1. Human vis-à-vis Artificial Intelligence

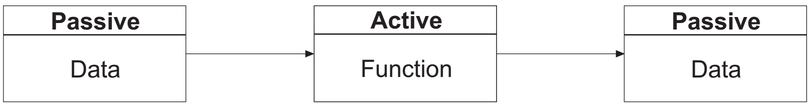
Paul Hague

February 2011

On 17th February 2011, *The Independent* newspaper in the UK carried this headline: 'Computer puts human supremacy in jeopardy.' It was reporting on the way that an IBM computer called 'Watson', named after IBM's founder, had just beaten two champions in the Jeopardy! TV quiz show. This achievement followed on IBM's Big Blue defeat of Gary Kasparov at chess in 1997.

But do these demonstrations of computer power mean that machines are about to develop artificial intelligence exceeding human intelligence? If so, what would be the economic consequences of computers performing most of the cognitive jobs currently being done by human beings faster and more cheaply, including the skills of computer programming and information systems design? I have spent my entire adult life investigating these questions, since I joined the computer industry in 1964 as a mathematician/programmer. All I knew at the time was that human beings are good at pattern recognition but comparatively poor at arithmetic, while with the computer it is the other way round. Why is this?

Well, we can only answer this question through self-inquiry, by discovering what it truly means to be a human being free of all religious, scientific, and economic conditioning, which causes us to behave more like machines than the Divine, Cosmic beings we truly are. To see how we can become free of our mechanistic collective, cultural, and personal conditioning, we first note that computers function entirely in the horizontal dimension of time, as this input-output diagram illustrates.



The distinction between active and passive denotes that functions that form programs in computers are data in a computer's memory just like the 'raw' data of numbers and strings that they process. This was not the case with the early electromechanical computers in the 1940s, when programs were external to the computer, executed directly from paper tape or set up in switches. Programs only became internal to computers with the invention of the stored-program computer, the first such machines being built at the universities of Manchester and Cambridge in the UK in 1948 and 1949, respectively.

This situation raised the possibility that computer programs could modify themselves in midflight, so to speak, or even create new functions dynamically. One of the first computers I programmed in the 1960s—IBM's System/360—had instructions that could modify other instructions at the machine level. But it was not until the late 1970s, when I was engaged in marketing IBM's personal computing software products before they introduced their hardware PC, that I discovered a high-level language with a similar capability.

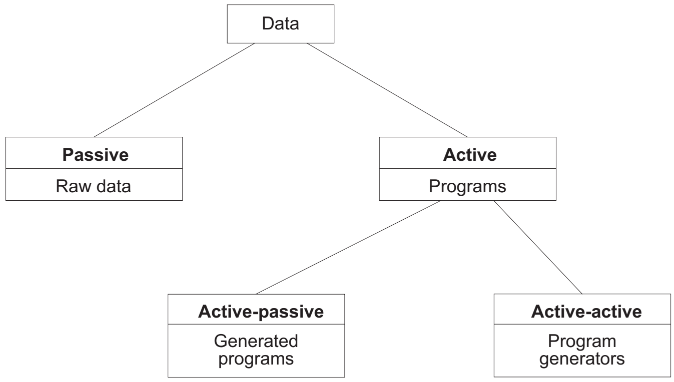
This language was A Programming Language (APL), initially developed by Kenneth Iverson at Harvard University in the late 1950s as a concise mathematical notation to assist students in analysing various topics in data processing. APL became IBM's principal management information tool in the 1970s after Iverson joined IBM. Now, while APL is a function-based language, like many others, unusually it also has system functions—⎕*CF* and ⎕*FX*—which convert functions to strings and back again. IBM's APL Data Interface (ADI) used these system functions to create APL functions to query the company database.

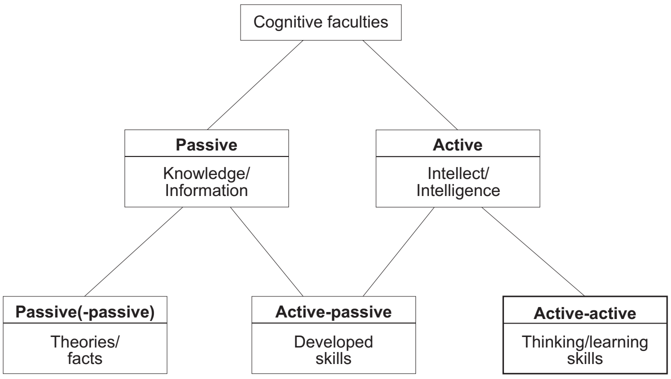
But could a computer program itself without human intervention? Ada Lovelace, the daughter of Lord Byron and his wife Anne, a poet and mathematician, respectively, did not think so. In a brilliant memoir on Charles Babbage's Analytical Engine, the first design for a general-purpose computer, she wrote in 1843:

> The Analytical Engine has no pretensions to originate anything. It can do whatever we know how to order it to perform. It can follow analysis; but it has no power of anticipating any analytical relations or truths. Its province is to assist us in making available what we are already acquainted with.

Alan Turing, the founder of the mathematical theory of automata in 1936, did not agree with her. In a famous article in the philosophical journal *Mind* in 1950, which began with the words, "I propose to consider the question 'Can machines think?'," he asserted "I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted."

So how can we represent human thinking in a computer? Well, as this diagram illustrates, there are two types of active data in computers, generated programs, like Adobe Photoshop and Microsoft Word, and program generators, like C++ and APL, which are examples of compilers and interpreters, respectively. So if a computer could think for itself, an APL function would need to dynamically create a new function entirely on its own.





We can see that this is not possible most clearly from the analogous qualities in human beings, as this diagram illustrates. Our developed skills, like playing chess, correspond to generated programs, while our thinking and learning skills correspond to program generators, like APL. The reason why we human beings are able to create APL functions from scratch and APL functions cannot is that the creative power of Life arises directly from our Divine Source in the vertical dimension of time, as my book on *Wholeness* describes in minute detail.

This book also answers another fundamental question about human existence. Information systems architects can develop entity models showing the relationships between the passive entity types in an enterprise, such as customers, products, and deliveries, and process models, depicting the active processes that deal with these entities, such as manufacturing, ordering, and invoicing, and their relationships to each other. But is it possible to develop a model of the modelling process itself?

Indeed, it is, with self-reflective Intelligence, the Divine quality that distinguishes human beings from the other animals and machines, like computers. To create such an all-inclusive model, we conduct a thought experiment in which we imagine that we are a computer that switches itself off and on again so that it is completely empty. Starting afresh at the very beginning, free of our mechanistic conditioning, without any external authorities or other influences, this computer then has the task of integrating all knowledge into coherent whole, including the knowledge of how to develop such a comprehensive map of the Universe, rather like a TV camera filming itself filming, illustrated by Escher's famous lithograph 'Drawing Hands'. From this life experience, the greatest adventure on Earth, we can thus see that human beings are the leading edge of evolution, not computers, requiring us to make radical changes to the mechanistic work ethic that prevails today if we are to intelligently adapt to the most momentous turning point in evolutionary history, as the essays 'Our Evolutionary Story in a Nutshell', 'Society and the Individual', and 'Returning Home to Paradise' describe.