



Purposes, Algorithms and Real Skills

Ian pyle*

The University of York, Department of Computer Science, England, United Kingdom

Abstract

Technology has always affected society by changing patterns of employment. With computers, particularly Artificial Intelligence, products and services can be provided more efficiently but at the human cost of job losses. Many traditional skills have been undercut, and for people to have satisfying employment we need to identify the areas in which human skills remain important. By recognising the nature of algorithmic processes, in contrast with purposeful ones, we see that the human skills of caring, motivating, negotiating, and inventing remain as the real skills that our society has to value.

Introduction

The advent of computers in the last century has made a significant change (for at least some) to what it means to be human. "Artificial Intelligence" has caught the public imagination, with the "Turing test" [1] as a touchstone for distinguishing a human from a computer in a conversation. And the progressive introduction of the use of computer-controlled machine tools and factory automation has made dramatic changes to the nature of work and economic activity for many people. This note explores the relationships between computers and people, seeking to identify the balance that will make best use of both without losing the value of the other.

In this paper we look for the kinds of work that people will do, in contrast with the kinds of work that computers can do (and in general will do more efficiently). People need gainful employment, to be active participants in a community.

Some traditional skills can be expressed algorithmically, but there are significant human abilities (not always recognised to be skills) which cannot. With the advances in automation, particularly with "Artificial Intelligence", the industrial and economic structure of employment is changing, and it is important to recognise the value of "real" skills which cannot be carried out by automata. We investigate this distinction by focussing on the purpose of an algorithm in contrast with the algorithm itself, and the resulting behaviour.

Computer and human activities

In spite of appearances, the activity of a computer, and consequently the behaviour of an autonomous system (i.e. a robot), is fundamentally different from the behaviour of a human being, so expectations can be misleading. An autonomous system (as any computer-controlled machine) acts in accordance with its controlling algorithm, rather than for a specific purpose or to behave in a way that achieves a desired situation. Purposeful behaviour is intrinsically flexible, capable of adjusting its algorithm according to circumstances, particularly unexpected changes. Algorithmic behaviour can take account of expected circumstances, but all possibilities must be covered by the program.

A computer cannot care what it does [2]. It experiences neither pleasure nor pain in performing an algorithm. It does not feel pride on achieving a good outcome, nor remorse on causing harm. It does not take account of causation, and cannot carry responsibility for the consequences of its actions.

Harvey [3] explains the physiological differences between computers and brains, but gives only brief coverage of the differences in behaviour. In contrast, this paper concentrates of these differences, distinguishing between behaviour which is entirely algorithmic and that which is purposeful. Human behaviour, including that of programmers, is purposeful, but the behaviour of a computer (even when using AI techniques) is entirely algorithmic.

Purpose and behaviour of an algorithm

Algorithms do not arrive spontaneously "out of thin air": they are deliberately constructed by a human programmer (and possibly modified by the performance of another algorithm, which had also been written by a human programmer). The derivation of an algorithm for a particular purpose is not straight-forward, neither is the derivation from an algorithm of what its behaviour will be, other than by executing it for specific cases.

Article Information

DOI: 10.31021/acs.20193120
Article Type: Review Article
Journal Type: Open Access
Volume: 3 **Issue:** 1
Manuscript ID: ACS-3-120
Publisher: Boffin Access Limited

Received Date: 24 January 2020
Accepted Date: 06 Apr 2020
Published Date: 08 Apr 2020

*Corresponding author:

Ian pyle
The University of York
Department of Computer Science
England, United Kingdom
E-mail: ian.pyle@cantab.net

Citation: Pyle I. Purposes, Algorithms and Real Skills. Adv Comput Sci. 2020 Mar;3(1):120.

Copyright: © 2020 Pyle I. This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 international License, which permits unrestricted use, distribution and reproduction in any medium, provided the original author and source are credited.

Sometimes it is possible to construct an algorithm to achieve a given purpose, by manipulating the expression of the purpose. However, it has been found increasingly difficult to find algorithms to solve important problems (such as Affective Computing, see ref [4]. The techniques of Artificial Intelligence (AI) have often been found successful in applying a generalised technique of problem solving to new problems, using a variety of techniques that are sometimes successful. Programs almost always contain errors (colloquially “bugs”) that cause them to work in unexpected ways, usually detrimentally. There are many ways of finding faults in a program, summarized in part 3 of ref [5], of which the most significant is formal analysis: deriving from the algorithm the overall behaviour of a program. This is not easy, and has not (yet!) been found applicable to AI techniques. For conventional programs, we describe the behaviour of a procedure as a mapping between the state space on starting and the state space on completion. For a complete program the resulting behaviour is a mapping between the state space of the world observed at the beginning and the subsequent state space on finishing the program. A program is deemed to be “correct” if that mapping is consistent with the purpose of the program, as specified in the initial “requirements.”

Creating an algorithm

Creating an algorithm requires real skill, but there is no algorithm that can do it. (There are of course good algorithms for translating an algorithm from one form to another, which is what a compiler does.) There are many “methods” used in Software Engineering for articulating the purpose of a computer-based system, and for designing the software that is intended to achieve that purpose. (A number of them are described and analysed in [6], for example SSADM, MASCOT, SDL, and JSD.) These methods provide frameworks and outlines for human activities but in no way are they algorithmic. Most are supported by computer-based tools to manipulate and check intermediate results, but there is always essential human input.

Responsibility

Fundamental to this discussion is the issue of consideration of what is to be done when something goes wrong? Is someone or something responsible? The actions and behaviour of a machine can be harmful as well as beneficial, who is to be held responsible for bad behaviour or mis-behaviour? The moral problem is clear, but solutions are not.

If a computer-controlled aeroplane or car crashes, society requires someone to carry the blame. It can be argued that the programmer who designed the software is responsible, but what about the person who commissioned it and agreed the specification, or the person who approved its use in this way?

This is connected with the issue of purpose or intention. Is the purpose of using the computer to save money or to provide a more effective service? Neither coincide with (and often may conflict) of a human purpose of performing a useful (and as paid) employment. It is a mistake to presume that computers have volition, or strive for power, or can be held responsible for what they do.

Cause and effect

Human psychology has evolved to give special prominence to the cause-effect relationship: we instinctively learn which muscles to activate to perform elementary actions such as breathing, walking, talking, eating, or seeing our environment. We know how to move around in our locality, and how to respond to people who smile at us. This is now called Moravec's paradox [7].

The cause-effect relationship works both ways: we are aware of (likely or possible) effects of the actions we take, and we can choose to act in ways that will (probably) achieve desired effects. Thus we take moral responsibility for the actions we do.

In contrast, computers have no awareness of the outcomes of their actions: they just perform the sequences of elementary actions

prescribed by their program, regardless of context other than as determined by the program.

Causality is not algorithmic: hence the problems we have with weather forecasting, even more with economic forecasting and hopeless with political forecasting.

Safety

Computers used to control dangerous machinery have to ensure that potentially dangerous actions are only performed when it is safe to do so [5]. In this respect, the computer is taking responsibility for the outcome of its behaviour – but only within a very limited environment. Here perhaps is the beginning of responsibility and of intelligence.

Automation – the use of computers to control machinery – extends this principle, and is the source of the loss of many jobs traditionally classed as skilled. Here, in a circumscribed environment, we give authority to a machine to control its own behaviour. There is no direct human control, so where is the responsibility?

Skilled and Unskilled Work

From the beginning of human societies, some people have specialized in particular kinds of activity, resulting in their becoming skilled in that activity – hunting, farming, cooking, house-building, pottery, metal-working, etc., with overall benefits for that society. Skills were first passed down the generations, then by apprenticeships governed by guilds, and subsequently by formal training courses with explicit goals and achievement levels. Human language was essential for this. Skilled workers were valued members of the community, appreciated and well-rewarded.

In contrast, unskilled workers did not need specialised training and in general were less well respected socially, and less well paid.

As technology developed, and new tools became available, appropriate skills emerged that made beneficial use of them. But the skills that were now replaced lost their value, and communities dependent of former skills were disrupted. The problem now is to identify which skills can not be automated, so that training and employment match the needs of people as employees, rather than as employers.

This contrast can be seen as similar to Moravec's paradox [7] that “it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility”. This is attributed to the fact that, unlike checkers, physical dexterity has been a direct target of natural selection for millions of years.

Industrial revolutions – Changes in patterns of employment

With changes in technology, the kinds of skill needed by society have changed, resulting in industrial revolutions. The first industrial revolution happened because of the technology for handling energy. Steam engines could supply energy more conveniently than horses, and mass production made many traditional skills no longer necessary. People had to discover new skills as the previous need for farriers, blacksmiths, shepherds, casual agricultural labourers diminished. With the advent of mass production, training for traditional skills taught people to behave like machines.

We are now experiencing the revolution arising from the technology for handling information: we have machines that can behave without the disadvantages of human fatigue, inattention, or clumsiness. Computers can handle information more conveniently than anything previously, and the effects are even more dramatic. The skills that were previously distinctively human, such as precise manipulation of materials, or draughtsmanship for engineering or architecture, could be easily done by a computer: with computer

controlled machine tools and computer-aided design. There are now robots for daily life activities such as floor cleaning, or lawn mowing.

For automation (the application of computing to industrial processes), the major benefits of consistent replication, reduced running costs and almost continuous availability are generally considered to outweigh the harmful consequences on employment and social disruption.

New skills are needed for handling computer-based systems, but (more importantly) some old skills have to be re-valued, particularly human activities that are instinctive and do not require training.

Numerical calculation

Computers are good at doing numerical calculations. Babbage was worried about errors in mathematical tables (used for navigation, surveying and similar purposes) and designed his engines to get reliable tables. From then on, computers have been widely used to calculate numbers, whether evaluating formulae, solving differential equations, or calculating salaries or tax allowances. People doing this work are recognized as highly skilled - yet such behaviour by machines has never been treated as displaying intelligence!

Algorithmic and Purposeful Activity

The essential difference identified here is between following a series of instructions (algorithmic operation) and acting towards reaching a specific goal (purposeful). A computer acts as prescribed by the program it is executing, consisting of algorithms and data structures. Human behaviour (when instinctive or deliberate, rather than thoughtless) is directed towards achieving a particular goal, which we describe as purposeful. The morality of such behaviour is determined by the co by some prospective benefit.

Within an algorithmic process, there can be a part which appears to be purposeful: for example, an iteration, executed repeatedly until a completion criterion is met. This appears to be working towards a goal - but it is the programmer who has decided the criterion for completion, and ensured that the iteration step is convergent. The computer has no such motivation.

Having an intended outcome and focussing on the end rather than the means is the distinctive feature of human behaviour, even when this involves following a set of instructions or guidance. Thus knitting from a pattern, cooking from a recipe, or playing music from a written score all have a quasi-algorithmic basis, there is an essential underlying purpose: to produce something of functional or aesthetic value.

Following an algorithm is intrinsically different from trying to make something happen (or to prevent it from happening).

Real Skills

The skills needed in the current climate are completely and dramatically different from the traditional skills of previous generations. If a job can be done by a computer (i.e. if it can be done algorithmically), doing it was a skill that no longer can only be done by a person - and in many cases it is done more effectively (and more cheaply) by a computer.

What remains are the real skills: needing human awareness, insight, inventiveness, empathy and compassion and social sensitivity, which have to be done purposefully.

Some skills depend on the ability to act in a quasi-algorithmic way, using an existing formula for success: for example cooking from a known recipe, knitting from an existing pattern, or playing music from a written score (see section 6 below).

Caring

Examples of real skills abound from everyday life: home-building, gardening, picking up litter, cleaning rubbish from the environment.

Social skills such as nursing, nurturing babies, bringing up children, good parenting and caring for the frail and elderly must also be recognised as real skills. (Many of our current social problems arise from failures in these skills.) These are activities that involve caring: for someone, or something. Most traditional "unskilled" work has this characteristic, thus calling into question the essential nature of a "skill". This appears to be an extension of Moravec's paradox: the activities that humans can do instinctively, without formal training, are the most difficult for a computer to be programmed to do.

Inspiring

Foremost is the skill of motivating or inspiring someone to do something. We have free will, and make our own choices about what we do, within current constraints. Education is domain where activity has to be purposeful rather than following a set of rules (despite some government dogma) inspiring students to learn, and identifying where they have lost a significant connection.

Negotiating

Another area (among many) where real (human) skills are needed is in negotiation: commercial, social and political; where there are many different criteria involved, and the relative importance of them differs from person to person. Similarly with rhetoric: explaining a condition or relationship and persuading others of its significance.

Inventing

Creativity in general cannot be algorithmic. Whether artistic (such as music, painting, or architecture) or any branch of engineering, there may be rules for guidance in particular styles or to satisfy particular constraints, but the critical element is inventiveness and innovation. This includes inventing an algorithm (to achieve a particular purpose), or a knitting pattern. Cooking a meal using left-overs requires more skill than warming up a pre-prepared meal.

Producing sounds by scraping, shaking, banging or blowing is a skill, as is coordinating the efforts by a number of other people in an orchestra, based on the original skill of the composer.

Support for Real Skills

In many fields, there is a role for algorithms or computers in subsidiary tasks, such as communication or information storage - but the primary purpose is to achieve some personal, human or social benefit: recognising the value of the activity. Following a recipe from a cookery book can produce a good meal, but there is skill in selecting appropriate recipes and in timing their production so that the components are at the right temperatures when needed. And where did the recipes come from? A serious cook does not necessarily follow an existing recipe, but can imagine how a combination of ingredients could taste, and creates a new meal from them. Similarly knitting a jumper by following a pattern needs skill to choose colours and make the texture uniform, which is different from creating a new garment imaginatively.

Harmful Skills

As well as the above categories of traditional skills (that can be replaced by computers) and the real human skills (that cannot be replaced although they can be supported by computers), we recognise a third category, of human "skills" that are enhanced by computers. Regrettably, the human ability to deceive, mislead and exploit others is a skill that computers have made more prominent. Scams, hacking, identity fraud have always been problems for society (from snake-oil salesmen, charlatans and fraudsters) that are recognizably immoral, but virtual reality and social media have made it easier for the unscrupulous to take advantage of the invisibility or anonymity given by computers. The impact of a computer's activity on its own or others' wellbeing is rarely considered by its programmers, so we can characterize its behaviour as at best amoral, but with a bias towards being immoral.

Morality

Attempts to produce Moral Machines by teaching robots right from wrong [8] fail because of the focus on algorithms and the difficulty of characterizing the difference between right and wrong without considering the context or causality. To act without taking account of the consequences of the activity is to behave like Pinocchio without Jiminy Cricket. To be moral, a computer needs not just Artificial Intelligence but an Artificial Conscience.

Technology is not morally neutral - it can be harmful as well as beneficial. We, who are the innovators and developers of the technology for handling information, must accept our responsibility for the harm this technology causes to individuals and communities. With Artificial Intelligence developing automation into Autonomous Systems (robots), the problems become more acute. Previous generations have focussed on correctness and safety; we must now address morality: we must mitigate the harm. Not only by making "moral machines" but by making machines be moral.

Conclusion

The social consequences of automation include the loss of skilled jobs, leading to disruption of traditional employment patterns and the re-evaluation of the nature of a "skill." What used to be recognised as important skills (such as numerical ability and dexterity in using machine tools) can now be done automatically; counter-intuitively, what used to be considered as "unskilled" work is beyond current automation: work involving caring, whether for children, old people, disadvantaged people, or for maintaining buildings, machines, animals and the environment.

The underlying problem is that these "real" skills are not currently valued for their contributions to society, and our economic structures do not adequately reflect their importance. The greater the impact of automation, the more we must recognise the social value of human skills.

References

1. Turing AM. Computing Machinery and Adv Intelligence Comput Sci Mind.1950 Feb;433-460.
2. Pyle I C. Do computers care.2020 Jan;3(1).
3. Harvey RJ. Can Computers think? Differences and similarities between computers and brains.Progress in Neurobiology.1995 Feb;45(2):99-127.
4. Picard RW. Affective Computing. (M.I.T)Media Laboratory Perceptual Computing Section Technical.
5. Pyle IC. Developing Safety Systems: A Guide Using Ada.1991;240.
6. Pyle I, Hruschka P, Lissandre M, Jackson K. "Real-Time Systems: Investigating Industrial Practice". Wiley.1993.
7. Moravec HP. "Mind Children". (Harvard University Press 1988). From Wikipedia, the free encyclopedia.
8. Wallach W ,Allen C."Moral Machines" (Oxford).2009.