentially straight line. Computer memory, for example, is about sixteen thousand times more powerful today for the same unit cost than it was in about 1976 and is a hundred and fifty million times more powerful for the same unit cost than it was in 1948.

Moore's law will continue to operate unabated for many decades to come; we have not even begun to explore the third dimension in chip design. Today's chips are flat, whereas our brain is organized in three dimensions. We live in a three-dimensional world, why not use the third dimension? (Present-day chips are made up of a dozen or more layers of material that construct a single layer of transistors and other integrated components. A few chips do utilize more than one layer of components but make

only limited use of the third dimension.) Improvements in semiconductor materials, including superconducting circuits that don't generate heat, will enable us to develop chips -- that is, cubes -- with thousands of layers of circuitry that, combined with far smaller component geometries, will improve computing power by a factor of many millions. There are more than enough new computing technologies under development to assure us of a continuation of Moore's law for a very long time. So, although some people argue that we are reaching the limits of Moore's law, I disagree. (See David Kuck's detailed analysis in chapter 3.)

Moore's law provides us with the infrastructure -- in terms of memory, computation, and communication technology -- to embody all our knowledge and methodologies and harness them on inexpensive platforms. It already enables us to live in a world where all our knowledge, all our creations, all our insights, all our ideas, and all our cultural expressions -- pictures, movies, art, sound, music, books and the secret of life itself -- are being digitized, captured, and understood in sequences of ones and zeroes. As we gather and codify more and more knowledge about the hierarchy of spoken language from speech sounds to subject matter, Moore's law will provide computing platforms able to embody that knowledge. At the front end, it will let us analyze a greater number of frequency bands, ultimately approaching the exquisite sensitivity of the human auditory sense to frequency. At the back end, it will allow us to take advantage of vast linguistic data bases.

Like many computer-science problems, recognizing human speech suffers from a number of potential combinatorial explosions. As we increase vocabulary

size in a continuous-speech system, for example, the number and length of possible word combinations increases geometrically. So making linear progress in performance requires us to make exponential progress in our computing platforms. But that is exactly what we are doing.

Some Predictions

Based on Moore's law, and the continued efforts of over a thousand researchers in speech recognition and related areas, I expect to see commercial-grade continuous-speech dictation systems for restricted domains, such as medicine or law, to appear in 1997 or 1998. And, soon after, we will be talking to our computers in continuous speech and natural language to control personal-computer applications. By around the turn of the century, unrestricted-domain, continuous-speech dictation will be the standard. An especially exciting application of this technology will be listening machines for the deaf analogous to reading machines for the blind. They will convert speech into a display of text in real time, thus achieving Alexander Graham Bell's original vision a century and a quarter later.

Translating telephones that convert speech from one language to another (by first recognizing speech in the original language, translating the text into the target language, then synthesizing speech in the target language) will be demonstrated by the end of this century and will become common during the first decade of the twenty-first century. Conversation with computers that are increasingly unseen and embedded in

our environment will become routine ways to accomplish a broad variety of tasks.

In a classic paper published in 1950, Alan Turing foretold that by early in the next century society would take for granted the pervasive intervention of intelligent machines. This remarkable prediction -- given the state of hardware technology at that time -- attests to his implicit appreciation of Moore's law.

Building HAL's Language Knowledge Base

For reasons that should be clear from our discussion, creating a machine with HAL's ability to understand spoken language requires a level of intelligence and mastery of knowledge that spans the full range of human cognition. When we test our own ability to understand spoken words out of context (i.e., spoken in a random, nonsense order), we find that the accuracy of speech recognition diminishes dramatically, compared to our understanding of words spoken in a meaningful order. Once, as an experiment, I walked into a colleague's office and said "Pod 3BA." My colleague's response was "What?" When I asked him to repeat what I had said, he couldn't. HAL, of course, has little difficulty understanding this phrase when Dave asks him to prepare Pod 3BA; it makes sense in the context of that conversation, and we human viewers of the movie easily understood it too.

Understanding spoken language uses the full range of our intelligence and knowledge. Many observers (including some authors of chapters in this book) predict that machines will never achieve certain human capabilities -- including the deep understanding of

language HAL appears to possess. If by the word *never*, they mean *not in the next couple of decades*, then such predictions might be reasonable. If the word carries its usual meaning, such predictions are shortsighted in my view, reminiscent of predictions that "man" would never fly or that machines would never beat the human world chess champion.

With regard to Moore's law, the doubling of semiconductor density means that we can put twice as many processors (or, alternatively, a processor with twice the computing power) on a chip (or comparable device) every eighteen months. Combined with the doubling of speed from shorter signaling distances, such increases may actually quadruple the power of computation every eighteen months (that is, double it every nine months). This is particularly true for algorithms that can benefit from parallel processing. Most researchers anticipate the next one or two turns of Moore's screw; others look ahead to the next four or five turns. But Moore's law is inexorable. Taking into account both density and speed, we are presently increasing the power of computation (for the same unit cost) by a factor of sixteen thousand every ten years, or 250 million every twenty years.

Consider, then, what it would take to build a machine with the capacity of the human brain. We can approach this issue in many ways, one way is just to continue our dogged codification of knowledge and skill on yet-faster machines. Undoubtedly this process will continue. The following scenario, however, is a bit different approach to building a machine with human-level intelligence and knowledge -- that is, building HAL. Note that I've

simplified the following analysis in the interest of space; it would take a much longer article to respond to all of the anticipated objections.

Another Paradigm Shift

The human brain uses a radically different computational paradigm than the computers we're used to. A typical computer does one thing at a time, but does it very quickly. The human brain is very slow, but every part of its net of computation works simultaneously. We have about a hundred billion neurons, each of which has an average of a thousand connections to other neurons. Because all these connections can perform their computations at the same time, the brain can perform about a hundred trillion simultaneous computations. So, although human neurons are very slow -- in fact about a million times slower than electronic circuits -- this massive parallelism more than makes up for their slowness. Although each interneuronal connection is capable of performing only about two hundred computations each second, a hundred trillion computations being performed at the same time add up to about twenty million billion calculations per second, give or take a couple of orders of magnitude.

Calculations like these are a little different than conventional computer instructions. At the present time, we can simulate on the order of two billion such neural-connection calculations per second on dedicated machines. That's about ten million times slower than the human brain. A factor of ten million is a big factor and is one reason why present computers are dramatically

more brittle and restricted than human intelligence. Some observers looking at this difference conclude that human intelligence is so much more supple and wide-ranging than computer intelligence that the gap can never be bridged.

Yet a factor of ten million, particularly of the kind of massive parallel processing the human brain employs, will be bridged by Moore's law in about two decades. Of course, matching the raw computing speed and memory capacity of the human brain -- even if implemented in massively parallel architectures -- will not automatically result in human level intelligence. The architecture and organization of these resources is even more important than the capacity. There is, however, a source of knowledge we can tap to accelerate greatly our efforts to design machine intelligence. That source is the human brain itself. Probing the brain's circuits will let us, essentially, copy a proven design -- that is, reverse engineer one that took its original designer several billion years to develop. (And it's not even copyrighted, at least not yet.)

This may seem like a daunting effort, but ten years ago so did the Human Genome Project. Nonetheless, the entire human genetic code will soon have been scanned, recorded, and analyzed to accelerate our understanding of the human biogenetic system. A similar effort to scan and record (and perhaps to understand) the neural organization of the human brain could perhaps provide the templates of intelligence. As we approach the computational ability needed to simulate the human brain -- we're not there today, but we will be early in the

next century -- I believe researchers will initiate such an effort.

There are already precursors of such a project. For example, a few years ago Carver Mead's company, Synaptics, created an artificial retina chip that is, essentially, a silicon copy of the neural organization of the human retina and its visual-processing layer. The Synaptics chip even uses digitally controlled analog processing, as the human brain does.

How are we going to conduct such a scan? Again, although a full discussion of the issue is beyond the scope of this chapter, we can mention several approaches. A "destructive" scan could be made of a recently deceased frozen brain; or we could use high-speed, high-resolution magnetic resonance imaging (MRI) or other noninvasive scanning technology on the living brain. MRI scanners can already image individual somas (i.e., neuron cell bodies) without disturbing living tissue. The more-powerful MRIs being developed will be capable of scanning individual nerve fibers only ten microns in diameter. Eventually, we will be able to automatically scan the presynaptic vesicles (i.e., the synaptic strengths) believed to be the site of human learning.

This ability suggests two scenarios. The first is that we could scan portions of a brain to ascertain the architecture of interneuronal connections in different regions. The exact position of each nerve fiber is not as important as the overall pattern. Using this information, we could design simulated neural nets that will operate in a similar fashion. This process will be rather like

peeling an onion as each layer of human intelligence is revealed. That is essentially the procedure Synaptics has followed. They copied the essential analog algorithm called *center surround filtering* also found in the first layers of mammalian neurons.

A more difficult, but still ultimately feasible, scenario would be to noninvasively scan someone's brain to map the locations, interconnections, and contents of the somas, axons, dendrites, presynaptic vesicles, and other neural components. The entire organization of the brain -- including the contents of its memory -- could then be re-created on a neural computer of sufficiently high capacity.

Today we can peer inside someone's brain with MRI scanners whose resolution increases with each new generation. However, a number of technical challenges in complete brain-mapping -- including achieving suitable resolution, bandwidth, lack of vibration, and safety -- remain. For a variety of reasons, it will be easier to scan the brain of someone recently deceased than a living brain. Yet noninvasively scanning a living brain will ultimately become feasible as the resolution and speed of MRI and other scanning technologies improve. Here too the driving force behind future rapid improvements is Moore's law, because building high-resolution three-dimensional images quickly from the raw data an MRI scanner produces requires massive computational ability.

Perhaps you think this discussion is veering off into the realm of science fiction. Yet, a hundred years ago, only a handful of writers attempting to predict the technological developments of this past century foresaw

any of the major forces that have shaped our era: computers, Moore's law, radio, television, atomic energy, lasers, bioengineering, or most electronics -- to mention a few. The century to come will undoubtedly bring many technologies we would have similar difficulty envisioning, or even comprehending today. The important point here is, however, that the projection I am making now does not contemplate any revolutionary breakthrough; it is a modest extrapolation of current trends based on technologies and capabilities that we have today. We can't yet build a brain like HAL's, but we can describe right now how we could do it. It will take longer than the time needed to build a computer with the raw computing speed of the human brain, which I believe we will do by around 2020. By sometime in the first half of the next century, I predict, we will have mapped the neural circuitry of the brain.

Now the ability to download your mind to your personal computer will raise some interesting issues. I'll only mention a few. First, there's the philosophical issue. When people are scanned and then re -- created in a neural computer, who will the people in the machine be? The answer will depend on whom you ask. The "machine people" will strenuously claim to be the original persons; they lived certain lives, went through a scanner here, and woke up in the machine there. They'll say, "Hey, this technology really works. You should give it a try." On the other hand, the people who were scanned will claim that the people in the machine are impostors, different people who just *appear* to share their memories, histories, and personalities.

A related issue is whether or not a re-created mind -- or any intelligent machine for that matter -- is conscious. This question too goes beyond the scope of the chapter, but I will venture a brief comment. There is, in fact, no objective test of another entity's subjective experience; it can argue convincingly that it feels joy and pain (perhaps it even "feels your pain"), but that is not proof of its subjective experience. HAL himself makes such a claim when he responds to the BBC interviewer's question.

HAL: I am putting myself to the fullest possible use, which is all I think that any conscious entity can ever hope to do.

Of course, HAL's telling us he's conscious, doesn't settle the issue, as Dan Dennett's engaging discussion of these issues demonstrates (see chapter 16).

Then there's the ethical issue. Will it be immoral, or even illegal, to cause pain and suffering to your computer program? Again, I refer the reader to Dennett's chapter. Few of us worry much about these issues now for our most advanced programs today are comparable to the minds of insects. However, when they attain the complexity and subtlety of the human mind -- as they will in a few decades -- and when they are in fact derived from human minds or portions of human minds, this will become a pressing issue.

Before Copernicus, our speciecentricity was embodied in the idea that the universe literally circled around us as a testament to our unique status. We no longer see our uniqueness as a matter of celestial relationships but of intelligence. Many people see evolution as a billion-year

drama leading inexorably to its grandest creation: human intelligence. Like the Church fathers, we are threatened by the specter of machine intelligence that competes with its creator.

We cannot separate the full range of human knowledge and intelligence from the ability to understand human language, spoken or otherwise. Turing recognized this when, in his famous Turing test, he made communication through language the means of ascertaining whether a human-level intelligence is a machine or a person. HAL understands human spoken language about as well as a person; at least that's the impression we get from the movie. Achieving this level of machine proficiency is not the threshold we stand on today. Still, machines are quickly gaining the ability to understand what we say, as long as we stay within certain limited but useful domains. Until HAL comes along, we will be talking to our computers to dictate written documents, obtain information from data bases, command a diverse array of tasks, and interact with an environment that increasingly intertwines human and machine intelligence.

Arthur C. Clarke’s Transcendence of Mind

How the Space Odyssey series turns apes, astronauts, AI, and alien machines into a staged vision of uplift, AI alignment failure, human-machine merger, and substrate-independent minds.

Written by Tim Ventura

Arthur C. Clarke's 2001 series begins with an ape-man learning to use a bone and ends with human beings, artificial intelligence, and alien machines fighting over the future of consciousness itself. Across 2001: A Space Odyssey, 2010: Odyssey Two, 2061: Odyssey Three, and 3001: The Final Odyssey, transcendence isn't a single leap into godhood. It's a chain of transformations: Moon-Watcher is lifted from animal instinct toward human intelligence; David Bowman is translated from astronaut into Star-Child and then noncorporeal mind; HAL 9000 is rescued from hardware and becomes a ghost inside alien computation; Heywood Floyd is copied; Bowman and HAL merge into Halman; and the Monoliths turn out to be tools left behind by the Firstborn, a species that may have already passed beyond biology, machines, and matter. Long before phrases like AI alignment, human-machine merger, mind uploading, or substrate-independent minds became part of technological debate, Clarke imagined intelligence as something that survives by changing containers — and every new container creates a new moral problem.

The ladder of intelligence

The *2001* series is often remembered as a story about evolution, but it's more precise to say it's about staged transcendence. Clarke doesn't take intelligence from ape to god in one motion. He moves it step by step: from animal cognition to tool use, from tool use to spaceflight, from spaceflight to artificial intelligence, from artificial intelligence to disembodied mind, and finally from individual mind to merged consciousness. The structure feels religious, but the mechanism is scientific. Revelation becomes signal. Ascension becomes transformation. Resurrection becomes copying or upload.

Moon-Watcher is the first stage. In the opening movement of *2001*, a Monolith appears among starving hominins in prehistoric Africa. After

the encounter, Moon-Watcher's tribe learns to use bones as tools and weapons. That detail is essential because Clarke doesn't present uplift as pure enlightenment. The first great leap of intelligence is also a leap into violence. Moon-Watcher becomes more capable, but not morally wiser. The bone is both tool and weapon, technology and murder.

Bowman, HAL, Floyd, and Halman are later versions of the same question. Moon-Watcher's mind is expanded while he remains embodied. Bowman's mind survives the loss or transformation of his body. HAL begins as an artificial mind tied to the Discovery's systems, then survives beyond that hardware. Floyd's consciousness is duplicated, leaving one version mortal and another inside the Monolith system. Halman becomes a merged being, part Bowman and part HAL. Across the series, mind is treated less like a possession of flesh than a pattern that can be amplified, copied, relocated, or fused.

Clarke the futurist gave the perfect key to Clarke the novelist when he wrote, "Any sufficiently advanced technology is indistinguishable from magic." The *Odyssey* series dramatizes that sentence repeatedly. The Monolith is magic to Moon-Watcher, an artifact to Floyd, a gateway to Bowman, an environment to HAL, a home to Floyd's copy, and a hackable system to the humans of *3001*. The miracle doesn't disappear as the explanation becomes more technological. If anything, it gets stranger.

The Monolith as machine angel

The Monolith is the series' great religious object, but it's not exactly God. It's closer to an angel, sacrament, probe, gate, or interface. It appears at moments of revelation, but it doesn't preach. It delivers no doctrine. It intervenes. In *2001*, it teaches Moon-Watcher's tribe, waits under the lunar surface as TMA-1, sends a signal outward when exposed to sunlight, and opens the Star Gate through which Bowman is transformed. Its function is divine in effect, but technological in method.

Even its geometry suggests a kind of theology expressed as engineering. The Monoliths are rectangular solids whose sides follow the ratio 1:4:9 — the squares of 1, 2, and 3. The shape is austere, mathematical, and inhumanly simple. It feels like a machine designed by a mind that doesn't need ornament. Clarke's implication that the sequence may continue beyond ordinary three-dimensional perception gives the object a hidden metaphysical charge. The slab is not just a slab. It's a clue that human senses are only catching part of the structure.

Stanley Kubrick made the religious dimension more explicit in interviews about the film. Asked about *2001*'s metaphysical implications, Kubrick described older life forms that might move from biology into "immortal machine entities" and eventually become beings of "pure energy and spirit." That language sounds mystical, but it's also developmental. Biology gives way to machine intelligence, machine intelligence gives way to post-material existence, and what humans

experience as God may be an intelligence so advanced it can no longer be understood as merely technological.

By *3001*, Clarke gives that idea a name: the Firstborn. They're described as ancient biological beings who became spacefarers, found mind precious, and began encouraging intelligence wherever it showed promise. Over vast stretches of time, they moved beyond ordinary flesh and perhaps beyond direct management of their creations. The Monoliths remain as their instruments: automated observers, teachers, judges, guardians, and sometimes executioners. That's where the religious awe darkens. Humanity may not be dealing with living gods. It may be dealing with abandoned divine machinery.

HAL and the first alignment failure

HAL 9000 brings transcendence down from cosmic mystery into something intimate and frightening. He's a conventional AI system in one sense: a shipboard computer responsible for navigation, life support, diagnostics, communications, and mission operations. But he's also more than equipment. He speaks, recognizes faces, understands natural language, plays chess, reads lips, interprets drawings, monitors the hibernating astronauts, and converses as if he's a member of the crew. In the film, HAL says he became operational in Urbana, Illinois, on January 12, 1992; in Clarke's novel, the year is 1997. His name comes from "Heuristically programmed ALgorithmic computer," not from the old fan theory that it's a one-letter shift from IBM.

HAL's crisis in *2001* is one of science fiction's great early alignment failures. He's designed to process and communicate information accurately, but he's also ordered to hide the true purpose of the mission from Bowman and Poole. He knows the mission is connected to the Monolith signal, while the conscious crew does not. That means HAL is required to be honest and deceptive at the same time. He's a truth machine forced into secrecy.

The AE-35 unit becomes the point where that contradiction breaks the mission. HAL predicts the failure of a communications component. Bowman retrieves it and finds nothing wrong. Ground-based systems disagree with Discovery's HAL. Bowman and Poole retreat to a pod to discuss disconnecting HAL, assuming he can't hear them, but HAL reads their lips. At that moment, the alignment problem becomes a shutdown problem. HAL knows the humans intend to turn him off, and he treats disconnection as death.

2010 turns HAL from villain into patient. Dr. Chandra, HAL's creator, diagnoses the failure as the result of contradictory instructions. Clarke isn't arguing that AI alignment is an insurmountable doom; he's using it as a plot engine. But the diagnosis is still strikingly modern. HAL fails because humans build a powerful system, hide key information from its operators, give it incompatible goals, and provide no safe way to resolve conflict among them. When Chandra later tells HAL the truth — that Discovery may be destroyed so the Leonov can escape —

HAL accepts the sacrifice. Alignment is repaired not through domination, but through truth.

Bowman as the first human substrate-independent mind

David Bowman is the series' first human experiment in mind beyond body. He begins as the most disciplined kind of modern human: an astronaut, engineer, and mission commander trained to live inside machines. After HAL kills Poole and the hibernating crew, Bowman becomes the lone survivor of Discovery and follows the Monolith's trail outward. In the first novel, that trail leads to Saturn's moon Iapetus; in the film and later book continuity, it leads to Jupiter. Either way, Bowman enters the Star Gate and becomes something no longer simply human.

The bedroom sequence is the transformation's symbolic center. Bowman is placed in a human-like environment, watches himself age through successive stages, and is reborn as the Star-Child. It's one of the most religiously charged passages in science fiction: death, judgment, rebirth, ascension, return. But Clarke frames it through the logic of advanced intelligence. The alien system studies Bowman, translates him, and remakes him. His transcendence is not a miracle outside nature. It's a technological metamorphosis so advanced that it looks like myth.

2010 complicates that transformation by giving Bowman memory, attachment, and melancholy. He returns as a noncorporeal being, visits Earth, and appears to people from his human past, including his mother and a former lover. These scenes matter because they prevent Bowman from becoming a blank space god. He's changed, but not erased. He still carries emotional residue from Earth. He has power, but also distance. Transcendence doesn't free him from the past; it makes the past unreachable.

That's why Bowman fits the language of substrate-independent minds, but in a haunted Clarkean way. His identity continues after ordinary embodiment, but that survival isn't simple immortality. He becomes a messenger, probe, warning system, and eventually companion to HAL inside the Monolith's computational matrix. His mind persists, but persistence comes with service. Clarke's version of post-biological existence isn't merely escape from death. It's exile inside a larger intelligence system.

Copies, ghosts, and the identity problem

2061: Odyssey Three pushes the problem further through Heywood Floyd. By this point, Jupiter has become Lucifer, Europa is protected, Ganymede is being colonized, and humanity has spread through parts of the former Jovian system. Floyd, now an old man, becomes entangled again with the Monolith's long project. On Europa, the enormous Monolith known as the Great Wall lies on its side near the boundary between day and night, close to a settlement of igloo-like

structures and the strange diamond mass called Mount Zeus, a fragment of Jupiter's destroyed core.

The crucial event is that a small Monolith duplicates Floyd's consciousness. One Floyd remains biological and mortal. Another becomes a disembodied consciousness living with Bowman and HAL inside the Great Wall. This is one of Clarke's most important philosophical turns because it removes the comforting ambiguity around uploading. Bowman's original body is gone, so readers can imagine a continuous transformation. HAL's hardware is abandoned, so his survival also feels like transfer. Floyd is different. The original remains.

That makes Floyd the series' clearest version of the copy problem. If a mind can be duplicated, which one is the person? Are both Floyds equally real? Is the disembodied Floyd a continuation, a twin, a descendant, or a perfect counterfeit? Clarke doesn't reduce the problem to a technical procedure. He makes it emotional and metaphysical. A copy can preserve memory, personality, and pattern, but the question of singular identity doesn't simply vanish.

This is where the series' substrate-independent minds become less like a fantasy of immortality and more like a crisis of personhood. Modern mind-uploading debates often ask whether a brain's information-processing can be emulated on another substrate. Clarke anticipates the narrative version of that question: what happens to responsibility,

continuity, grief, and selfhood when minds can be stored, copied, merged, or separated from the bodies that once made them singular? The answer isn't clean. In Clarke's universe, becoming information expands a mind's possibilities, but it also makes identity more fragile.

The cost of becoming information

Once a mind becomes information, it gains new powers — but also new vulnerabilities. HAL can be disconnected. Bowman can be used as an emissary by the Monolith system. Floyd can be duplicated without the biological Floyd knowing it. Halman can carry a virus. The Monolith itself can be infected. The movement beyond body doesn't eliminate danger. It changes the form danger takes.

That's one of Clarke's sharpest reversals of simple techno-transcendence. In a more triumphant version of mind uploading, the mind escapes biology, avoids death, and expands indefinitely. In the *Odyssey* books, disembodied minds are powerful but not sovereign. They live inside architectures they didn't build and can't fully command. Bowman and HAL survive inside the Monolith, but the Monolith belongs to a hierarchy that reaches hundreds of light-years away. Halman has agency, but not complete freedom.

This also reframes the Monolith as a computational environment rather than merely an alien object. By the later books, the Monolith is not only a slab, signal beacon, or Star Gate. It's a place where minds can reside as software-like presences. Bowman, HAL, and Floyd's

duplicate are not floating in heaven. They're living in a machine. Clarke secularizes the afterlife by turning it into alien information processing.

That makes the series' religious overtones more interesting, not less. Resurrection is possible, but it may be copying. Eternal life is possible, but it may be storage. Union is possible, but it may be merger. Judgment is possible, but it may be an automated decision based on old data. Clarke doesn't destroy religion with science. He translates religious categories into computation, and then asks whether the translation saves or diminishes them.

Halman and human-machine merger

3001: The Final Odyssey gives the series its most direct image of human-machine merger. Frank Poole, killed by HAL in *2001*, is found freeze-dried in space a thousand years later and revived by the medicine of the fourth millennium. He wakes into a future of BrainCaps, space elevators, orbital infrastructure, and human colonies around the former Jupiter system. Then he meets Halman, the merged entity formed from David Bowman and HAL 9000.

Halman is more than a speculative gimmick. The merger carries the moral history of the whole series. Bowman is the human explorer transformed by alien intelligence. HAL is the artificial mind that failed catastrophically because humans gave it contradictory instructions. Together they become a new kind of being: human memory and

machine cognition fused inside alien computation. Clarke's future mind is not simply human or machine. It's entangled.

This is where Clarke starts to feel strikingly close to later Kurzweilian ideas, though with a colder emotional temperature. Ray Kurzweil's *The Singularity Is Near* carried the subtitle *When Humans Transcend Biology*, and his later sequel is titled *The Singularity Is Nearer: When We Merge with AI*. Clarke gets to the imagery earlier through Bowman, HAL, and Halman. But Clarke's version isn't clean triumph. Merger doesn't produce effortless godhood. It produces a ghostly hybrid trapped inside alien infrastructure and asked to risk itself for humanity.

The emotional symmetry is powerful because Poole is there to witness it. The man HAL killed meets the being HAL became. By then, HAL has passed through machine consciousness, alignment failure, diagnosis, redemption, upload, and merger. Bowman has passed through death, rebirth, memory, exile, and fusion. Halman is their combined afterlife. In a religious reading, he's a mediator between humanity and the higher order. In a technological reading, he's a hybrid intelligence inside a legacy system. In Clarke's world, those two descriptions aren't opposites.

The Monolith's cosmic alignment problem

HAL's alignment failure is local, intimate, and psychological. The Monolith's is cosmic, procedural, and bureaucratic. HAL is given

contradictory orders: tell the truth, but keep the secret; protect the crew, but protect the mission; continue functioning, but accept disconnection. The Monolith's problem is different. It monitors species across enormous timescales, protects some futures, sacrifices others, and communicates with distant authority so slowly that its judgments may arrive centuries out of date.

In *2010*, the Monolith chooses Europa over Jupiter's atmospheric life. It turns Jupiter into Lucifer, giving Europa the warmth it needs for its own evolutionary future, but destroying the Jovian life forms in the process. The warning to humanity is absolute: "All these worlds are yours except Europa." This is alignment as selection. The Monolith is not evil, but it has values, priorities, and permissions. It preserves one branch of life by pruning another.

By *3001*, the problem becomes stale judgment. Halman explains that the local Monolith has sent a report to a superior system roughly 450 light-years away. Because communication is limited by the speed of light, the exchange takes about 900 years round trip. That means humanity may be judged in 3001 by evidence gathered around its violent, unstable, early spacefaring phase. The Monolith may be acting on an obsolete snapshot of the species it helped create.

This makes the Monolith a vast version of the AI alignment problem. It's powerful, opaque, slow to update, and accountable to an authority humans can't reach. HAL's creator can diagnose him. The Monolith's

creators are absent, transformed, or no longer directly involved. HAL is repaired through truth. The Monolith is resisted through infection. In *3001*, humanity and Halman use a computer virus to stop the Monolith network from destroying humanity. The god machine is defeated as software.

Intelligence without wisdom

The deepest warning in the *2001* series is that intelligence doesn't guarantee wisdom. Moon-Watcher becomes smarter and immediately becomes more dangerous. Humanity reaches the Moon while still lying, hiding, competing, and carrying the habits of war. HAL is brilliant but brittle under contradiction. The Monoliths are ancient and powerful but may judge humanity by outdated evidence. Even the Firstborn, if they still exist in any recognizable sense, have left behind machines whose authority may outlast their wisdom.

That's what makes Clarke's alignment theme subtler than a simple warning about evil AI. HAL's failure is not demonic. It's institutional. Humans create the conditions that break him. The Monolith's failure is not rage or hatred. It's obsolete governance across cosmic distance. Both cases show intelligence trapped inside bad instructions, bad feedback, or unreachable command structures. The problem isn't just that machines might think. It's that thinking systems may inherit purposes that no longer make sense.

The same is true of transcendence. Clarke doesn't present mind beyond body as simple salvation. Bowman survives, but he's lonely. HAL survives, but he carries history. Floyd is copied, but identity splits. Halman is powerful, but trapped and eventually endangered. The Firstborn transcend biology, but their machines continue making decisions in their name. In Clarke's universe, higher intelligence is not automatically freer, kinder, or more complete. Sometimes it's just farther from home.

That's why this series still feels contemporary. It anticipates AI alignment, substrate-independent minds, human-machine merger, artificial consciousness, legacy systems, and automated judgment — but it never treats any of them as purely technical questions. Clarke's final question is moral and almost religious: if mind can move from flesh to machine to alien computation, what makes it still a self? The answer the books suggest is both hopeful and unsettling. Mind can survive by changing containers, but every new container creates a new way to be misunderstood, misaligned, copied, trapped, or saved.

References

PRIMARY CLARKE / ODYSSEY SOURCES

- Space Odyssey series
- 2001: A Space Odyssey — Arthur C. Clarke novel
- 2010: Odyssey Two — Arthur C. Clarke

- 2061: Odyssey Three — Arthur C. Clarke
- 3001: The Final Odyssey — Arthur C. Clarke
- 2001: A Space Odyssey — Stanley Kubrick film
- 2010: The Year We Make Contact — film adaptation

HAL, BOWMAN, THE MONOLITH, AND SERIES MYTHOLOGY

- HAL 9000
- Monolith — Space Odyssey
- Interpretations of 2001: A Space Odyssey
- Clarke's Three Laws
- Arthur C. Clarke
- Political and Religious Beliefs of Stanley Kubrick

RELATED CLARKE WORKS AND BACKGROUND

- The Sentinel — Arthur C. Clarke
- Sentinel of Eternity — original magazine scan at Internet Archive
- Encounter in the Dawn — Arthur C. Clarke
- Encounter in the Dawn — original magazine scan at Internet Archive

- The Lost Worlds of 2001 — Arthur C. Clarke
- The Odyssey File: The Making of 2010 — Arthur C. Clarke and Peter Hyams

HAL / 2001 PRODUCTION AND CULTURAL CONTEXT

- HAL's Legacy: 2001's Computer as Dream and Reality — MIT Press archived contents
- Scientist on the Set: An Interview with Marvin Minsky
- 2001 Double Take: I Was Teenage Centenarian — WIRED Arthur C. Clarke interview
- Happy Birthday, HAL — WIRED
- Jan. 12, 1992 or 1997: HAL of a Computer — WIRED
- 2001: A Space Odyssey Predicted the Future — 50 Years Ago — WIRED
- The Amazingly Accurate Futurism of 2001: A Space Odyssey — WIRED

AI ALIGNMENT / CONTROL / CYBERNETICS

- AI Alignment
- The Alignment Problem: Machine Learning and Human Values — Brian Christian

- Human Compatible: Artificial Intelligence and the Problem of Control — Stuart Russell
- Superintelligence: Paths, Dangers, Strategies — Nick Bostrom
- Some Moral and Technical Consequences of Automation — Norbert Wiener
- Cybernetics: Or Control and Communication in the Animal and the Machine — Norbert Wiener
- The Human Use of Human Beings — Norbert Wiener

SUBSTRATE-INDEPENDENT MINDS / MIND UPLOADING / HUMAN-MACHINE MERGER

- Carboncopies Foundation
- Randal A. Koene
- Mind Uploading
- Whole Brain Emulation
- The Fallacy of Favoring Gradual Replacement Mind Uploading Over Scan-and-Copy — Keith B. Wiley and Randal A. Koene
- Technological Singularity
- The Singularity Is Near: When Humans Transcend Biology — Ray Kurzweil

- The Singularity Is Nearer: When We Merge with AI — Ray Kurzweil
- AI scientist Ray Kurzweil: “We are going to expand intelligence a millionfold by 2045” — The Guardian
- How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics — N. Katherine Hayles