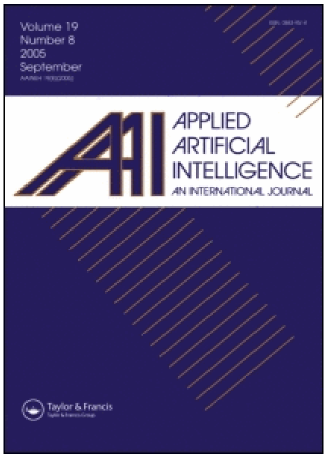

Applied Artificial Intelligence
An International Journal
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713191765

AI TURNS FIFTY: REVISITING ITS ORIGINS

To cite this Article: , 'AI TURNS FIFTY: REVISITING ITS ORIGINS', Applied Artificial Intelligence, 21:4, 259 - 279
To link to this article: DOI: 10.1080/08839510701252304
URL: http://dx.doi.org/10.1080/08839510701252304

Applied Artificial Intelligence, 21:259–279

ISSN: 0883-9514 print/1087-6545 online
DOI: 10.1080/08839510701252304

Taylor & Francis
Taylor & Francis Group

AI TURNS FIFTY: REVISITING ITS ORIGINS

Roberto Cordeschi □ *Department of Philosophical and Epistemological Research, University of Rome "La Sapienza," Italy*

□ *The expression "artificial intelligence" (AI) was introduced by John McCarthy, and the official birth of AI is unanimously considered to be the 1956 Dartmouth Conference. Thus, AI turned fifty in 2006. How did AI begin? Several differently motivated analyses have been proposed as to its origins. In this paper a brief look at those that might be considered steps towards Dartmouth is attempted, with the aim of showing how a number of research topics and controversies that marked the short history of AI were touched on, or fairly well stated, during the year immediately preceding Dartmouth. The framework within which those steps were taken was the development of digital computers. Earlier computer applications in areas such as complex decision making and management, at that time dealt with by operations research techniques, were important in this story. The time was ripe for AI's intriguingly tumultuous development, marked as it has been by hopes and defeats, successes and difficulties.*

As is well known, the 1956 summer Dartmouth Conference on AI was preceded by a preparatory document dated August 31, 1955, whose authors were John McCarthy, Marvin Minsky, Nathaniel Rochester and Claude Shannon. The meeting's aim was to examine "the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it," as one reads in the document (McCarthy et al. 1955). Some of the main pioneers in computer programming were present at Dartmouth, such as Allen Newell, Arthur Samuel, Oliver Selfridge, and Herbert Simon. After Dartmouth, the historical centers of AI research would be formed: at Carnegie-Mellon University with Newell and Simon, at MIT with Minsky, and at Stanford

This paper is based on a talk given at the 9th Congress of AI*IA, Milan, September 21–23, 2005. I would like to thank Stefania Bandini and Luigia Carlucci Aiello for inviting me. I had the opportunity to present an analogous talk in other places, and to further discuss the main topics of the present paper. I would like to express my thanks to Floriana Esposito and Luciano Floridi (University of Bari), Maurizio Dapor, Ernesto D'Avanzo, and Oliviero Stock (ITC-irst, Trento). Last, but not least, thanks to Marco Colombetti and Giuseppe Trautteur for helpful suggestions.

Address correspondence to Roberto Cordeschi, Dipartimento di Studi Filosofici ed Epistemologici, Università di Roma "La Sapienza," Via Carlo Fea 2, I-00161 Roma, Italy. E-mail: cordeschi@caspur.it

University with McCarthy. In England, Alan Turing's legacy was taken up by Donald Michie at Edinburgh, before AI research spread to other European countries and around the world.

Differently motivated analyses of the origins and developments of AI have been suggested (see McCorduck 1979; Crevier 1993; Cordeschi 2002; Mirowski 2002). In the present paper some less well-known contributions and events that precede the Dartmouth Conference are investigated, with the aim of showing how earlier attempts in mechanizing intelligence raised several questions which were to become controversial and much debated issues in AI research over the following years.

MACHINE INTELLIGENCE

Developments in computability theory dating back to Alan Turing in the 1930s and in early computer science in the 1940s are well known and investigated (see, e.g., the syntheses by Hodges [1983], and Davies [2000]). Given this background, I shall deal with the following in this section: which features of the computer made people think of it as an *intelligent* machine, even though it is so different from the structure of the human brain? I think these features were effectively summarized by Claude Shannon, one of the four proponents of the Dartmouth Conference, in his 1950 seminal article on chess programming.

> Machines of this general type [i.e. general-purpose digital computers] are an extension over the ordinary use of numerical computers in several ways. First, the entities dealt with are not primarily numbers, but rather chess positions, circuits, mathematical expressions, words, etc. Second, the proper procedure involves general principles, something of the nature of judgement, and considerable trial and error, rather than a strict, unalterable computing process. Finally, the solutions of these problems are not merely right or wrong but have a continuous range of "quality" from the best down to the worst. We might be satisfied with a machine that designed good filters even though they were not always the best possible (Shannon 1950: 256).

The first feature mentioned by Shannon transforms the computer from a calculator (capable of performing standard arithmetic functions) into a symbol manipulator or general-purpose machine (i.e., a machine performing a large variety of tasks, not simply numerical ones). The second feature endows the computer with the ability to make a decision or a choice from among different alternatives, using a trial-and-error procedure based on previously obtained results. The third feature concerns the ability of the computer to deal with complex problem domains, in which the combinatorial explosion of the alternative solutions to a problem (think of the game of chess) make it necessary to devise procedures, or strategies, that give place to a "good filter," not necessarily an optimum one, in the selection of different alternatives. These abilities of computers provided a backdrop to the Dartmouth preparatory document, and would always be judged to be important in the next decade.

The last two features Shannon alluded to (trial and error and satisfactory-filter procedures) are made possible in computer programming by a particular kind of instruction, conditional branching, which states that, if a given condition has been satisfied, then a certain sequence of instructions must be performed; otherwise, another sequence must be performed. To quote Shannon again:

> The computer operates under the control of a "program". The program consists in a sequence of elementary "orders". ... [A] type of order involves a decision, for example: C291, 118, 345. This tells the machine to compare the contents of box 291 and 118 [in memory]. If the first is larger, the machine goes on to the next order in the program. If not, it takes its next order from box 345. This type of order enables the machine to choose from alternative procedures, depending on the results of previous calculations (Shannon 1950: 264).

A conditional branching instruction (or "order") is such commonplace in programming techniques that it may appear dull to insist on it, as well as on the aforementioned computer features.[1] But if we followed a suggestion of Hofstadter (1979), and were to go back to the time when this was actually first observed in computers, we might realize why there was the feeling that the notion of "program" was more general and powerful than one could suppose.

The EDSAC machine, fully realizing those features of computers, had just been built by Maurice Wilkes's group at the Cambridge Mathematical Laboratory, when Anthony Oettinger, the pioneer of machine translation at that time in Cambridge, wrote two of the early "intelligent" programs, both running on EDSAC. These programs were able to modify their performance on the basis of obtained results, i.e., to exhibit a simple learning ability (Oettinger 1952).

One of these programs, the Shopping Program, simulated the behavior of a child sent on a shopping tour. The task was to learn where to buy certain articles in a simulated world of different shops. The program hunted for the article requested by going from shop to shop in a random fashion, until it came to the desired one, and storing in its memory the shop's location. When the same article was required, the program went directly to the right shop, without further searching. Moreover, the program was endowed with a certain "curiosity," as Oettinger put it, in its random search, it stored in its memory other non-requested articles, so that when it had a specific request for one of these articles, it was able to go directly to the right shop without further searching. Needless to say, the simple learning ability of this program (a kind of rote learning) is crucially based on conditional branching instructions.

Oettinger's approach to machine intelligence was influenced by three articles that he mentions, published in the immediately preceding years. The first was the aforementioned article by Shannon, the other two were written by Turing and Wilkes, and concerned "mechanical thought," that is, the alleged intelligence of the new digital machines.

Turing's article, "Computing Machinery and Intelligence," went on to become one of the best-known and most frequently mentioned texts in AI literature, both for its profound insights which anticipated future developments in computing machines and for what Turing called the "imitation game" (Turing 1950). There were three participants in the game: a man, a woman, and an interrogator. The latter, by asking the most varied questions and receiving the answers in a standard form on two different terminals, had to guess which was the man and which the woman. Turing imagined that, in giving his answers, the man would try to fool the interrogator, while the woman would try to help him. He proposed, therefore, to substitute a machine for the man, in fact, a general-purpose digital computer, and to see how it would manage in the game, i.e., to what extent it would manage to fool the interrogator. Would the latter, Turing wondered, be mistaken in identifying his fellow players "as often" as when a man, and not a machine, played the game? He intended this question, posed as it is "in relatively unambiguous words," to replace the more popular but misleading one, "can machines think?" (p. 433).

Wilkes, referring to the imitation game in his article "Can machines think?," maintained that, to believe seriously that one could "simulate human behavior" using a computer, it would be necessary to design a "generalized 'learning' program," i.e., a program able to learn *any* subject chosen by the programmer—a very distant goal, given the performance of the programs that had been devised so far (Wilkes 1951).

Oettinger held that his programs provided at least partial responses to the requirements set by Turing and Wilkes. Far as they were from manifesting the "generalized" ability to learn indicated by Wilkes, these programs still managed to improve their performance in certain specified and well-defined tasks. They would have been able, therefore, to pass at least "restricted versions," as he put it, of the imitation game. Oettinger thus seems to be the first to interpret the imitation game as a *sufficiency test* (a "criterion," he called it) in evaluating the performance of individual programs in well-defined domains. Notice that it is in this "restricted" version, not in the "generalized" one put forward by Wilkes (and probably closer to

Turing's insights), that the imitation game, known as the *Turing test*, would become popular afterwards among the AI community. This gave rise to different interpretations of the validity of the test, and also to certain misunderstandings, up to the controversial Loebner Prize.[2]

Oettinger pointed out some central issues regarding the new-born computer simulation of intelligent behavior. Computers, he said, could simulate certain *functions* of the brain, not its physical *structure*. Thus "Turing's criterion" could only be used to test the *functional correspondence* between computer and brain (Oettinger 1952: 1261). There are many ways of physically realizing certain brain functions. A program able to learn by conditioning could be "synthesized," as he put it, as a physical structure or a "special machine" different from EDSAC (pp. 1261–62). However a "universal digital computer" like EDSAC has the interesting feature of being able, "when provided with a suitable *program*, to mimic arbitrary machines in a very general class" (p. 1243). It is precisely with this feature of computers, clearly stated here, that one could hope to grasp *universality* as a distinctive feature of the human mind.

Oettinger observed how the non-numerical (i.e., symbolic) nature of computers should appeal to those who, "like psychologists and neurophysiologists, are interested in [their] potentialities . . . as models of the structure and functions of animal nervous systems" (p. 1244), thus explicitly introducing the issue of computers modeling organism behavior—a crucial issue, as we shall see in the next section. Further, his interpretation of the conditional branching instruction would have been particularly engaging for many of them. As seen previously, Shannon described conditional branching as a procedure giving the machine the ability of *deciding* or *choosing* between different alternatives, based on previously obtained results. Oettinger emphasized that this procedure was crucial for his own programs, because it allowed them "to organize new information meaningfully and to select alternative modes of behavior on the basis of this organisation" (p. 1247).

As can easily be seen, a branching instruction or "order" is merely EDSAC's ability to simulate the behavior of an analogue feedback-controlled device. But it was Rosenblueth et al. (1943) who stressed the *discriminative* abilities of these devices, which justified the psychological language of "choice," "purpose," and so forth. As Marvin Minsky observed, cybernetics provided "a sufficiently concrete (i.e., technical) foundation for the use of mentalistic language as a constructive and powerful tool for describing machines" (Minsky 1968: 2). And both the utility and the legitimacy of using mental language to describe the behavior of machines is an issue later debated in the philosophy of AI (see Cordeschi 2002).

An invitation to use with caution psychological terms suggested by conditional branching, such as "decision" or "discrimination," let alone "thought," came from a further article by Wilkes. On the one hand, Wilkes recognized the importance of conditional branching instructions in the design of learning programs, like those Oettinger had just implemented on EDSAC, since "they give the machine a power of discrimination." On the other hand, he pointed out that the use of those psychological terms might be simply metaphorical, and that "the use of the word *think* in connection with [branching instructions] would be justifiable only if the use of a convenient technical term were thereby secured" (Wilkes 1953: 1232).

To conclude, it would seem that Oettinger, in reviewing the features of his (admittedly simple) programs, raised some of the issues then far debated among philosophers, psychologists, and AI researchers. His very definition of machine intelligence in connection with a program "capable of performing functions which, in living organisms, are considered to be the result of intelligent behavior" (Oettinger 1952: 1251) seems to antedate the definition of AI given by the authors of the Dartmouth preparatory document, i.e., to make "a machine behave in ways that would be called intelligent if a human were so behaving" (McCarthy et al. 1955).[3]

STEPS TOWARDS DARTMOUTH, I: SIMULATING INTELLIGENT FUNCTIONS ON COMPUTERS

The second of the aforementioned articles by Wilkes was published as a reprint in the *Computer Issue*, a special issue of the *Proceedings of the IRE* (Institute of Radio Engineers), published with the collaboration of the PGEC (the IRE Professional Group on Electronic Computers) in October 1953. This special issue provides excellent evidence of results in computer design and technology achieved in the 1950s. Wilkes's article was followed by one by Shannon, "Computers and Automata," a review of computer performances comparable to those of humans (Shannon 1953), and by a long series of articles describing digital computers in all their aspects, as regards both software and hardware. In several of these articles there were glimpses of the advantages stemming from the imminent spread of transistors, which, by replacing the cumbersome and unreliable vacuum tubes, would characterize second-generation computers.

The building and dissemination of computers in the United States and Europe was strongly sponsored by government and industry. In the United States, IBM had already supported Howard Aiken's projects in the 1940s. Starting from the 1950s, almost at the same time as Ferranti was completing the Mark 1 computer in England, IBM began producing the type 701 computer, which was carefully described in the *Computer Issue*. This was the first in a series of electronic general-purpose, stored-program computers which would be used for both theoretical research aims and government and industrial applications. As a researcher at IBM, Nathaniel Rochester, then one of the proponents of the Dartmouth Conference, was responsible for the logical organization of the type 701, and wrote the first assembly program for it. In 1952, the first checkers program by Arthur Samuel, the author of the opening article for the *Computer Issue*, was run on this computer.

This and other programs were illustrated by Shannon in his article in the *Computer Issue*, including Oettinger's programs and the checkers program by Christopher Strachey, who had published a report in 1952. Other programs were able to play games fairly well: the program by D.W. Davies for tic-tac-toe, which ran on a DEUCE computer, and that for nim, running on the NIMROD electronic computer, built by Ferranti. In 1954, Samuel completed the implementation of the first learning checkers program on an IBM 704 computer, later on acknowledged as a milestone in machine learning research. Newell and Simon were designing computer chess strategies, then turning to logic theorem proving: their hand simulation of Logic Theorist was completed in December 1955 (its first proof was printed by a Johnniac computer in August 1956). Early computer simulations of perceptual tasks had been developed by Oliver Selfridge and Alfred Uttley. Computer simulation of neural nets, stemming from the seminal work by McCulloch and Pitts (1943), were in progress, in particular by Farley and Clark (1954), and by Rochester and some co-workers (including John Holland), regarding Donald Hebb's theory of learning and concept formation. In turn, both Minsky and McCarthy were dealing with several issues concerning machine intelligence.[4]

The latter experiments are alluded to or mentioned in the Dartmouth preparatory document of August 1955. But another important event took place around that time: the symposium on "The Design of Machines to Simulate the Behavior of the Human Brain," sponsored by the PGEC at the IRE National Convention held from March 21–24, 1955. The panel members were McCulloch, Oettinger, now at Harvard, Rochester, and Otto Schmitt, a biologist and an eclectic figure of science. John Mauchly, Marvin Minsky, Walter Pitts, and Morris Rubinoff were among the invited discussants.[5] The transcripts of this less-known symposium are enlightening. They are a unique inventory of the main issues involved in the building of intelligent machines, of methodological approaches, ambitions, and difficulties that would move to the forefront during the following decade, and in some cases even in more recent times.

Among the issues dealt with at the symposium, I could mention analog vs. digital computation, creativity in computers (based on Gödel's undecidability results), pattern recognition (Selfridge "unfortunately wasn't unable to be here," said Rochester, and described his early experiments), and distributed memory (in a brief discussion with McCulloch, Minsky claimed his scepticism about models with distributed memory, and rejected, as suggested by McCulloch, that a good example of such models would be a machine equipped with simple self-organizing abilities such as William Ross Ashby's homeostat). Leaving aside these albeit interesting topics, I would like to focus in more detail on some other points here.

One of the main issues dealt with at the symposium was the possibility of using computers for different aims, and what might be the "the neurophysiologists' contribution" to the building of machines reproducing brain functions. In his talk "Contrasts and Similarities" (McCulloch et al. 1956: 242–242), Oettinger distinguished two approaches in simulating human brain functions by computers, which, "although related, are far from being identical." The aim of the first, more engineering-based approach is the building of efficient machines per se, as aids in human intellectual tasks; the aim of the second, a more theoretically oriented approach, is the understanding of the human brain and behavior. Here is probably the first clearly formulated statement of a distinction between two approaches in machine intelligence, which was to become canonical in the AI community.

In the former, the more engineering-based case, the aim of simulation is to build computers that effectively duplicate or amplify human mental abilities. One might ask to what degree knowledge of the brain could be useful to the machine designer in this case. Oettinger's claim was that this issue is a controversial one. The designer might try to solve many computing and control problems using abilities in which the computer excels, e.g., speed and accuracy of computation, eventually trying to join these abilities with those in which human brain excels, e.g., degree of freedom, adaptability to new situations, and so forth. But in any case, simulation deals with brain functions, *not* with brain structure.

More clearly than in his 1952 article, Oettinger pointed out here that most successful simulations of living functions had usually been achieved not by "following the example of nature," but by using structures and means not used by living organisms, thus attaining also superior performances of living functions: "for example, while the flight of birds undoubtedly stimulated man's urge to fly, human flight was achieved by significantly different means" (this is an example, by the way, which would become popular in the AI community afterwards). As for digital computers, on the one hand, their structural features are different from those of the human brain (Oettinger mentioned here John von Neumann's estimates regarding the reliability of the components of brain and computer), on the other hand, computers successfully perform arithmetic operations using processes different from those of humans, and it can be expected that "many machines of the future will continue to have only a functional resemblance to living organisms."

In the second case, the more theoretical one, the aim of simulation is quite different in Oettinger's view: Computers are tools for testing hypotheses

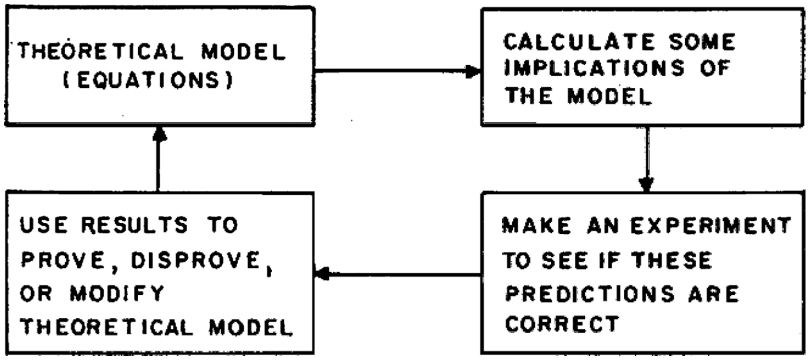

FIGURE 1 The methodological cycle proposed by Rochester in 1955, where the computer is used to test Hebb's cell assembly theory (McCulloch et al. 1956).

regarding brain functions, i.e., they can be used as neurological and psychological *models*, as already stressed by him in his 1952 article. For Oettinger, two distinct cases are possible here. First, one has a theory of brain functions stated *in mathematical form*, such as Bush and Mosteller's theory of conditioned learning (Bush and Mosteller 1955). In this case, the computer can be used as in ordinary engineering applications, to solve differential equations, to obtain numerical values of functions, and so forth. Second, one has a theory stated so to speak *in verbal form*, as Hebb's theory of learning and concept formation. Hebb (1949) introduced the notion of "cell assemblies," or nets of neurons strongly connected through excitatory synapses. As a result of repeated coactivation of constituent neurons, cell assemblies develop, as stated by Hebb's well known postulate.[6] In this case, Oettinger concluded, "the digital computer may be programmed to simulate the neuron network with its environment," with the aim of testing Hebb's theory, as shown by the simulation program illustrated by Rochester at the symposium, which I mentioned among the early attempts in simulating neural nets.

Rochester presented a set of simulation experiments on an IBM 701 in his talk "Simulation of Brain Action on Computers" (McCulloch et al. 1956: 242–244). To put it briefly, a first simulation of cell assembly theory seemed to show that Hebb's postulate was not sufficient: co-activated neurons did not spontaneously develop cell assemblies. Further simulation experiments were carried out, based on a modification of Hebb's theory proposed by "one of Hebb's students," as Rochester said at the symposium without mentioning him (it was, in fact, Peter Milner). Then a network of 63 simulated neurons, each connected to about eight others, was considered, and simulation tests of a revised version of Hebb's theory were in progress at the time with better results.[7] Rochester summarized the role of these simulation experiments.

> If the model which represents the theory works as [Hebb] says it should, the experiment gives support to the theory. Of course it could not prove that the theory correctly represents living nets. On the other hand, if the model does not work as expected it might force modifications in the theory (see McCulloch et al. 1956: 249).

This exemplifies a general computer modeling methodology, here explicitly stated for the first time in the framework of the nascent AI, which would become pervasive in brain and behavioral sciences up to our time.[8] Figure 1 shows the methodological cycle illustrated by Rochester at the symposium, which includes the process of model testing and revision through computers. It goes from formulating the model as a computer simulation of a theory of brain functions, to determining the implications of the model, to testing them, and finally to using data to prove, disprove, or modify the model, or the theory itself.

Rochester's simulation methodology was positively evaluated by AI pioneers concerned with the aim of *realistic* simulation of human behavior by computers, such as Newell and Simon. They believed that a brain or behavioral theory stated as a computer simulation model was, in general, the best alternative both to verbal (qualitative) descriptions of the theory, such as that originally given by Hebb (1949), and to mathematical (quantitative) statements, such as that by Bush and Mosteller (1955), mentioned also by Oettinger at the symposium (Newell and Simon 1963: 396–397). As to the latter, given the complexity of the human behavior, "we have a greater chance of building a theory by way of the computer program than by a direct attempt at mathematical formulation" (Simon and Newell 1956: 81). As to verbal descriptions of the theory, substituting a verbal description with a computer program, or model, could bring to light inconsistencies and lacunae of the theory, as demonstrated by Rochester. According to Newell and Simon, to formulate a theory of human behavior in terms of a computer program is to state an information processing theory (IPT).

At the symposium, Rochester pointed out how Hebb made the attempt, in his theory, to bridge the gap between psychology and neurophysiology. For Rochester, computer simulation could contribute to this aim. In his view, the relationship between the two disciplines is summarized as shown in Figure 2, in which several intermediate levels are considered between them. What is called Theorem A in Figure 2 is an assumption about the behavior of neurones, which does not contrast with established neurophysiology. This assumption, which might be identified with Hebb's postulate, is alleged to imply Theorem B, which might be seen as relating to a less microscopic level, that of concept formation, based on the development of cell assemblies. In turn, Theorem B is alleged to imply something regarding behavior, which here is called Theorem C, and can be tested against

Established Psychology
↑
Theorem C
↑
Theorem B
↑
Theorem A
↑
Established Neurophysiology

FIGURE 2 Different levels in Hebb's theory according to Rochester (McCulloch et al. 1956).

what Rochester call "Established Psychology." Although quite vaguely formulated,[9] this multilevel image of the relationships between psychology and neurophysiology makes it clear that computer simulation was intended to test the "transition from Theorem A to Theorem B," as Rochester put it, thus reducing the gap between psychology and neurophysiology in Hebb's theory. As seen previously, it is this very transition that initially failed to be proved by simulation, so making it necessary to modify Hebb's original postulate.

These brief statements by Rochester about neurophysiology implicitly touched on the issues of levels of explanation and the role of neurophysiology—this time not as a contribution to computer design, but as a level of explanation of behavior in its relationship to computer science, an issue far debated in the philosophy of AI. According to Newell and Simon, for example, IPT level could be seen as *a less microscopic* level than both Theorem A and Theorem B levels. The theoretical hypotheses and constructs involved in the latter, e.g., Hebb's cell assemblies, are neurological—albeit "neurological in a broad sense,"—as admitted by Newell and Simon (1963: 396). IPT hypotheses and constructs, e.g., elementary information processes, are at a more abstract level, actually an *intermediate* level between neurophysiological microlevel and behavioral macrolevel (for Rochester, presumably the "Established Psychology" level). This more abstract level is a *new level*, and concerns AI as a science of the mind: it is the level of building and testing models of human behavior through the newborn heuristic programming. It is at this level that simulation provides a *functional* test of the theory.

Notice that the relationship between neural net simulation and IPT simulation was stated by Simon during the early stages of heuristic programming. In the first place, Simon distinguished two uses of computer simulations: on the one hand, to build machines "imitating the human processes only when this proves the most efficient way to do the job" (this is "the goal of 'artificial intelligence,'" as he put it), and on the other hand, to understand "the human mind by imitating it."[10] In the second place, Simon observed that the latter use of computer simulation could be classified according to its "closeness . . . to, or its remoteness from, underlying physiological processes" (Simon 1961: 111). Thus, "the goal . . . in simulating complex human behavior [through IPTs] is the same as the goal in simulating neural nets: we wish to explain behavior" (p. 113). But the first kind of simulation, that of IPTs, is at a more abstract level than the latter (he mentions here, among others, Farley and Clark, Rochester, and Frank Rosenblatt's Perceptron): IPTs say very little about the underlying neurophysiological processes that occur in the central and peripheral nervous systems, although the hope is that it would be possible to explain the latter by reducing them to the former at their more fundamental level (see Cordeschi 2007 for further details).

To conclude, as effectively observed by Pitts at the 1955 Western Joint Computer Conference, in the field of computer simulation there were people like Farley, Clark, Selfridge, and Dinneen who were "imitating the nervous system," and people like Newell who preferred "to imitate the hierarchy of final causes traditionally called the mind."[11] Contrary to his conclusion ("it will come to the same thing in the end, no doubt"), the gap between the two trends in computer simulation was to become wider.

STEPS TOWARDS DARTMOUTH, II: EMBODYING FLEXIBILITY IN COMPUTERS

At the 1955 symposium, questions raised by Otto Schmitt in his talk "The Brain as a Different Computer" (McCulloch et al. 1956: 244–246) were much debated. As a biologist, he stated contrasts between the ordinary digital computer and the biological brain from a point of view different from Oettinger's. For Schmitt, computers should imitate the flexibility of reasoning usually shown by humans in order to be good simulators of brain functions. Thus, computers would have to use a kind of loose or "grey logic," as he put it, not the rigid, bivalent, or "black-and-white logic" that presently characterizes them. This would allow computers to grasp ill-defined and abstract concepts, as well as to exploit the incomplete, conflicting, or partially inappropriate knowledge commonly available to humans in real life, e.g., in problem solving or decision-making situations.

These rather vague statements took on a slightly more specific form in the discussion that followed Schmitt's talk. The issue could be so stated: how can common-sense knowledge be embodied into computer programs, as regards both complex and real-time human decision making? As to complex decision making, Schmitt judged, in replying to Oettinger, that programmers should seriously consider how to embody in programs those flexibility-based features of the brain, not only when the aim is "a realistic simulation of brain behavior" (as Oettinger put it in the discussion) but also when the aim is building efficient machines, as aids in human intellectual tasks, i.e., "as tool[s] to do something for [us]" (again Oettinger). Even in the latter case, Schmitt concluded,

> It is necessary to abandon the idea of perfectly correct, uniformly logical solutions in any machine which is to arrive at generally appropriate quick solutions to complex problems when provided only with sketchy, conflicting, and partially inappropriate information and instructions (see McCulloch et al. 1956: 247).

This is also true as to situations regarding quick, real-time decisions, as in Schmitt's example of a driver who might have to decide on exceeding established speed limits, given a particular road situation—this and analogous examples are presently proposed as instances of *situated actions* in AI. This decision is easy to make for a human, but it would be most difficult for a rigidly programmed computer. Thus the programmer should give the machine "a great deal of tradition and factual information, and some personal opinion," and an ability to revise its conclusions, a move not allowed in classic logic-based reasoning.

> The strength of the human brain in this [real time] situation lies in the human's ability to make a decision promptly and forcibly on the basis of inadequate evidence, to carry through on the basis of these decisions as though they were axiomatic—unless forced to modify them—and to be successful at doing all this (*ibid.*).

In Table 1, I sum up certain contrasts between ordinary computers and brains according to Schmitt. For him, the former should be equipped with information and procedures that would imitate those of the latter, in order to be effectively used both in the *realistic* simulation of the human behavior and in *efficient* complex problem solving by machines.

Based on these contrasts, Schmitt wondered: "Suppose that you wanted . . . to insert the knowledge of what 'democracy' is into a computer:

TABLE 1 Computers (left) and Brains (right): Contrasts in Schmitt's View

Determinacy	vs.	Indeterminacy
"Purely digital" computations	vs.	"Statistical," "distributed" computations
"Black and white" logic	vs.	"Grey" logic
Logical reasoning	vs.	Common-sense (knowledge-based) reasoning
"Systematic," fully-informed procedures	vs.	"Non systematic," partially informed procedures

would it not be impossible? It would certainly be very difficult...." Oettinger was more optimistic.

> With computers it seems to me that we are able in principle, by the use of appropriate programming or designing of structure, to build in one swoop the whole background of explicit existing knowledge (see McCulloch et al. 1956: 249).

This seems a prelude to the future debate on how to embody abstract concepts in computer programs, and on the very possibility of grasping the background of explicit knowledge by them: an issue regarding what will be called the knowledge representation problem in AI. How to get a computer with common sense has been at the core of McCarthy's and Minsky's research, albeit from different points of view, since the very beginning of AI.

Some of the contrasts in Table 1 were stated quite vaguely and even improperly. Consider, for example, the contrast between a "systematic" (i.e., computer programmed) and "non-systematic" (i.e., sketchily informed) search for complex-problem solution. Non-systematicity seems to include some kind of random elements of a not clearly specified nature—an issue touched on in the Dartmouth document, which, as with the transcript of the symposium, includes a brief discussion of some search procedures including randomness, such as the Monte Carlo method (McCarthy et al. 1955). But randomness apart, as Schmitt put it,

> ...with feedback checks of results, you can, in general outguess the systematic machine and can do it at lesser cost. Probably, it is by this kind of process that we get answer to difficult problems: we go to a point known to be in the vicinity of a solution, and then feel about and lash out blindly until we chance upon a solution (see McCulloch et al. 1956: 248).

One should notice that computer programs began to be capable of such *non-systematic* search procedures. Schmitt's vaguely defined non-systematic machine, able to get answers to problems "with feedback checks of results," is precisely the machine equipped with "considerable trial and error, rather than a strict, unalterable computing process," described by Shannon in his aforementioned 1950 paper. At the time, Newell and Simon, with Clifford Shaw, were experimenting on such a procedure in Logic Theorist, as a particular problem-solving procedure (actually, a heuristic one) "obtaining a feedback of the results [of a choice] that can be used to guide the next step" towards the solution (Newell et al. 1957: 121). It was a similar procedure that Oettinger had experimented with in his learning programs, albeit in a much simpler form. Schmitt seems to share here the idea that computers were machines following logically unalterable procedures, and thus not capable of modifying their behavior under differing circumstances.

> I believe . . . —he claimed—that we are taking an unbalanced view of the problem, based on the phenomenal success of the large digital machines, and are thereby depriving ourselves of a tremendous complementary development of more brain-like machines (see McCulloch et al. 1956: 245).

Notice that similar claims of "more brain-like machines" were made by those whose aim was to oppose the self-organizational ability of earlier neural nets, such as Rochester's, to the alleged inflexible performance of computer programs. Schmitt's criticism, however, is directed not solely towards simulation programs, imitating "the hierarchy of final causes traditionally called the mind," but also towards neural net simulation "imitating the nervous system," to use Pitt's words again. Thus what Schmitt called "more brain-like machines" were not neural net systems, as opposed to computer programs. As a biologist and a biophysicist, Schmitt was well aware of earlier work in self-organizing and neural net systems, to begin with Nicolas Rashevsky, Asbhy, McCulloch, and Rochester himself (to mention those he mentions). But he seems to judge those systems as sharing with ordinary digital computers the same bivalent logic, based on "black-and-white" units. Thus, for Schmitt "more brain-like machines" seem primarily to be machines endowed with a non bivalent logic, a "grey logic" indeed. Nothing enables us to speculate that Schmitt, when speaking of such a grey logic, was thinking to a kind of *fuzzy logic* as we intend it presently. Rather, he seems to allude to a notion of biological computation, characterized by analog and statistical information transmission, as can be seen in his own description of the real nervous system.

Contrary to Schmitt's conclusions, on the one hand, later self-organizing system research would try to grasp certain nervous system computational features he pointed out, and on the other hand, the "sketchy, conflicting, and partially inappropriate information and instructions" he alluded to as characterizing human problem solving would become the core of computer programming in the field of complex decision making. Let us see the latter point in further detail.

Another symposium is a case in point here, the one sponsored by the PGEC at the March 1956 IRE National Convention, thus a few months before the Dartmouth Conference. It was one of the first meetings devoted to "The impact of computers on science and society" (Astin et al. 1956), and most speakers came not only from the academic world but primarily from government and industry. The impact of computers on science concerned engineering, physics, chemistry and biology, as well as human sciences. The impact on society concerned different computer applications in data processing, mainly in industry and government, e.g., in management, defense, and welfare. It would seem that the discussions on the ability of new machines to simulate human decision processes are here converging on an applied research field: how "to seek more effective techniques and devices to assist us in managing and arriving at best solutions to our complicated and varied problems" (p. 143).

At the symposium, speakers agreed on the current limits of computers as to this goal, but also on the fact that computer capabilities were either underestimated or not fully appreciated, so that the computer was "a new tool with great and still unrealized potential."[12] A common claim was that, up to the time, computers had been firstly considered as large calculating machines, useful in business applications (i.e., in "computations concerning money"), but less in those areas, regarding government and industry, in which complex data processing and optimisation procedures are involved. The following tasks were at the center of the various talks

- information classification and retrieval
- optimization in complex decision making
- planning

At the time, the prevalent techniques in assisting humans in such tasks were borrowed from operation research (OR, in the sequel). Computers played an important role in this field starting from World War II at least, and OR was explicitly mentioned at the symposium, as well as its difficulty in dealing with data processing and complex activities involving information processing and planning.

In the symposium the interaction between OR and AI can be vividly seen at its germinal stage. It is not by chance that the newborn expression "artificial intelligence" is used here perhaps for the first time publicly before the Dartmouth Conference. It was used by John Mauchly—one of the builders of ENIAC along with Prosper Eckert—in his talk at the symposium, in dealing with the issue raised by David Sayre, at the time at IBM and one of the authors of FORTRAN with John Backus. The issue concerned decision-making procedures in complex problem solving and planning (scheduling of production, control of traffic in airline systems, and so forth).

As is well known, the expression "artificial intelligence" was introduced by John McCarthy in the 1955 document proposing the Dartmouth Conference. Sayre's name was present on the list, attached to the document, of the people whom the organizers of the conference believed might be potential participants, as interested in the AI research program. At the symposium, Sayre touched on the issue of machine intelligence, speculating about a way to endow a machine with what he called "something that approaches intelligence."

> I would envisage placing a machine in an environment which it can affect by its actions and from which the consequences of actions are feed back to it. It would begin with a procedure that had been given to it, but it would at the same time execute neighboring procedures and test out by its interaction with its environment whether one of the neighboring procedures might be more successful than the one it had executed; after enough favorable evidence it would adopt this procedure (see Astin et al. 1956: 157).

This kind of machine might have satisfied certain of the criteria questioning the flexibility requirements of computers, and appears to be endowed with that "self-improvement" ability which characterizes the "truly intelligent" machine alluded to in the Dartmouth document (McCarthy et al. 1955). Sayre, however, explicitly related such an intelligent machine to decision making in OR, when he suggested that complex activities or tasks, such as the aforementioned problem solving and planning, required "a rather different technique of machine use than we have yet developed." Given that no "exact procedure" had been evolved for solving these problems, the issue at point was "how to cause a machine, which has been given a fairly exact procedure, itself to amplify and correct it, constantly producing better and better procedures" (Astin et al. 1956: 149). These "fairly exact" or "inexact" procedures were underlined by Mauchly as a mark of machine intelligence.

> It is certainly true that many of us are interested in what has been given the name "artificial intelligence." This is indeed a field in which a great deal is going to be done, and there will be much influence on the future applications if we are successful in some of the endeavors which [Sayre] described as coming under "inexact" rules, procedures, and applications (see Astin et al., 1956: 155).

The point at issue here is the ability of computers to make decisions, simulating human problem-solving procedures, in order to assist humans in complex information processing, planning, and decision making. At the time, it was Simon who would have pointed out the limits of OR techniques in dealing with such complex situations, where information is sketchy, and procedures do not guarantee optimization in decision making. Developments of new, AI-based programming techniques were promptly applied in the field of management and decision making, where economics and psychology seemed to converge. As Simon viewed it:

TABLE 2 Simon's Oppositions in Decision Making

economic man	vs.	administrative man
omniscience	vs.	partial information
ideal rationality	vs.	bounded rationality
maximising	vs.	satisficing
mostly well-structured problems	vs.	mostly ill-structured (real-life) problems
linear programming (OP)	vs.	heuristic programming (AI)

> AI was born in the basement of the Graduate School of Industrial Administration at Carnegie Mellon University, and for the first five years after his birth, applications to business decision making (that is OR applications) alternated with applications to cognitive psychology (Simon 1997: 5).

This is a personal view of the origins of AI, but it effectively points out the role of early AI in evolving new techniques in data processing. Briefly, for Simon, the model of the decision maker is not the omniscient economic man, the *Homo oeconomicus* of classic economics, who maximizes his choice as predicted by the game theory. Endowed as he is with an ideal rationality, economic man is assumed to be fully informed on the problem domain or environment, as complex as this may be. In fact, this model is an extreme idealization too far removed from the actual decision maker who is commonly dealing with complex, usually ill-structured problem domains, about which he is poorly informed. Another, more realistic model was proposed by Simon, that of the "administrative man." This deals with both computationally complex and real-life problems, and endowed as he is with a kind of "bounded rationality," as Simon put it, is usually unable to maximize his choice, so using "satisficing" decision procedures, finally called *heuristics.* To put it a bit crudely: disciplines initially involved in using these two different models of decision maker were, on the one hand, OR, based on common linear programming and probability theory techniques, and, on the other hand, AI, based on the new born heuristic programming. I sum up Simon's oppositions in different views of decision making in Table 2. Briefly, in Simon's view it was OR's failure in dealing with more human-like problem solving procedures that was the major cause of its early divorce from early AI (for details, see Cordeschi 2007).

To conclude, both Schmitt's "sketchy, conflicting, and partially inappropriate information and instructions" and Mauchly's "'inexact' rules, procedures, and applications" seem to state requirements then met by early-AI heuristic, often human-like, rules or procedures. Regarding the previously mentioned areas concerned with management and complex decision making and planning, the background in which those requirements were initially met is the theory of the administrative man, developed by Simon starting from the 1940s.

CONCLUSION

On the thresholds of the Dartmouth Conference, and against the backdrop of the spread of early large digital computers, several issues were raised that would influence both future research areas and future controversies in AI. In summary, I would mention the following:

- how to use non-numerical, i.e., symbolic, programming in the simulation of human abilities by machines
- how to state different uses of computers: on the one hand, in realistic simulations of organism behavior and, on the other, in efficient engineering and management applications
- how to state the theory-model relationship within non-numerical computer simulation, given empirical facts and theoretical hypotheses regarding the brain or behavior
- how to justify the role of neurophysiology, having identified different levels of investigations—behavior processes and brain processes—both considered, however, as functional levels
- how to embody knowledge in computers, and what kind of logic would be useful, first regarding real-life situations
- how to relate decision making and OR with newborn AI techniques—apparently more capable of dealing with complex and ill-structured problem domains

Heuristic programming has been the case in point here. Logic Theorist has been considered the first heuristic program. Although it played an important role at the Dartmouth Conference, programs different in complexity, from Oettinger's to Samuel's above all, included procedures that could be called heuristics. Advances in earlier heuristic programming, also seen as a promising approach in data management and complex decision making, are among the most relevant causes that made program simulation of human behavior prevail over distributed, self-organizing, and neural net approaches. These began to be rapidly and diffusely seen as a more brain-like style of computation, in particular when AI—as a new science of the mind—was suggested to be a level of behavior explanation autonomous from the nervous system level (or levels).

Two years after Dartmouth, at the 1958 Teddington Symposium, the aforementioned opposition between imitators of the mind and imitators of the brain was definitively stated by Minsky, in his review of earlier advances in heuristic programming (see Cordeschi 2002: 187–189). Minsky (1959) opposed hierarchic systems, "dealing with rather clear-cut syntactic processes involving the manipulation of symbolic expressions" to "'network' machines," endowed with fairly simple self-organizational capabilities. He claimed his disaffection from the latter, if "really sophisticated behavior" is to be simulated. Moreover, it would not have been surprising if, once presently unknown nervous system mechanisms of intelligent activities had been identified, "the remaining heuristic theory would not be very different from the kind concerned with the formal or linguistic models." At the moment, it might be thus worthwhile, Minsky concluded, to devote major efforts to heuristic programming, or what "some of us call 'artificial intelligence'."

REFERENCES

Astin, A. V., L. Cohen, J. W. Forrester, A. W. Jacobson, J. Mauchly, R. E. Meagher, and D. Sayre. 1956. The impact of computers on science and society. *IRE Transactions on Electronic Computers* EC-5:142–158.

Bush, R. and F. Mosteller. 1955. *Stochastic Models for Learning*. New York: Wiley.

Cordeschi, R. 1996. The role of heuristics in automated theorem proving J.A. Robinson's resolution principle. *Mathware and Soft Computing* 3:281–293.

Cordeschi, R. 2002. *The Discovery of the Artificial: Behavior, Mind and Machines Before and Beyond Cybernetics*. Dordrecht: Kluwer Academic Publishers.

Cordeschi, R. 2007. Steps towards the synthetic method. Symbolic information processing and self organizing systems in early artificial intelligence modeling. In *The Mechanisation of Mind in History*, eds. M. Wheeler, P. Husbands, and O. Holland, Cambridge, MA: MIT Press.

Crevier, D. 1993. *AI: The Tumultuous History of the Search for Artificial Intelligence*. New York: Basic Books.

Davies, M. 2000. *The Universal Computer*. New York: Norton.

Farley, B. G. and W. A. Clark. 1954. Simulation of self-organizing systems by digital computer. *IRE Transactions on Information Theory* 4:76–84.

Feigenbaum, E. A. and J. Feldman, eds. 1963. *Computers and Thought*. New York: McGraw-Hill.

Hebb, D. O. 1949. *The Organization of Behavior*. New York and London: Wiley and Chapman.

Hodges, A. 1983. *Alan Turing: The Enigma*. New York: Simon and Schuster.

Hofstadter, D. R. 1979. *Gödel, Esher, Bach*. New York: Basic Books.

McCarthy, J., M. L. Minsky, N. Rochester, and C. E. Shannon. 1955. A proposal for the Dartmouth summer research project on artificial intelligence. http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html

McCorduck, P. 1979. *Machines Who Think*, 2nd ed. 2004. San Francisco: Freeman.

McCulloch, W. S. and W. Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5:115–137.

McCulloch, W. S., A. G. Oettinger, N. Rochester, and O. Schmitt. 1956. The design of machines to simulate the behavior of the human brain. *IRE Transactions on Electronic Computers* EC-5:240–255.

Minsky, M. L. 1959. Some methods of heuristic programming and artificial intelligence. In *Proceedings of the Teddington Symposium on Mechanisation of Thought Processes*, vol. 1:3–27. H.M. Stationery Office.

Minsky, M. L., ed. 1968. *Semantic Information Processing*. Cambridge, MA: MIT Press.

Minsky, M. L. 1987. *The Society of Mind*. New York: Simon and Schuster.

Mirowski, P. 2002. *Machine Dreams: Economics Becomes a Cyborg science*. Cambridge: Cambridge University Press.

Newell, A., C. Shaw, and H. A. Simon. 1957. Empirical explorations with the logic theory machine: A case study in heuristics. In *Proceeding of the Western Joint Computer Conference* 15:218–239. Reprinted in E.A. Feigenbaum and J. Feldman, eds. 1963:109–133.

Newell, A. and H. A. Simon. 1963. Computers in psychology. In *Handbook of Mathematical Psychology*, eds. R. D. Luce, R. R. Bush, and E. Galanter, 361–428. New York: Wiley.

Oettinger, A. G. 1952. Programming a digital computer to learn. *Philosophical Magazine* 43:1243–1263.

Rochester, N., J. H. Holland, L. H. Haibt, and W. L. Duda. 1956. Test on a cell assembly theory of the action of the brain using a large digital computer. *IRE Transactions on Information Theory* IT-2:80–93.

Rosenblueth, A., N. Wiener, and J. Bigelow. 1943. Behavior, purpose and teleology. *Philosophy of Science* 10:18–24.

Shannon, C. E. 1950. Programming a computer for playing chess. *Philosophical Magazine* 41:256–275.

Shannon, C. E. 1953. Computers and automata. *Proceedings of the IRE* 41:1234–1241.

Simon, H. A. 1961. Modeling human mental processes. In *Proceedings of the Western Joint Computer Conference* 19:111–120.

Simon, H. A. 1997. The future of information systems. *Annals of Operation Research* 71:3–14.

Simon, H. A. and A. Newell. 1956. Models: Their uses and limitations. In *The State of the Social Sciences*, ed. L. D. White, 66–83. Chicago: Chicago University Press.

Simon, H. A. and A. Newell. 1962. Simulation of human thinking. In *Management and the Computer of the Future*, ed. M. Greenberger, 95–114. Cambridge, MA: MIT Press.

Turing, A. M. 1950. Computing machinery and intelligence. *Mind* 59:433–460.

Webb, B. 2001. Can robots make good models of biological behaviour? *Behavioral and Brain Sciences* 24:1033–1050.

Wilkes, M. V. 1951. Can machines think? *Spectator.* N. 6424:177–178.

Wilkes, M. V. 1953. Can machines think? *Proceedings of the IRE* 41:1230–1234.

ENDNOTES

1. But notice that they had already been pointed out by Charles Babbage in reference to his "analytic machine."
2. See http://www.loebner.net/Prizef/loebner-prize.html.
3. Cf. also the definition of the goals of AI given by the editors of *Computers and Thoughts* a decade later: "To construct computer programs which exhibit what we call 'intelligent behavior' when we observe it in human beings" (Feigenbaum and Feldman 1963: 3). Also, see Minsky (1968: v): AI is "the science of making machines do things that would require intelligence if done by men."
4. It should be recalled that Minsky's original interests were aimed precisely at neural nets and self-organizing systems. In 1951, he, along with a former classmate, Dean Edmons, and thanks to financing obtained by George Miller, had built a machine consisting of a net of forty-plus artificial neurons (actually, vacuum tubes) linked randomly. The machine learned a path through a maze by means of a reinforcement rule. This machine is mentioned by Minsky under the name of SNARC (Minsky 1987: 76). Minsky himself recalls how this experience helped convince him of the difficulty in capturing the nature of intelligence using machines of the sort, essentially self-organizing random nets.
5. The original transcripts, edited by the speakers, were published later; see McCulloch et al. (1956).
6. This was to become known as the "Hebb rule" among the connectionists up to our time (but see Cordeschi 2002: 216–218).
7. Further simulation experiments on the faster IBM 704 computer dealt successfully with a very much larger net of 512 neurons (Rochester et al. 1956).
8. This machine simulation methodology has its own ancestors, see Cordeschi (2002). See also Webb (2001) for a general discussion.
9. This point does not appear further developed in the later article by Rochester et al. (1956).
10. This distinction, already stated by Oettinger at the 1955 symposium, was to become canonical in the AI community up to our days. At the time, it reflected the double nature of the newborn heuristic programming, as both an *efficient* and *human-like* programming technique (see Cordeschi 1996).
11. Quoted by Simon and Newell (1962).
12. The computers referred to were above all the ILLIAC and the SEAC computers.