Five 1951 BBC Broadcasts on Automatic Calculating Machines

Journal Item

Five 1951 BBC Broadcasts on Automatic Calculating Machines

Allan Jones
Open University

In May and June 1951, five leading figures of British computing—Douglas Hartree, Max Newman, Alan Turing, Frederic ("Freddie") Williams, and Maurice Wilkes—spoke about their work on BBC radio. This article examines surviving texts of their broadcasts, and the speakers' principal points are summarized through quotations and commentary. The broadcasts are placed in the context of contemporary developments in computing and the particular BBC service on which they were broadcast.

Researchers of Britain's early postwar history of computing have known for some time that a series of five British Broadcasting Corporation (BBC) radio broadcasts under the general title of "Automatic Calculating Machines" was broadcast on the BBC's Third Programme radio service in May–June 1951. In these broadcasts, five British pioneers of computing spoke about their work. In the order of their broadcasts, they were Douglas Hartree, Max Newman, Alan Turing, Frederic ("Freddie") Williams, and Maurice Wilkes. Apart from Turing's broadcast, which has been discussed by B. Jack Copeland[1] and Andrew Hodges,[2] these broadcasts have received little attention from historians of computing.

No sound recordings of the broadcasts survive, although they all were recorded on acetate phonograph discs prior to transmission. However, texts of all five broadcasts survive as BBC transcripts, which were taken from the recordings shortly after they were made. These transcripts are held at the BBC's Written Archives Centre in Caversham, near Reading, and are the basis for this article.

In addition to the existence of the five BBC transcripts, three of the speakers' scripts are known to have survived. These are Turing's, held at the Alan Turing archive at King's College, Cambridge, and those of Wilkes and Newman, copies of which are held by Wilkes. Turing's script has been published,[3] although curiously not in the *Collected Works of A.M. Turing*,[4] and is also available on the Word Wide Web.[5] None of the other scripts has been published.

All five of the speakers in this series were, or had been, involved with one or more of the three major computing projects in the UK in the immediate postwar period:

- ACE (Automatic Computing Engine), at the National Physical Laboratory, designed by Turing, launched in 1946 and experimentally operational in a pilot version in 1950, although not completed until late 1951.[6]
- EDSAC (Electronic Delay Storage Automatic Computer), at Cambridge University, designed by Wilkes, begun in 1947 and operational in May 1949.[7]
- Mark 1 Prototype at Manchester University, associated with Newman, Williams, and (from 1948) Turing. Operational from April 1949 to August 1950, having evolved from an earlier "Baby" test machine (operational June 1948) and replaced in February 1951 by the Ferranti Mark 1.[8]

Table 1 (see p. 4) gives the titles and broadcast dates of the talks, and the computers that the speakers were associated with at the time of the broadcasts.

As Table 1 shows, the Cambridge and Manchester projects were well represented in the five broadcasts. The National Physical Laboratory's ACE computer project was represented only indirectly, via Turing, who was no longer associated with it when he made his broadcast. This machine was not, in any case, fully completed at the time of these broadcasts.

The Third Programme

Before discussing the content of the broadcasts, I should mention the BBC's Third Programme, on which these five talks were transmitted. This idiosyncratic radio service—so unlike almost anything in modern-day broadcasting—occupied an important position in Britain's intellec-

1058-6180/04/$20.00

Table 1. "Automatic Calculating Machines" broadcasts.

Broadcast date	Repeat date	Speaker	Title	Computers associated with at time of broadcast
5 May 1951	24 June 1951	Douglas Hartree	"Automatic Calculating Machines"	Cambridge, EDSAC
8 May 1951	26 June 1951	Max Newman	"Automatic Calculating Machines"	Manchester Mark 1 (Ferranti)
15 May 1951	3 July 1951	Alan Turing	"Can Digital Computers Think?"	Manchester Mark 1 (Ferranti)
2 June 1951	4 July 1951	Freddie Williams	"Automatic Calculating Machines"	Manchester Mark 1 (Ferranti)
5 June 1951	10 July 1951	Maurice Wilkes	"The Use of Automatic Calculating Machines"	Cambridge, EDSAC

tual life, and an appreciation of its philosophy sheds useful light not only on the broadcasts but on the nature and size of the audience that would, or could, have heard them.

The Third Programme was a national domestic radio service inaugurated by the BBC in September 1946 with an avowedly intellectual and cultural character. Two other national domestic radio services, the Home Service and the Light Programme, already existed—hence the name Third Programme. Central to the activities of the Third Programme were broadcasts of serious music, literature, and speech. Many leading thinkers of the day were invited to speak on the service, and in general the style of presentation was for the speaker to deliver a scripted talk typically lasting around 30 minutes. (All the talks in this series lasted about 20 minutes.) Interview-style presentations were unusual, although widely used on the BBC's Home Service. The fact that the five broadcasts were made by the computing pioneers themselves, rather than by journalists or commentators, is thus typical of the approach used on the Third Programme and is what makes them particularly interesting as historical sources.

The Third Programme had no regular timetable of program "slots"—there was no regular time of the week for science broadcasts, poetry, or anything else. The only way for listeners to find out about forthcoming broadcasts was to consult the program listings published daily in newspapers or the BBC's own weekly *Radio Times*. The five computer broadcasts discussed here, therefore, did not form part of a regular science and technology stream, and were not even broadcast at equal intervals or regular times. They nevertheless were conceived of and presented as a series, and at the end of each broadcast there was an announcement of when the next would take place.

The Third Programme operated only during the evening, and listeners were not expected to spend the whole evening listening to the service. Indeed, it was considered undesirable for them to do so. Rather, listeners were expected to tune in for just the broadcasts that interested them or which aroused their curiosity, and then switch off, or listen to another station. As the BBC's historian Asa Briggs noted, "The Third Programme set out not to meet the wishes of listeners who would be engaged in continuous listening but rather to recruit 'patrons'."[9]

How many patrons the Third Programme had at the time of these broadcasts is hard to pin down. In the late 1940s, it was claimed to be between 1.5 million and 2.5 million.[9] A typical audience for any single Third Programme broadcast would naturally have been much smaller than this. In 1949, two years before these five broadcasts, the audience for a Third Programme broadcast was estimated to be around 90,000, and it appears not to have grown during the next few years.[10] Indeed, the percentage of BBC radio listeners tuning in to the Third Programme was generally 1 percent of the total radio audience during the early 1950s.[11] The more popularly oriented Home Service and the Light Programme would typically have audience figures of a few million for their more popular broadcasts. The Third Programme was subject, in any case, to technical constraints that restricted its coverage to the more populous parts of the UK. Reception in many parts of the country was poor, and in remote areas nonexistent.

Though small in absolute terms, the Third Programme's audience was nevertheless influential, as Britain's academics, artists, and intelligentsia were disproportionately represented among it. However, professional intellectuals were by no means the Third Programme's only listeners. A 1949 survey reported that 35 percent of the audience was working class, although it appears that *working class* was then defined more widely, and *middle class* more narrowly, than would now be the case.[12]

Radio broadcasts relating to computers, cybernetics, and artificial intelligence (as we would now call it) were by no means rare on the BBC in this period. Between 1946 and December 1956, there were 24 such broadcasts, not counting repeats. Most of these broadcasts date from after 1950, and most were on the Third Programme. Speakers in these other broadcasts included Norbert Wiener, Colin Cherry, Wolfe Mays, Frank H. George, and Christopher Strachey.[13] The extent of this coverage of computer-related matters is perhaps surprising given the widespread perception in the UK that press and broadcasting personnel are biased against (and ignorant of) science and technology. However, computer-related broadcasts on the Third Programme were probably a special case owing to the particular interests of their producer, to whom I will later return.

The broadcasts

Although much of the Third Programme's output was broadly educational, the Third Programme was not part of the BBC's educational service. Third Programme broadcasts were therefore not didactic in the usual sense, and speakers were encouraged to address the listener as an equal who just happened not to be conversant with the speaker's subject. Accordingly, none of the speakers in this series pitched his talk at a high technical level, and none adopted the style of a formal academic lecture where one might expect a progression of ideas from fundamentals to higher-level concepts. In this respect, the style of these broadcasts was similar to that used for other factual talks (not just science talks) on the Third Programme at that time. In general, the speakers confined themselves to fairly simple factual accounts of what computers were and what they did. As so often, however, Turing was something of an exception. His presentation, although not of a high technical level, certainly made greater demands on the listeners' comprehension.

With the exception of Williams, the speakers said relatively little about the hardware, concentrating instead on software concepts such as programs, data, subroutines, and so on, and also touched on the recurring theme of what a program in principle could and could not do. As regards the prehistory of computing, no speaker referred to wartime code-breaking activities, although Williams did mention the importance of wartime radar research for the development of computers. The names of Charles Babbage and Lady Lovelace (that is, Ada Byron, mathematician and associate of Babbage) are occasionally invoked as important pioneers, but those of John von Neumann, J. Presper Eckert Jr., and John Mauchly are not mentioned at all. Their absence was probably more out of consideration for the listener, to whom those names would have meant little, than out of chauvinism. Hartree and Wilkes, in particular, were happy to pay tribute to these American pioneers in their writings.

It cannot be claimed that the broadcasts significantly change our view of the history of computing. The transcripts of them do nevertheless offer a valuable insight into the relationship of the then emerging field of computer technology to the public understanding of that technology, as revealed through the mouths of its leading British practitioners. It is against this background that the broadcasts are most profitably viewed. Through the broadcasts we get a sense of what the speakers thought was significant in their work, what might be comprehensible to a nonspecialist audience, and where developments might lead. Significantly, we also get repeated reassurances about where the work was not likely to lead—toward the "electronic brains" so frequently invoked in popular journalism of the time. Once again, however, Turing was something of an exception.

In the space of this article, it is impossible to discuss each broadcast in depth. In the following five sections, therefore, I summarize each broadcast through quotations and commentary, taking the broadcasts in the order in which they were made. Because the transcripts were made by nonspecialist clerical staff, there are occasionally places where the transcriber has clearly misinterpreted what the speaker has said. In my quotations I have corrected such misinterpretations without comment, and in a few places I have adjusted the punctuation to something more appropriate for a written presentation. Occasional interpolations of my own are enclosed within square brackets.

Douglas Hartree

The first speaker in the series, Douglas Hartree, had broadcast about computers on the BBC five years earlier, in December 1946, on the Home Service. In his earlier broadcast, he had mainly been concerned with the ENIAC machine, which he had recently used during a visit to the US.

At the time of his 1951 broadcast, Hartree was Plummer Professor of Mathematical Physics at Cambridge University, although in the immediate prewar period he had been associated with developments in analog computing at Manchester University, particularly the differential analyzer.[14] His inaugural lecture at Cambridge had been titled "Calculating Machines: Recent and Prospective Developments," and he had already published various writings relating to digital computers, notably his account of the ENIAC machine in *Nature*[15] and in his book *Calculating Instruments and Machines*.[16]

In his May 1951 broadcast, Hartree was concerned at a basic level with the differences between computers and other sorts of machine. He described the parts of a computer, the relationship between data and information in terms of the computer's operation, and the tasks computers could be made to do (such as calculating, playing games, and other apparently human-like activities). He began by emphasizing three salient points about the machines that were to be the subject of this series of talks: that they were automatic, general purpose, and digital. Only the first two of these three points were elucidated:

> By an "automatic machine" is meant one which can carry out numerical calculations of any length without the attention of an operator, once the schedule of operations to be carried out has been supplied to the machine, in a suitable form; and by a "general-purpose machine" is meant one which can be used for a large number of different kinds of calculations, by supplying it with the appropriate schedules of operating instructions.

The third of Hartree's introductory points, the digital nature of computers, was not expanded (although Newman enlarged on it in the second broadcast).

The concept of a general-purpose machine can be traced back to Babbage's proposed analytical engine. Hartree was aware of Babbage's work and mentioned it in passing as representing the first conception of a general-purpose digital machine. He then launched into the anatomy of the modern (that is, von Neumann) machine:

> An automatic digital calculating machine consists of five main parts, an arithmetical unit, a store, a control unit, an input unit, and an output unit. The purpose of the store is to hold information, either numbers or operating instructions, for as long as they may be required, in the course of the calculation. In some of the older machines, the store consisted of two distinct parts, one for numbers and one for instructions. But in most of the more recent machines, the same store is used both for numbers and for instructions.

By "the older machines" Hartree was referring to machines such as the Automatic Sequence Controller at Harvard University, in which data was held on counters and instructions on punched paper tape that was read as the calculation proceeded, or the ENIAC, in which the "program" was assembled physically by setting switches and by patching together processing units via plugboards and cables. The more modern machines not only held instructions and data in the same memory, but made no distinction in the way they were held:

> But in most of the recent machines there is no distinction between the form used for numbers and for instructions. The distinction between words representing numbers and words representing instructions lies in the way in which they are used.

A consequence of this lack of distinction between data and instructions is the possibility of self-modifying programs (something on which more than one speaker was to comment):

> This possibility of modifying instructions as the calculation proceeds provides the means of instructing the machine to carry out much of the discrimination and selection between alternative procedures which a human computer would exercise in doing the same calculation by pencil and paper methods.

Hartree raised here the contentious issue of the analogy between humans and computers. From the announcement of the ACE project in autumn 1946 (the first of the British computing projects to be announced publicly), the press had had a tendency to refer to the new computers as "brains," or "electronic brains."[17] Hartree was anxious to correct what he viewed as a misapprehension:

> But do not jump to the conclusion that [in modifying its own program] the machine is thinking for itself. All these instructions for modifying other instructions, and for evaluating and using the criteria of any discrimination, have to be thought out and programmed in detail. The machine only carries out literally and blindly and without thinking, the instructions which the programmer has thought out for it.

Turing, later in the series, took a different view, as we shall see.

Martin Campbell-Kelly[18] has written that one of the distinctive features of computing as done at Cambridge at this time was the emphasis on building up a library of commonly used subroutines. Hartree alluded to this policy:

> … the machine and the process of providing it with instructions may be such that groups of operations for standard processes, such as the evaluation of square roots and cube roots, can be programmed once and for all. […] The main work in the preparation of the calculations for the machine may then be the programming of a master routine consisting mainly of instructions for calling in the subroutines in the proper sequence.

Hartree concluded by mentioning the possibility of computers playing games such as chess, but he would not regard this as evidence of thinking:

> … [playing games] would come very near what, in ordinary speech, we would call thinking—an aspect of those machines on which I understand Dr. Turing will be speaking. But remember, that the sequence of operations for such a process still has to be programmed, and Lady Lovelace's words still apply—"the machine can only do what we know how to order it to perform."

This remark of Lady Lovelace's is a recurring theme in the first three broadcasts, with each speaker giving a different verdict on its veracity.

Max Newman

The mathematician Max Newman had worked on breaking German Enigma–coded messages at Bletchley Park during World War II along with Turing, who had been one of his students at Cambridge University before the war. At the time of his 1951 broadcast, Newman was professor of mathematics at Manchester University, and his interest in computing was mainly with a view to their use as a mathematics research tool. Although Newman had largely initiated the project to build a computer at Manchester, he had little involvement with the design of its hardware.[19]

In his talk, Newman was concerned with what made computers so powerful. He located their power in the fact that they used a limited set of elementary operations, and any sequence of operations could be repeated until a stipulated condition was satisfied.

At the start of his talk, he picked up Hartree's idea of the general-purpose machine, which could perform a wide range of tasks despite its relatively small repertoire of elementary operations:

> It is the arrangement of these elementary operations, and the way they are interrelated, that causes us to call one series [that is, program] a way of solving equations, and another a routine for playing bridge. [….] Problems that appear not to be arithmetical at all may often be made so, by quite trivial changes in the way they are stated.

However, the existence of a set of elementary operations is not by itself what gives a computer its power:

> If an automatic computing machine really needed a tape containing 100,000 instructions in order to do 100,000 elementary operations, somebody would have to punch the tape; and that "somebody" might be just as usefully employed in doing the 100,000 elementary sums himself, with a pencil and a piece of paper.

This is perhaps debatable. Even if a task with 100,000 operations required a program tape containing 100,000 instructions, there might still be a benefit in creating the tape because the program could be used many times to process different sets of data. As far as Newman was concerned, however, the utility of a computer lay in the fact that a multistep operation can be specified in fewer steps than the operation itself would take:

> The machines that are the subject of this talk […] all have the essential property of being able to do a big job from a few instructions. … [The] arrangements by which this is achieved are the most characteristic feature of these machines, and are the source of those complexities of behaviour that give some colour to comparisons with certain mental processes, […].

Newman went on to mention the "jump" instruction as one technique for doing "a big job from few instructions," by enabling a sequence of operations to be repeated:

> The normal procedure, when the machine is started off, is for the instruction in line 1 to be carried out first; then control passes to line 2, the instruction in it is carried out, control passes on to line 3, and so on. … [There] is a special type of instruction whose function is precisely to interrupt the normal succession. For example, Instruction 100 might be "Jump back to instruction 25."

Of course, one needs to be able to exit from the loop created by jumping back to an earlier instruction:

> There must be some way of bringing the repetitions of a cycle to an end when they have gone on long enough. […] This is accomplished by

> introducing a conditional or branched type of instruction, for example: "If line 27 is empty (i.e., contains 0) step on to the next instruction in the normal way; but if it is not empty jump to instruction 12."

Such accounts of conditional jumps have a way of being couched in anthropomorphic terms, which Newman wished to counter:

> Now there is some danger here that the jargon of "obeying instructions" and "choosing alternatives" which has become the customary way of describing the behaviour of these machines, may evoke a picture of the machine "conning" [that is, reading and memorizing] the branched instruction, looking to see if line 27 is empty, and then faithfully choosing the appointed alternative. In fact the machine "obeys" its instructions in exactly the same sense that a railway train "obeys" the points [that is, switches], going to Crewe if they are set one way and to Macclesfield if the other.

Presumably, Newman intended his railway analogy to suggest that a program can no more vary its route during a calculation than can a train driver during a journey: Once a program and its data are read into a computer, the future course of the data-processing operation is as completely determined as is the route of a train. However, the analogy is potentially misleading. The course of a train is knowable in advance, but this is not necessarily true of a computing program's calculations, as Newman himself said later (quoted below).

Like Hartree, Newman saw self-modifying programs as holding an intriguing possibility for something close to what we would now call artificial intelligence:

> The machine will add lines 2 and 3, if instructed to do so, without the least regard to whether one or the other of these lines is to be used later on as an instruction. This means that we can modify not only the true numerical material, but also the instructions themselves, in the course of the computation. […] This has, with some justification, been described as the ability to learn from results.

However, whereas Hartree was clear that such self-modifying programs could only "literally and blindly" carry out the programmer's instructions, Newman was less certain:

> It is not difficult to make up programmes of moderate length leading to networks of operations so complex that even the composer [that is, programmer] cannot predict what course the calculations will take, and it is not obvious that anyone could discover a routine to obtain the results of such a programme […]. In view of these facts it seems that the dictum of Lady Lovelace, as quoted by Professor Hartree, that "the machine can only do what we know how to order it to perform," needs to be received with some reserve. However the end of my talk is not the place to enter on these fascinating but controversial topics.

Alan Turing

By the time of Turing's broadcast, roughly a year had passed since the publication of his now famous *Mind* article in which he discussed the issue of whether computers could be said to think.[20] At the outset of his broadcast, Turing made it clear where he stood:

> Digital computers have often been described as mechanical brains. Most scientists probably regard this description as a mere newspaper stunt, but some do not. One mathematician has expressed the opposite point of view to me rather forcefully in the words "It is commonly said that these machines are not brains, but you and I know that they are." […] I shall give most attention to the view which I hold myself, that it is not altogether unreasonable to describe digital computers as brains.

Much of the rest of the talk is a summary of Turing's justification for regarding computers potentially as brains, and the kinds of reason that people put forward to oppose the suggestion that computers might one day be able to think. Of these objections, the principal one is Lady Lovelace's argument that computers only do what they have been programmed to do.

Turing was careful to make clear that the computers of his day could not plausibly be called brains; his point is that digital computers had the potential for being plausibly regarded as brains. His argument, familiar from the *Mind* article, depends on the concept of the universal machine, which he had conceived in connection with his celebrated 1936 paper,[21] although he did not mention that paper here:

> A digital computer is a universal machine in the sense that it can be made to replace any machine of a certain very wide class. It will not replace a bulldozer or a steam-engine or a telescope, but it will replace any rival design of calculating machine, that is to say any machine into which one can feed data and which will later print out results.

The next step of Turing's argument depends on a view of the brain that remains controversial, although it has a long ancestry in materialist philosophy:

> If it is accepted that real brains, as found in animals, and in particular in men, are a sort of machine, it will follow that our digital computer, suitably programmed, will behave like a brain.

The success or otherwise of this emulation is to be assessed in the test that now bears Turing's name, which he explained in his *Mind* paper and which he summarized here briefly:

> I think it is probable for instance that at the end of the century it will be possible to programme a machine to answer questions in such a way that it will be extremely difficult to guess whether the answers are being given by a man or by the machine. I am imagining something like a viva-voce examination, but with the questions and answers all typewritten in order that we need not consider such irrelevant matters as the faithfulness with which the human voice can be imitated.

Allowing the computer to respond non-orally, via a typewriting machine, suggests that Turing thought the problem of programming a computer to speak convincingly was even more challenging than that of programming it to respond plausibly to questions. Turing's own work on speech sampling and encipherment, in the latter part of World War II, may have led him to view speech synthesis as a particularly intractable problem.

Given Turing's view of the potentially brainlike behavior of computers, it is perhaps no surprise that he considered they might one day be capable of originality:

> If we give the machine a programme which results in its doing something interesting which we had not anticipated, I should be inclined to say that the machine had originated something, rather than to claim that its behaviour was implicit in the programme, and therefore that the originality lies entirely with us.

Turing acknowledged that there are immense difficulties to be overcome before a computer could behave in a convincingly human way, nor did he know how one would go about programming a machine to behave in such a way:

> I will only say this, that I believe the process should bear a close relation to that of teaching.

The essential point of Turing's observations on the difficulty of programming brainlike behavior is that the programmer may not always know what the consequences of a program may be:

> Let us now reconsider Lady Lovelace's dictum. "The machine can do whatever we know how to order it to perform." The sense of the rest of the passage is such that one is tempted to say that the machine can only do what we know how to order it to perform. But I think this would not be true. Certainly the machine can only do what we do order it to perform, anything else would be a mechanical fault. But there is no need to suppose that, when we give it its orders, we know what we are doing, what the consequences of these orders are going to be.

Thus whereas Hartree accepted the Lovelace dictum and Newman felt reservations about it, Turing rejected it—or at any rate rejected what it is usually taken to imply. For Turing, the fact that a program together with its data is a deterministic system (that is, its initial state fully determines its route to its final state) does not preclude brainlike or original behavior because we cannot necessarily predict what that final state will be and how it is reached.

Turing's talk was the only one of the series for which I have been able to find a review. Writing in *The Listener*, a weekly BBC publication, Martin Armstrong wrote:[22]

> … I was moved every few minutes to hold up my schoolboy hand with a "Please, Sir … One moment, Sir …. Will you explain what you mean, Sir, by …" this that and the other. Mr Turing remarked that many people dislike the idea that a machine could be made to think. "If machines could think," they say, "where would we be?
>
> Now I, as it happens, am one of those who dislike the idea, not, however, because it frightens me, but because it seems to me to be based on a misuse of words. To say that a machine thinks is surely, by implication, to define thought as a mechanical process, …

One can sympathize with the reviewer's difficulties in following Turing's talk, which was certainly the densest of the series. To have grasped all Turing's points on a single hearing would not have been easy.

Regarding the definition of "thinking" or "thought," which the reviewer had trouble with, Turing did actually give a sort of definition in his talk, although it would have been easy to miss:

> … to programme a machine to imitate a brain, or as we might say more briefly, if less accurately, to think.

Turing's comment here explicitly, if approximately, equates imitating a human brain with thinking, which is rather different from what we find in his *Mind* article. In that article, Turing stated that he did not want to be drawn into defining "thinking," and proposed his test as a way of avoiding the ambiguities associated with the word. What the test actually tests is not stated explicitly in the *Mind* article, although commentators have usually interpreted it as an operational test of either machine-based intelligence or machine-based thought. The informality of this radio presentation, however, appears to have encouraged Turing to use the terms "thought" and "thinking" in connection with machines rather more openly than he did in more formal contexts. Here are two further examples:

> I will not attempt to say much about how this process of "programming a machine to think" is to be done.
>
> I have tried to explain what are the main rational arguments for and against the theory that machines could be made to think …

Earlier, I mentioned that Turing's broadcast has been discussed by Hodges and Copeland. Hodges's discussion is confined to a few sentences pointing out that Turing reiterated in his broadcast ideas he had already expounded elsewhere, principally in the *Mind* article. This is true, but the broadcast at least presented a concise summary of those ideas, aimed at a nonspecialist audience. Copeland's discussion is much longer, and for the most part is less concerned with the broadcast than with the interpretation of the phrase "any machine" (taken from the broadcast). Copeland devotes a few paragraphs to Turing's suggestion that the appearance of free will in a computer may be created by the inclusion of a random process in the program, but this portion of Turing's talk was relatively short.

Freddie Williams

Freddie Williams's down-to-earth talk could hardly have been in greater contrast to Turing's. Williams was an electrical engineer who, along with Tom Kilburn, had devised the highly innovative cathode-ray tube memory used in the Manchester Baby machine, the Mark I prototype, and the Ferranti Mark I (installed February 1951).[23] His talk was almost entirely concerned with the principles of computer memory, of which—leaving aside electronic memory—there were really only two practical types at the time of the broadcasts: mercury delay lines and cathode-ray tube devices.

After briefly describing the power of computers to do large-scale calculations at high speed, Williams outlined the engineering problem faced by the designer of computer memory:

> Thus the problem reduces to finding somewhere to put strings of 0's and 1's, about half a million of them altogether, and it must be somewhere where they can be got at in sets of 20 or 40 within say a thousandth of a second when they are wanted.

The Cambridge EDSAC machine and the ACE machine used batteries of mercury delay lines for their memory. These were tubes of mercury down which pressure waves were transmitted. The operation of these devices was likened by Williams to a man shouting to a distant cliff face:

> If he shouted a number and then listened, after a certain time he would hear an echo. He could then shout the number again, the only tax on his memory would be between hearing an echo and shouting again; after a further delay he would again hear an echo and shout, and so keep the number circulating between himself and the cliff. If the echo were delayed a longish time he could shout several numbers before the first one came back and so keep several numbers in circulation. Thus one man with a poor memory could store a lot of numbers in the air, just by repeating what he heard.

Williams acknowledged that this analogy is only appropriate up to a point. In a practical mercury tube, the data (in the form of ultrasonic pressure waves) are not echoed back to the transmitter along the tube. Instead, they are received by a transducer at the far end of the tube which

> … returns the signals to the near end electrically, where they are regenerated and retimed relative to the clock by an electrical circuit.

The cathode-ray tube memory, in contrast, arranges the data two-dimensionally in space:

> Imagine a man supplied with a square of dust subdivided by low walls into a lot of little squares, just like an egg box, and supplied also with a stick to scratch the dust. Now let him be read a row of 0's and 1's and let him start at the top left-hand box and progress from box to box from left to right and top to bottom, as in reading, writing

"1" when we say "1" and "0" when we say "0".

In the electrical version the man, the stick, and the square of dust are replaced by a cathode-ray tube […] The square of dust is the face of the cathode-ray tube, the dust on it is the inexhaustible supply of secondary electrons that can be knocked out of it by a high velocity primary electron beam. This beam is itself used as the stick, and it is moved about in much the same way as in television …

The electron beam not only wrote data into the memory but also read it. By scanning the screen, the beam could detect excited and unexcited regions of the screen, as in a television camera of the sort used in those days.

A snag with this sort of memory was that data in the memory tended to be corrupted by what Williams referred to as the "splashing of secondary electrons from one box to its neighbours," and therefore the image on the screen needed regular regeneration:

> We have found that over a thousand separate "boxes" can be set up on one cathode-ray tube before the "splashing" from box to box makes the fading between regenerative visits of beam too great to be tolerated.

William's cathode-ray tube memory was a formidable engineering achievement and similar types of memory were used not only in the UK (in early Ferranti machines) but also in the US, where it was used in, for instance, the Whirlwind machine at the Massachusetts Institute of Technology during its early years. Nevertheless, computer memory remained an awkward and unreliable technology:

> But neither [mercury delay-line nor cathode-ray-tube memory] really solves the whole problem, since to store 500,000 digits would require no less than 500 [cathode-ray] tubes or [mercury] delay lines. We have progressed beyond this point by using in conjunction with cathode-ray tubes secondary methods of storage which lack the property of extremely rapid access to individual numbers.

Possibly Williams was here alluding to the innovative magnetic-drum storage used at Manchester. Even this combination of cathode-ray tube memory with other forms of storage left much to be desired:

> But the ideal has not been reached—in fact one may well conclude by saying, "the research continues."

Within a year or two, research into computer memory (mainly in the US) yielded a new type that, as Wilkes put it in a September 2001 interview with me, rapidly transformed the memory from being the least reliable part of a computer into being the most reliable. This was magnetic-core memory, which used arrays of thousands of small magnetic rings threaded on current-carrying wires. However, practical applications of such memory were still in the future at the time of these broadcasts.

Maurice Wilkes

The final speaker in the series, Maurice Wilkes, was at the time of the broadcasts running what was probably the most successful of the three British computer ventures, not only from the technical point of view but also from the organizational point of view. By mid-1951 the Cambridge EDSAC, designed by Wilkes, was not just a functioning laboratory machine but a facility used by several departments of the university.[24]

Wilkes's talk concentrated on the scientific use of the EDSAC machine:

> Already the EDSAC has contributed to a number of [research projects]. Astronomy and astrophysics are represented by problems connected with the orbits of minor planets and the equilibrium of gaseous stars, geophysics by calculations concerning the propagation of wireless waves in the ionosphere, and the effect of the motion of a ship on a pendulum used for gravity survey at sea. The machine has also been used for statistical calculations arising in applied economics and for problems in X-ray crystallography. We are about to start on a problem connected with the transmission of impulses along nerve fibres.

As mentioned, a distinctive feature of Wilkes's policy at Cambridge was the early creation of a large library of subroutines:

> You may be interested to know that when a high-speed electronic calculating machine is being used, it is generally better to calculate sines and cosines afresh from a series whenever they are required rather than to put a table into the store of the machine. It has also proved possible to construct sub-routines for carrying out some of the standard processes of numerical mathematics, such as numerical integration, or the numerical solutions of differential equations. […] The library associated with the EDSAC now contains about 150 sub-routines and it is still growing.

For comparison, the subroutine library at

Manchester at the time of this broadcast was about one-third the size given here by Wilkes.[25] The EDSAC library was, in fact, highly influential and adopted directly by some later machines such as the LEO machine, used by the J. Lyons Company, and the TAC machine in Tokyo.[26]

Another of Wilkes's concerns was to streamline, and even mechanize, the business of programming:

> There are a number of other tasks connected with programming which at first sight appear to require the application of human intelligence but which can really be done according to a set of rules.

Not surprisingly, given the large amount of computing activity that had been going on at Cambridge, issues of programming errors and reliability had started to loom large:

> When drawing up a programme it is very easy to make slips of a trivial kind; for example, one may forget to make sure that the accumulator register of the machine is cleared before beginning to add up a series of numbers. You might think that these slips could all be detected by going through the programme carefully before putting it on the machine, but experience has shown that it is not quite as simple as that. Some of the slips are sure to get through and a good deal of the time taken in putting a new problem on the machine is spent in finding them.

Some commentators, or perhaps users, were worried that the computer itself might introduce errors in calculations. Wilkes said:

> … it has even been suggested that the ideal computing system would consist of two identical machines connected together in such a way that unless they produced identical numbers at each stage the calculation would stop. I am rather against this for electronic machines at their present stage of development, mainly because the machines are quite complicated enough as it is without making them any more so. The EDSAC, for example, contains 3000 valves [tubes]. The more equipment there is in a machine, the more likely it is to go wrong.

The likelihood of a machine breaking down was indeed high. In the EDSAC's early days, the interval between failures was typically a matter of minutes, although reliability steadily improved.[27]

In rounding off his talk, Wilkes was also rounding off this series of five broadcasts:

> That the future will bring important and exciting developments, I do not doubt, but it must be remembered that from the point of view of practical achievement the subject is still in its early stages; the number of electronic digital calculating machines in operation at the present time can be counted on the fingers of one hand, and no machine has yet been programmed to play a game of bridge.

Context of the broadcasts

To help take stock of this series of broadcasts, it is useful to look at the wider context in which they took place.

The broadcasts were examples of the topical interest in computers coinciding with the emergence of working computers in Britain around 1949–1950. Other manifestations of the same interest can be seen in the article "Can Machines Think?" that Wilkes wrote for the *Spectator* around this time,[28] and the articles published in Penguin's *Science News*.[29] Both of these publications would have been aimed at the same audience as the Third Programme reached—that is, well-educated nonspecialists. At a more populist level, there was the Ferranti computer displayed at the Festival of Britain (May–September 1951), which aroused a great deal of interest. Members of the public could challenge it to a game of Nim—a simple game in which opposing players take turns to remove one, two, or three counters from an initial arrangement of 13 pieces. The winner takes the last counter. In all these cases, the issue of mechanical intelligence was never far away, and frequently alluded to—as is the case with the five broadcasts discussed here. Clearly in the lay mind, or in the minds of the people who were addressing the lay mind, this issue could not be ignored.

Aside from this rather sensationalist interest, however, the broadcasts were timely in another respect. The Cambridge and Manchester machines were beyond being laboratory novelties when these broadcasts were made, and for some time had been earning their keep as scientific and mathematical tools. The Manchester machine, for instance, besides being used by Turing for his research, was available for outside research projects such as the UK's first atomic bomb project.[30] As for the Cambridge machine, Wilkes's broadcast summarized some of its research uses. For some months, indeed, Wilkes had been thinking about a replacement machine, funds for which were being canvassed at the time of his broadcast.[31]

Another context in which to consider these broadcasts was the Third Programme itself and its relationship to other BBC radio services. (Television was still very much a minority medium throughout most of the 1950s in the UK.) It is at the very least surprising that the Third Programme, which carried no regular science output, carried these broadcasts and several others relating to computers, whereas the Home Service, which broadcast a weekly science program, should have had relatively few broadcasts on the subject (only four significant broadcasts in the period 1950–1955). In my view, several factors account for this.

To a degree the Home Service's science-magazine approach would have been more strongly driven by news values than was coverage on the Third Programme, and although computers were "new" during the early 1950s, they ceased to be news as they evolved from being laboratory projects into scientific tools. Also, the Home Service's own news bulletins (as opposed to science broadcasts) would probably have covered computers at their most topical and newsworthy moments. There was, for instance, an item on a Home Service news broadcast on 9 December 1946 covering the announcement of the ACE project. The Third Programme, by contrast, was not so concerned with the news agenda and did not even carry news bulletins during that period. Thus, by being less driven by a news agenda, the Third Programme's coverage could afford to be more reflective and long term.

The most significant factor in the Third Programme's coverage, however, appears to have been a remarkable producer of talks on the Third Programme: Theophilus Stephen Gregory. He was responsible not only for the five BBC broadcasts but also for most of the other computer-related broadcasts on the Third Programme during this period. Gregory was a singular character, having been a Methodist minister during the 1920s, later converting to Roman Catholicism.[32] His particular areas of interest were philosophy and theology, and it seems probable that the contemporary debates about "electronic brains" caught his attention in a way that they might not had he simply been a science-trained producer of science broadcasts. For instance, a year before the five broadcasts, Gregory had produced two talks entitled "Mind-like Behaviour in Machines," both given by Donald M. MacKay, a physicist at King's College, London, with a particular interest in the compatibility of science and religious faith. Later broadcasts produced by Gregory had such titles as "On Comparing the Brain with Machines" (two broadcasts, again with MacKay as the speaker), "Machines and Human Behaviour" (with Frank H. George), and so on. Further evidence of the philosophical nature of Gregory's interest in computers can be found in a continuity announcement that survives with the BBC transcripts and which would have been written by Gregory himself (the emphasis in the following quote is mine):

> This evening we are repeating the first of five talks on the history and theory of *thinking mechanisms* ....

Whatever the particular bias of Gregory's interest in the subject, however, he appears to have made no attempt to influence the broadcasters in the content of their talks, if the experience of one speaker can be taken as typical. Wilkes reported in the September 2001 interview with me that Gregory gave him no briefing about his talk, and made no intervention apart from requesting him to alter the pronunciation of certain words (to no benefit, in Wilkes's view).

Approximately seven months after the series of talks discussed here was broadcast, Turing and Newman took part in another computer-related broadcast produced by Gregory on the Third Programme. This was a panel discussion entitled "Can Automatic Calculating Machines Be Said To Think?," broadcast on 14 January 1952 and repeated on 23 January 1952.[33] Other participants included Geoffrey Jefferson, a professor of neurosurgery at the University of Manchester, and Richard Braithwaite, a philosopher and Fellow of King's College, Cambridge. However, apart from this panel discussion, none of the broadcasters mentioned here took part in any further computer-related broadcasts on the BBC. Other broadcasts relating to computers continued to be made, however, and often by distinguished speakers.

As for the significance of these broadcasts to either the development of computing in Britain, or to the public understanding of the subject, these are imponderable matters pending further research. However, one concrete outcome deserves mention, even if it could not be said to be typical. After hearing Turing's broadcast, enterprising amateur computer enthusiast Christopher Strachey—at the time a teacher at Harrow School—dashed off a letter to the speaker:[34]

> Dear Turing,
>
> I have just been listening to your talk on the Third Programme. Most stimulating and, I sus-

> pect to many people, provocative, but it fits extraordinarily well with what I have been thinking on the subject.

The remainder of Strachey's four-page letter consists of his observations on the idea that making a computer think would be similar to the process of teaching, a matter touched on in passing by Turing and clearly related to Strachey's professional interests as a teacher. In closing, Strachey wrote:

> Please excuse such a long letter—I am quite sure you are far too busy to answer it—you must blame your talk for being far too stimulating.

Strachey's letter thus stands as an example of the power of this type of broadcast to stimulate and illuminate. Nor was this letter the end of the matter for Strachey. He went on to become a remarkable theorist of computer programming and the founder, in the 1960s, of the Programming Research Group at the Oxford University Computing Laboratory.[35] Although I would not wish to imply that Strachey's subsequent career was entirely attributable to his hearing Turing's talk, the talk was nevertheless part of his intellectual background and an inspiration to him. Fittingly, Strachey was himself later to broadcast at least three times on computer-related matters on the BBC.

Acknowledgments

I am grateful to Sir Maurice Wilkes for his many helpful comments on a draft of this article, and to Jeff Walden and the staff of the BBC Written Archives Centre at Caversham, Reading, UK.

References and notes

1. B.J. Copeland, "A Lecture and Two Radio Broadcasts on Machine Intelligence by Alan Turing," *Machine Intelligence*, vol. 15, K. Furukawa, S. Michie, and S. Muggleton, eds., Oxford Univ. Press, 1999, pp. 445-446 and 448-453.
2. A. Hodges, *Alan Turing: The Enigma,* Vintage 1992, pp. 441-442.
3. K. Furukawa, S. Michie, and S. Muggleton, eds., *Machine Intelligence,* vol. 15, Oxford Univ. Press, 1999, pp. 462-465.
4. Neither of Turing's radio broadcasts is reprinted in the most relevant volume of his collected works, namely *Collected Works of A.M. Turing, Mechanical Intelligence*, North Holland, 1992, D.C. Ince, ed., (part of the *Collected Works of A.M. Turing* series).
5. The script is item AMT/B/5 in the Turing Archive, King's College, Cambridge; http://www.turingarchive.org/browse.php/B/5.
6. M. Campbell-Kelly, "Programming the Pilot ACE: Early Programming Activity at the National Physical Laboratory," *Annals of the History of Computing*, vol. 3, no. 2, Apr. 1981, p. 133.
7. M. Campbell-Kelly, "Programming the EDSAC: Early Programming Activity at the University of Cambridge," *Annals of the History of Computing*, vol. 2, no. 1, Jan. 1980, p. 7.
8. M. Campbell-Kelly, "Programming the Mark I; Early Programming Activity at the University of Manchester," *Annals of the History of Computing*, vol. 2, no. 2, Apr. 1980, pp. 130-131.
9. A. Briggs, *Sound and Vision*, Oxford Univ. Press, 1979, p. 66.
10. H. Carpenter, *The Envy of the World: Fifty Years of the BBC Third Programme and Radio 3, 1946–1996*, Weidenfeld and Nicholson, 1996, pp. 96, 109.
11. B. Paulu, *British Broadcasting: Radio and Television in the United Kingdom*, Univ. of Minnesota Press, 1956, p. 368.
12. A. Briggs, *Sound and Vision*, p. 83.
13. A. Jones, "Pioneers on the Air," *Actes du sixième Colloque sur l'Histoire de l'Informatique et des Réseaux*, Éditions ACONIT, 2002, pp. 14-28.
14. M. Croarken, *Early Scientific Computing in Britain*, Oxford Univ. Press, 1990, pp. 50-53.
15. D. Hartree, "The ENIAC, an Electronic Computing Machine," *Nature*, vol. 158, no. 4015, 12 Oct. 1946, pp. 500-506.
16. D. Hartree, *Calculating Instruments and Machines*, Univ. of Illinois Press, 1949.
17. "An Electronic Brain," *The Times*, 1 Nov. 1946, p. 2; "The Mechanical Brain," *The Times*, 11 June 1949, p. 4; and "The Mechanical Brain," *The Times,* 16 June 1949, p. 2.
18. M. Campbell-Kelly, "Introduction," to *The Preparation of Programs for an Electronic Digital Computer* by M. Wilkes, D. Wheeler, and S. Gill; Charles Babbage Inst. Reprint Series, Tomash, 1982 (first published by Addison-Wesley, 1951), pp. xix-xx.
19. M. Croarken, *Early Scientific Computing in Britain*, pp. 119-122.
20. A.M. Turing, "Computing Machinery and Intelligence," *Mind*, vol. LIX, no. 236, 1950, pp. 443-460.
21. A.M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proc. London Math. Soc.*, vol. 2, no. 42, 1936, pp. 230-267.
22. M. Armstrong, *The Listener*, 24 May 1951, p. 851.
23. M. Campbell-Kelly, "Programming the Mark I; Early Programming Activity at the University of Manchester," *Annals of the History of Computing*, vol. 2, no. 2, Apr. 1980, pp. 130 and 165.
24. M.V. Wilkes, *Memoirs of a Computer Pioneer,* MIT Press Series in the History of Computing, B. Cohen, ed., 1985, MIT Press, p. 184.

25. M. Campbell-Kelly, "Programming the Mark I; Early Programming Activity at the University of Manchester," *Annals of the History of Computing*, vol. 2, no. 2, Apr. 1980, p. 145.
26. M. Campbell-Kelly, "Introduction," to *The Preparation of Programs for an Electronic Digital Computer* by M. Wilkes, D. Wheeler, and S. Gill, p. xiv.
27. M. Campbell-Kelly, "Programming the EDSAC: Early Programming Activity at the University of Cambridge," *Annals of the History of Computing*, vol. 2, no. 1, Jan. 1980, p. 12.
28. M.V. Wilkes, "Can Machines Think?," *The Spectator*, 10 Aug. 1951, pp. 177-178.
29. "'Memory' in a Mercury Tube," *Science News*, vol. 5, Penguin, Nov. 1947, pp. 64-66; G. Rabel, "Mathematical Instruments and Calculating Machines," *Science News,* vol. 7, Penguin, June 1948, pp. 112-124; and S. Byard, "Robots Which Play Games," *Science News,* vol. 16, Penguin, June 1950, pp. 65-77.
30. A. Hodges, *Alan Turing: The Enigma,* p. 440.
31. M.V. Wilkes, *Memoirs of a Computer Pioneer,* pp. 184-188.
32. H. Carpenter, *The Envy of the World: Fifty Years,* p. 193.
33. A copy of the BBC transcript of this broadcast is held in the Turing Archive at King's College, Cambridge, as item AMT/B/6. The text is available online at http://www.turingarchive.org/browse.php/B/6 and has been published in B.J. Copeland's "A Lecture and Two Radio Broadcasts on Machine Intelligence by Alan Turing," *Machine Intelligence*, vol. 15, K. Furukawa, S. Michie, and S. Muggleton, eds., pp. 465-476. The broadcast is also discussed by Copeland here on pp. 453-458 and by A. Hodges, *Alan Turing: The Enigma,* pp. 450-452.
34. Turing archive, item AMT/D/5, http://www.turingarchive.org/browse.php/D/5.
35. For a biographical note about Strachey, see M. Campbell-Kelly, "Christopher Strachey, 1916–1975: A Biographical Note," *Annals of the History of Computing*, vol. 7, no. 1, Jan. 1985, pp. 19-42.

Allan Jones is a lecturer in the Department of Information and Communication Technologies at the Open University, the UK's largest distance-teaching institution. His interests include communication technology, music and the history of science broadcasting. Jones holds engineering and science degrees from Liverpool University and the Open University.

Readers may contact Jones at a.jones@open.ac.uk.