Automation and Ethics

34

Micha Hofri

Contents

Abstract

This chapter surveys aspects of automation in technology, its design, implementation, and usage, as they interact with values that underpin our society and its civilization. The framework chosen for this survey is that of moral or ethical theories and attitudes. To this avail we describe several of the ethical theories used currently to anchor discussions about values in our society. Several significant failures of automatic systems, including a fictional one, are quarried for ethical insights and lessons. The currently intractable nature of "neural networks" artificial intelligence systems trained via machine learning is shown to be a moral conundrum. The concepts of code of ethics and of ethical analysis are presented in some detail.

M. Hofri (✉)
Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA, USA
e-mail: hofri@wpi.edu

Keywords

Ethical framework · Theory · Model · Conflict analysis · Role of ethics in automation design · Automation effect on society · Automation failure analysis · Artificial intelligence opacity · Internet and communications technology · Code of ethics and professional conduct

34.1 Introduction

Mankind has evolved to be the dominant species in the world. On the way humans have developed technologies that are changing this world.

A convenient time point for the beginning of our "technological age" is 1804; that year the first steam locomotive was built for a railroad in a coal mine in Wales, and the US Census Bureau estimates that on that year the global population reached one billion, for the first time. It is now (at end of Summer, 2022) just under eight billion. This growth has required the development of many technologies, notably in food production, construction, transportation, and communications. We now need these technologies; the current way much of humanity lives depends on them and on many others that provide for needs and wants beyond mere subsistence; with time, this dependence is certain to increase in depth and criticality. On the way to the present, people have discovered that these technologies, and their applications, have downsides as well. This chapter considers the ways that the balancing is handled. The framework which has been found useful for the needed consideration is that of ethics, also referred to as morality. Note that "the ethics of technology" is a topic in numerous publications, where the actual discussions have little to do with ethics: mostly they raise worries of technology influence on society, or report about damage due to technology, and discuss ways to control and repair these concerns; the discussion is technical or legal (regulatory) rather than ethical; that usage is not

S. Y. Nof (ed.), *Springer Handbook of Automation*, Springer Handbooks,
https://doi.org/10.1007/978-3-030-96729-1_34

considered in this chapter. It is a sign of the times that many universities have obtained generous grants to build research centers focusing on this topic. Such centers tend to have brief lives (probably due to the grant that started them petering out).

The quotation heading the next section suggests an intuition behind the choice of our topic.

34.2 What Is Ethics, and Why Is It Used for Automation?

> Two things fill the mind with ever new and increasing admiration and awe …. the starry heavens above me and the moral law within me. …. I see them before me and connect them immediately with the consciousness of my existence.
>
> —Immanuel Kant, Critique of Practical Reason, 1788.

As Kant claims, a sense for ethics, a "moral law" in his words, is innate to humans. Similar realizations have moved thinkers to analyze this sense in the earliest writings of mankind [1], going back to Classical Greece. Our moral sense is needed, so that we act properly, in ways that this sense approves of, and it lets us then feel we are "doing the right thing." Humans have other senses, such as vision and touch, and we have found that they can be enhanced by instruments, such as a microscope and a thermometer. You may think of the various ethical theories and models that people have spawned as tools to sharpen our moral sense

The most direct answer to the question in the title of this section is to show how easy it is to draw a list of woes, scenarios of automatic systems or operations which give rise to undesirable, or even catastrophic results; these would not be the outcomes of "doing the right thing." This can happen in many and diverse ways:

1. Traffic lights that "favor" one road over another intersecting it, creating large, unneeded delays. A distributed system of such lights can have even more elaborate failure modes, such as the creation of a "red wave" along a main road.
2. Airline reservations system open to the Internet makes it possible for anyone to discover who flies on a given flight by using an ordinary Internet search engine.
3. A driverless car which does not allow adequately for weather conditions.
4. The machines in a printer room of an academic department manage the queue of documents to be printed, unaware that some of them are confidential (such as tests for current courses), and the room needs to be monitored when these are printed.
5. Automatic assembly machine which manufactures products inferior to those that humans employees produced before. Customer complaints would now be answered by blaming the machine, "the computer did it."
6. Automatic irrigation system which is not provided input on the state of the soil and predictions of precipitation and wastes water/or spaces the watering sessions too sparsely, and plants die off.

In the next section, we give a more detailed description of the concerns with automatic systems, which can be ameliorated through a disciplined process of ethical analysis of the decision space. The list of examples above where faulty design led to unanticipated behavior suggests how much decision-makers could be helped by such analysis.

Often failures of automation are assumed to be due to its novelty; this need not be the case. Automation came early: The above mentioned beginning of the technical age cited the use of a steam engine in a locomotive. Such steam engines were even then provided with a *centrifugal governor* that regulated the flow of steam to the cylinder and maintained the engine speed within a prescribed range, automatically. Now consider the examples in the list above; how do they differ from the early locomotive? They are all *computerized systems*. As will become evident, in much of the discussion of automation, we need hardly distinguish between the terms "automatic system" and "computer-based system." Similarly, the important distinction in computers between the hardware and the software is rarely material for us.

It is important to note: the creation and operation of an automatic system involves several categories of personnel, not all necessarily acting or even existing concurrently. Among those we distinguish designers, implementers, deployment teams, maintainers, operators, and possibly more. On another level there are the owner(s), decision-maker(s), and external users (think of an ATM), all of them, at different levels of involvement, stakeholders.

There is a basic question that underlies much of our discussion: does technology carry in and of itself any values, or is it value-neutral, and ethical concerns relate to it through its embedding in human society? This chapter hews closely to such an attitude; humans, their needs and wants—that is the measure of our ethics concerns. The question is treated in some detail in [28, Chapter 3].

34.3 Dimensions of Ethics

The discussion begins with the main kinds of theories of ethics that have been developed and then presents a list of issues or concerns that the personnel connected to automation needs to take into account.

34.3.1 Theories of Ethics

> The problem is that no ethical system has ever achieved consensus. Ethical systems are completely unlike mathematics or science. This is a source of concern.
> —Daniel Dennett, Interview with Hari Kunzru, December 17, 2008.

The concern Dennett raises leads some ethicists to refer to *ethical frameworks*, or *systems*, as he does, rather than theories. We use these collective names as equivalent. How can an ethical issue be addressed? We have an inherent moral sense, and it is natural for us to appeal to its guidance. It is, however, personal and as such cannot be the basis for moral principles that are widely accepted, ideally—by all society. What can be the source of authority or persuasion that can back such wide acceptance of an ethical principle? Traditionally we recognize three such sources: religion, the Law, and theories of ethics. We argue that only the last is a suitable choice [2, Chapter 2].

Using **religion** as the basis of ethical decisions has an attractive side; it has been practiced in various forms for a very long time, and for many among us it is invested with the recognition and demands of a higher authority, even a supreme being. However, in the diverse, pluralistic society we exist, this cannot provide a common source recognized by most of the people we consider. In addition, it may be pointed out that automation systems exist at a societal level that is not addressed well by most religious moral teachings. This cannot be an adequate basis. No more than we can appeal to the code of Hammurabi to order our lives.

Can the **legal system** provide us the support we need to determine the ethical value of actions? Unlike religious teachings it is recognized as impersonal, it is not subject to disputations, and it is quite explicit. There are difficulties in using it as source of consensus. It is not uniform across national borders (and in several ways even across state borders within the United States). The main reason we hesitate to see it as our source of moral authority is the incongruity: not all that is moral is legal; not all that is legal is moral. For example, till the 1860s slavery was legal in the United States; conversely, in many states, due to public health concerns, restaurants may not give charity kitchens their surplus prepared food. What is morally acceptable by society changes with time, and so does legislation, but their changes are rarely well-synchronized.

What theories of ethics are at our disposal? A readable survey of the area is provided by Michael Sandel in [6]. Over time, several types have been proposed:

1. Relativistic ethics frameworks
2. Consequence-based ethics theories
3. Duty-based ethics theories
4. Character-based ethics theories

This list immediately reinforces the statement of Daniel Dennett above.

Note We keep to the main flavors; the complete record would also mention social contract-based ethical theories, [3, 5] and rights-based theories [7]. When automation is considered, these are probably less significant than in other contexts.

Relativistic Ethics Theories

Such theories deny the existence of universal ethics norms [8, Chapter 2]. It came to be considered a valid approach in the twentieth century only [4]. Ethicists have defined two such approaches to ethics, with self-explanatory names.

The *Subjective Relativism* posits that each person makes moral decisions according to his individual values. The working principle is acceptance that "what is moral for me may not be right for you." While this may be a workable arrangement for a community of sensible people, it is not appropriate as the ethical protocol for managing a technology likely to be novel to many persons or organizations.

The second type of relativism, *cultural relativism*, enlarges the domain of agreement about moral norms from an individual to a community, which may encompass few or many individuals; yet similarly, it denies the need for agreement among such groups. The same difficulty holds: since the technologies we consider are expected to transcend any particular culture, its design and operation cannot be held hostage to any such local set of norms.

There is even a third view, sometimes called "ethical egoism," which defines the good and the bad actions or issues as those that advance or, conversely, harm the interests of a person. In such a view, any issue which neither abets nor harms the person has no moral dimension. Such thinking is commonly associated with the author Ayn Rand; it is not better than the relativistic formulations for our concern. David Oderberg is quoted in [17], saying that relative ethics ultimately dissolves into moral nihilism.

Albert Einstein observed "Relativity applies to physics, not ethics…."

Consequence-Based Ethics Theories

We look now at theories that judge an action moral to the extent that it leads to the increase of the happiness of the most people. This is the way it was offered in the eighteenth and nineteenth centuries by its founders, Jeremy Bentham (1748–1832) and John Stuart Mill (1806–1873), [2, 9]. The underlying idea is attractive, but its application requires attention, to avoid unacceptable outcomes. A known example suggests that if a society enslaves 1% of the population, making them produce desirable products and services for the

rest and improving their quality of life, many more people would experience greater happiness, swamping the suffering of the few, making it qualify as a highly ethical act!

Nevertheless, this approach has found much favor and has been modified in several ways, to improve its usability. While happiness is desirable by all people, comparing happiness levels provided by different acts is not easy, arguably impossible, and has been typically replaced by quantifying the merit, often by equivalent sums of money. Economists that have been using it found that a more flexible application is possible introducing the concept of utility function, to represent the merit of decisions [10]. Consequently, a common name for this approach is *utilitarianism*.

Unlike the relativistic approaches to ethics, utilitarianism is factual and objective and allows decision-makers to explain and justify their decisions. Yet another important development of the theory emerged as it was realized that there is merit in avoiding the need to conduct a complete analysis for each situation which needs an ethical decision, if we adopt *rule-utilitarianism*, which calls on us to formulate rules of action by analyzing their utility, rules that would allow us a simpler decision-making, by finding for a pending action the rule that would provide us guidance for it.

It has been observed that much of the merit of utilitarianism is that its application requires a systematic, exhaustive, prudential, and objective analysis of the decision space that is relevant to the act, or rule under consideration, as the case may be, guaranteeing, to the extent possible, that the decision-makers understand the context of the decision and are in command of the facts available.

Duty-Based Ethics Theories

Immanuel Kant, the German philosopher, claimed, as seen when Sect. 34.2 was introduced, that his consciousness is tied up with a moral law. Throughout his work he struggled to reconcile this fact with his conviction that morality of action must ultimately be grounded in the duty that speaks to the action, which mainly springs from the obligations that humans have to each other, but *never* in the possible consequence of the actions taken. He refused to tie ethics to the promotion of happiness in any form, explaining in his *Prolegomena to Any Future Metaphysics* that it is so, "…because happiness is not an ideal of reason, but of the imagination," and noting further that performing a duty may be unpleasant or even lead to a result which is undesirable from certain points of view. (Could this observation have led to George B. Shaw, acute social critic that he was, to observe "An Englishman thinks he is moral when he is only uncomfortable," in *Man and Superman*, Act 3, 1903?). But 'duty calls,' and this call is supreme. The term *deontics* or *deontology* is used to describe such theories. (A traditional view states that justice flows from three sciences: economics; deontics, study of duty; and juridics, study of law).

Kant had a second string to his bow; his conviction that while the moral law is an essential difference between humans and all other beings, there is another difference, our ability to reason and be rational. Hence, he stated, moral decisions need to be reached by rational analysis of the situation at hand and judging which moral rules or principles apply. Moral analysis, like the mathematical kind, needs a basis. In mathematics the need is provided by axioms, and for the moral variety, Kant formulated similar primary directives, for which he coined the term *categorical imperative*. Kant has provided a few versions of this basis, in several of his books, as his theories developed. The two most often used for such analysis are on their face quite different.

Kant's categorical imperatives	
First	The only valid moral rules, at any time, are such that can be universal moral laws
Second	Act so that you always treat yourself and other people as ends in themselves, and never as only means to an end [8]

A more direct reformulation of the first imperative states that you should only act on moral rules that you can imagine everyone else following, at any suitable time, without deriving a logical contradiction [6]. The second imperative is based on Kant's special view of humans.

While Kant's duty-based moral theories have achieved much acclaim, applying them can run into difficulties, especially when the decision-maker is faced with competing duties. Resolving the difficulty calls for ranking different duties, which is usually a fraught procedure. Kant already considered duties that can be primary and secondary, and subsequent treatments went much further than we wish to follow in the current discussion. The names of Benjamin Constant [11] and David Ross are relevant here [12].

To conclude this brief exposition, let us quote one more statement of Kant that exhibits his standards: "In law, a man is guilty when he violates the rights of another. In ethics he is guilty if he only thinks of doing so."

Character-Based Ethics Theories

The approach to ethics described as character-based is the oldest among those we consider, following the writings by Plato and Aristotle, more than two millenia ago. Similar tracks were laid in oriental traditions by Confucius and later Mencius; the latter was a contemporary of Aristotle. Another term applied to it is *virtue ethics*, since in this approach it is the acting person and his virtues, rather than the action (or the decision about it), which is at the center of attention, and accordingly neither the consequences of actions nor duties that may compel them are considered. The virtuous man is described as the person who, at the right time, does the right act, for the right reason. The "right reason" is culture appro-

priate. In pre-platonic times, for example, physical prowess was prized; in our time tolerance may be claimed to have a similar significance or in a different vein—high scholarly achievement, from which the right actions follow.

Over the last several decades, numerous ethicists have seen the consequentialist and duty-based approaches as obscuring the importance of living the moral life, acquiring moral education, with ethical and emotional family and community relationships; virtue ethics appears as a way to attend to these missing components. What sets apart this view of ethics is that it is not based on inculcating a theory, or learning principles, but on practicing the ethical behavior. It is a process of growth, with the aim for the person to get to a stage where doing the correct action, adopting honest behavior, becomes a part of his or her nature and that doing so will feel good and become a character trait.

We argue for our preference of this approach when considering decisions about speculative situations, where deontic rules may be hard to apply, and consequentialist analyses encounter numerous unknowns. Automation practice may often give rise to just such situations.

A shortcoming recognized for this approach is that the tight view of the actor as an individual does not lend it for easy use by institutions, including agencies of government. Yet such an agency is a human institution, and distinguishable people create its actions.

It must be clear that the distinction between these frameworks, or theories, is not a moral one but a matter of what a person values at a given situation. Looking at healthcare system, and at regional transportation control, arguably should lead to different types of argumentation.

34.3.2 Principles of Ethics

The theories above display how thinking about morality over time has become codified in a variety of formal ways, exhibiting deep differences, tied to cultural and personal moral values. Cultures have found other ways to preserve and display the reactions of thoughtful people to the ethical conundrums life throws at us at every turn: they have created ethical proverbs or maxims.

There is the classic Golden Rule, "do unto others as you would have them do unto you." It has been attributed to many people, going back thousands of years, and is found in the teachings of most religions. It encapsulates the ethic of reciprocity, a basic sense of justice. (This exposition would be remiss if it did not mention the demurral of George Bernard Shaw, who objected to this rule, observing that your and the others' taste or preferences may well differ.) The negative form of this adage is not as common. Hillel the Elder, a Jewish scholar living in the first century BC, is quoted in the Talmud: "That which is hateful to you, do not do to your neighbor; this is the whole law. The rest is commentary on it; go, study." Cognate forms can be found in Muslim and Confucian writings.

Societies can disagree in surprising ways. Ancient Greeks believed that revenge is sanctioned by the gods (and discharged by Nemesis). It provides a unifying basis to many of the classical tragedies of Greek theater. Hence, it was said, "to seek revenge is rational." Kant, indeed, elevated the role of reason to a main tool of ethical analysis, yet he would not have approved.

Reason may encounter further hindrances: "It is useless to attempt to reason a man out of a thing he was never reasoned into" (Jonathan Swift, 1721), in *A Letter to a Young Gentleman*…. Many have expressed this sentiment since. "As they were not reasoned up, they cannot be reasoned down" (Fisher Ames).

Here is a scattering of others; most can be seen as pertaining to one of the frameworks discussed above. Several have known first formulators:

"Any tool can be used for good or bad."
"If it is not right, do not do it; if it is not true, do not say it" (Marcus Aurelius, Meditations, second century AD).
"Honesty is the first chapter in the book of wisdom" (Thomas Jefferson).
"There cannot any one moral rule be proposed whereof a man may not justly demand a reason" (John Locke, 1690).
"Integrity has no need of rules" (Albert Camus).
"Practice what you preach" (After Dean Rusk).
"Even the most rational approach to ethics is defenseless if there isn't the will to do what is right" (Alexander Solzhenitsyn).
"Ethical axioms are found and tested not very differently from the axioms of science. Truth is what stands the test of experience " (Albert Einstein, 1950).
"No moral system can rest solely on authority" (A. J. Ayer).
"Morality is the custom of one's country, and the current feeling of one's peers" (Samuel Butler, 1900).
"The needs of a society determines its ethics" (Maya Angelou).
"There can be no final truth in ethics any more than in physics, until the last man has had his experience and has said his say" (William James, 1896).
"Science, by itself, cannot supply us with an ethic" (Bertrand Russel, 1950).

Asimov's Laws of Robotics—Consideration

In 1940 Isaac Asimov, a professor of chemistry and also a very prolific science and science-fiction writer, invented three laws designed to constrain (very advanced) robots so they operate ethically [13].

First Law	A robot may not injure a human being or, through inaction, allow a human being to come to harm
Second Law	A robot must obey the orders given it by human beings except where such orders would conflict with the First Law
Third Law	A robot must protect its own existence as long as such protection does not conflict with the First or Second Law

The laws were included in a collection of science-fiction short stories, and they appeared plausible. The stories, however, for the most part were designed to display a variety of situations which were not captured adequately by the laws, mostly due to their informal language. A subsequent collection of stories [14] continued this demonstration.

Remarkably, even though these laws were merely a literary device—included in a book considered a low-brow genre, with stories that mostly displayed modes of their failure—the laws captured wide attention, of a variety of other writers and also scholars in machine ethics [15, 16]. It soon became obvious that the laws are inadequate in many ways for the purpose intended; we referred to their informal, even sloppy language, but a deeper difficulty is their adversarial formulation. It would make a robot that adheres to them a strictly deontological-ethics device, an approach which is not satisfactory on its own, as mentioned. The two papers just referenced describe some of these inadequacies in depth. While there have been many attempts to extend or convert these laws to a working design (one of them by Asimov himself, in [14]), the consensus appears to be that no adequate extensions exist; this is probably due to the unhappy fact that we do not have currently a sufficient understanding of the possible nature(s) of advanced General Artificial Intelligence (GAI) devices. An interesting initiative to clarify the situation is due to the authors of the "2017 Montreal Declaration for a Responsible Development of AI" [26]. They produced a genuine attempt to corral the ethical issues in the development of automatic systems. The web page shows the principles only; the PDF document it points to has a thoughtful expansion of these principles. How successful is this recent attempt? It remains to be seen.

34.3.3 Automation Ethical Concerns

As explained above, ethical concerns arise when engineers design systems whose operation may lead to improper outcomes, regardless of their actual nature. Of particular interest are failure modes which should have been avoided during the system design phase, since there are now powerful design verification tools.

This could be due to any of the personnel bringing the system to operation. The ethical concerns that automation raises can be parsed as follows (some overlap of the categories is inherent):

1. Viewing the system as a software-based operation raises the issues known as those of any information processing operation [8, 17, 39]; we remind the reader that some of those are correctness of operation, privacy and safety of users and operators, data security, intellectual property integrity, maintainability, and fairness.
2. Concerns due to the physical actions associated with the functions of the automation. These include safety of personnel, integrity of the physical plant (including products), acceptable wear and tear, and conservation of input and process materials.
3. Intellectual property issues that are unrelated to the software, in particular, ensuring the validity of agreements needed to use patented or copyrighted material. Also the applications to protect patented material created in the design of the system, including devices and methods of operation.
4. Issues that arise during system design. These include beyond all those above the need to ensure that the system achieves its purposes, at the desired level of quality, subject to budgetary constraints, and satisfies subsidiary requirements, such as accessibility and physical security of the system, and satisfies relevant electrical and building codes.
5. AI & ethics—artificial intelligence gives rise to issues that are unique to it and differ from other software-based systems. It is considered in Sect. 34.3.5.

Note on Verification and Validation This topic is of such importance in the development of automated systems that omitting it, or not using its full potential, amounts to acting unethically. The same consideration prompted us to include this note. Discussing it however is beyond the mandate of this chapter. It has generated by now an enormous corpus of sources: books, software tools, journals, conference proceedings, and blogs. For reference we suggest the relevant current standard, "1012-2016—IEEE Standard for System, Software, and Hardware Verification and Validation" [29], and a collection of articles specific to an industry or technique [30].

34.3.4 Automation Failures and Their Ethical Aspects

Several automation systems with unfortunate stories are presented. Why tell such sorry tales? Rumor has it that the way

of people, engineers included, to mend their erring ways is complex, and often the cues need to be multiple and pitched right: to fit the context of a narrative, that will send a lesson home. We now show a few; indeed, they are very different from each other!

Royal Majesty Grounding, 1995

The cruise ship Royal Majesty (RM) left St. George's in Bermuda bound for Boston at 12:00 noon on June 9, 1995, and ran aground just east of the island of Nantucket at 22:25 the next day [21]. None of the over 1000 passengers were injured; repairs and lost income amounted to $7 million [18].

The RM was equipped with an automatic navigation system consisting of two main components, a GPS receiver and a Navigation and Command System (NACOS), an autopilot that controls the ship steering to the desired destination. There are a few subsidiary subsystems, such as a gyrocompass, for azimuth, Doppler log for speed, radar, Loran-C, an alternative location system, and more. However, the bridge personnel referred almost exclusively to the NACOS display. At the time of this trip, the ship has been in service for 3 years, and the crew came to trust the ship's instrumentation.

The accident has several component-reasons, as is the rule; the immediate technical problem was the rupture of the cable from the GPS antenna to its receiver that happened within an hour of departure, for a reason unknown. As a result the receiver changed to dead-reckoning mode, using compass and log. Its display shows its status in much smaller letters than the position reading, as DR, instead of the usual SOL (Satellite on Line)—none of the crew noticed it till the grounding. When the ship was equipped, the GPS and NACOS were provided by different manufacturers and used somewhat different communications standards. The following is from [18]:

> Due to these differing standards and versions, valid position data and invalid DR data sent from the GPS to the NACOS were both "labelled" with the same code (GP). The installers of the bridge equipment were not told, nor did they expect, that position data (GP-labelled) sent to the NACOS would be anything but valid position data. The designers of the NACOS expected that if invalid data were received it would have another format. Due to this misunderstanding the GPS used the same "data label" for valid and invalid data, and thus the autopilot could not distinguish between them. Since the NACOS could not detect that the GPS-provided data was invalid the ship sailed on an autopilot that was using estimated positions until a few minutes before the grounding.

That is all that was needed: an immediate cause, a broken cable, and building errors, incompatible communications protocols and a device which exhibited unusual status unobtrusively. The contribution of human nature is here: a crew, which included experienced professionals, that has been lulled to complacency, by the automation that has worked flawlessly for 3 years.

The authors of [18], who are, respectively, a researcher in maritime human factors and an academician who is very prolific about safety and automation, argue that the answer to ineffective automation is not more automation, due to human behavior, so long as humans are kept in the control loop—and while they do not say so explicitly, they do not seem to consider any alternatives viable.

Boeing 737 Max Grounding, 2019

Boeing announced the 737 Max, fourth generation of its very popular 737 series, in August 2011 and had the first 737-8 enter service in May 2017. (The 737 Max was offered in several configurations 737-7, 737-8, 737-200, 737-9, and 737-10. By December 2019 Boeing had 4932 orders and delivered 387 planes of the series [Boeing press releases].) Nearly half a million flights later, on October 29, 2018, Lion Air Flight 610 crashed, and on March 10, 2019, Ethiopian Airlines Flight 302 crashed. Not one of the 346 passengers and crew on the two flights survived. A rate of four crashes per million flights is very high for the industry, and there were consequences beyond the loss of life.

The US FAA grounded all 737 Max planes on March 13, 2019, as did similar agencies in other countries. At the time of writing (November 2020), the plane has just been re-certified, with first flight expected early next year. The monetary cost to Boeing so far is close to $20 billion.

Is there an automation-design story behind this tale, which is extreme in several ways? Indeed, there is. The 737 series was introduced in 1967: a narrow-body plane, with the attraction (for airlines) of requiring only two pilots in the cockpit. As time passed and customer needs changed, Boeing introduced more planes in the series, larger, higher capacity, with larger engines, yet maintained the "airframe type," which means that a pilot licensed to fly on one of the series can fly all the others. For the airlines, no need to train the pilots on simulator, or for re-certification test flights, means significant savings—and since Boeing was in hot competition with Airbus, the European manufacturer, an important marketing feature. The 737 Max was the largest of the series, and in particular, its engines were so large that to maintain the shape of the plane, the engines needed to be moved forward, beyond the wing, and positioned higher, so that they protruded above the wing (Figure 2 in [19]). This changed the aerodynamics of the plane compared with its predecessors in the series; when power is applied, most planes tend to raise their nose, and this was much more pronounced in the 737Max.

Autopilots in planes have been used for a long time: Sperry corporation introduced it as early as 1912 (we note that auto-piloting a plane is much simpler than achieving this in a car, on land, since the "lanes" are assigned by air traffic controllers and are non-intersecting). To have pilots feel the

plane is still a 737, Boeing introduced *stealthily* a software function, MCAS, into the autopilot computers, that would prevent the plane from increasing its angle-of-attack (AoA) too much: this can cause aerodynamic stall at low speeds. The MCAS in that case moves the tail elevators to direct the nose down; this is done with considerable force. The MCAS relied on an AoA sensor mounted on the outside of the cockpit. This is a fragile device, often damaged, and when activated in error causes the plane to descend abruptly. After the Lion Air crash, Boeing distributed to the airline instructions how pilots can recover from this state; they needed to complete the process—pulling the plug on the MCAS and the elevator motors and moving them manually to the correct position—within 4 seconds. The Ethiopian Airlines crew knew of the procedure, but apparently did not complete it in time.

Boeing could pull this off since the FAA had relinquished most of its certification oversight to the company team of Designated Engineer Representatives (DER). These experienced, veteran engineers were under pressure from the management to get the plane certified early. Not only was the MCAS barely mentioned in the operation manual, it was not mentioned at all in the documentation provided to the FAA [19].

Multiple investigations by the US Congress, Transportation Department, FBI, FAA, and NTSB (and comparable agencies in other countries) faulted Boeing on an array of issues. This is entirely unlike the ship grounding: a mix of greed, hubris, disdain of authorities, snubbing of law and professional responsibility, and callousness toward customers and passengers were combined to a revolting outcome. Automation was introduced to hide the deviation of the 737Max from the 737 airframe type. As detailed in [19], it failed because it was poorly and "optimistically" designed.

Therac-25 Linear Accelerator, 1980s

Radiation therapy is one of the common methods to treat cancer patients (the same treatment is occasionally also used to destroy non-cancerous tissue). A linear accelerator, like Therac-25, is a device that can provide it, producing a high-energy beam directed at the target tissue. The Therac-25 is a dual mode machine. Its basic operation is to create a narrow beam of electrons, tunable in the energy range 5–25MeV. The beam can be either sculpted by an array of magnets and directed to the target or be used (when at top energy) to generate a beam of X-rays, which penetrate more deeply into the patient's body. The Therac-25, manufactured by AECL of Canada in the 1980s, followed a series of earlier accelerators, Therac-6 and Therac-20 [8, §8.5]. This last version introduced several advances in the accelerator itself; yet for the purpose of our discussion, the main difference was that it was entirely software-controlled. All the interaction between operator and machine was via a keyboard-driven computer interface. The software inherited some modules that were used in the earlier models but was much enhanced.

One of the principal changes in this last stage was that several safety measures that were before implemented by hardware interlocks were replaced by software controls. This led to a simpler, smaller machine and cheaper to manufacture. Between 1985 and 1987, 4 machines, of the 11 installed then in Canada and the United States, suffered several incidents where patients were treated with overdoses of radiation. Four patients died, and several others carried radiation damage for the rest of their lives.

The report by Nancy Leveson [22] gives a detailed survey of the machine structure, the software architecture, and these incidents, as well as of the resulting interactions of AECL with the FDA and similar Canadian agencies. Careful analyses revealed that previous safety evaluations ignored the software entirely! They assumed it would operate as expected. When the software was put to thorough analysis, a plethora of defects was revealed; some of them were found to have been inherited from Therac-20, where they caused no recorded damage, presumably due to the hardware preventive measures it carried. Some software errors were minor mistakes, never noticed before. A notable one was a 1-byte flag that was tested before the beam was turned on and had to be zero for the operation to continue and the beam to be activated. That flag was *incremented* by one whenever the readiness tests failed. Since the tests were run continually, while the machine reacted to commands that required physical arrangement of components—an activity that took up to 8 seconds—and while the computer in question, a DEC PDP 11, was slow, it still performed the testing that the machine is set up correctly hundreds of times during such a hardware reconfiguration. The 1-byte flag cycled through 0 every 256 tests; if at that instant its status was queried, a go-ahead would result, regardless of the actual situation.

While this flaw is trivial to repair (just set the flag to one, when the status was not ready—as AECL did, later), it is a symptom of a different and deeper problem: poor comprehension of the software operations. The basic software activity consisted of a cycle of tasks, re-initiated by a clock-interrupt every 100 milliseconds. Several of those tasks used shared variables to coordinate activities and modified them as needed. In cases where the tasks did not complete within the renewal interval (0.1sec), they were interrupted, occasionally left in inconsistent state, and restarted. It is impossible to predict the erroneous results that may then occur….

In summer 1985 the FDA had AECL recall the machine and introduced some modifications, but lethal incidents continued. In February 1987 the FDA had AECL recall the machine and stop its usage. Several iterations between the FDA and AECL during that year resulted in several changes

to the software and installation of hardware enhancements. No further accidents ensued.

Some of the software errors that were revealed are embarrassing, when viewed with modern understanding of real-time, multi-threaded software. AECL did not use any of the operating systems available at the time, but rolled its own [22]. May we assume such mistakes would not occur in current software? Probably not. Unfortunately, the half-life of basic software errors is near infinity.

Yet times have changed. A significant effort has been expended on software development environments and methodologies and later on tools for software verification and validation. While in systems as complex as the one we now discussed, involving hardware of a variety of types, software components prepared separately, and user operations, this will always be a scene that involved art and mathematics, much improvement has been achieved.

The publicly available information about the Therac-25 debacle shows AECL as a negligent, callous, unresponsive, and quite irresponsible company [22]. No personal information is known to identify the internal processes that led to this turn of events. The company is now out of the business of medical devices.

The Machine Stops [20]

Here we encounter a different kettle of fish. It is an imagined automatic system; implied though it is quite vividly portrayed, yet it is fictional, brought to us as a short story, by the master story-teller Edward M. Forster, who published it in 1910, likely influenced by interactions with his friend H.G. Wells, and conversation about the tale the latter wrote "The Time Machine," published in 1895, and described in it the insufferable morlocks, living underground.

In Forster's tale the entire world population lives in warrens underground. All their needs are provided by a system called "the Machine." When the story begins, we are led to understand this has been the state of affairs for a long time; durations are vague, but the people take the machine for granted, and many, including the story protagonist, begin to regard it as deity. The surface of the earth is abandoned, considered deadly. Communications technology is presented as we know it in the twenty-first century, with Internet capabilities, all mediated by the Machine. Although Forster witnessed none of that, he was in company with people who could so fantasize already. A very effective transportation system covers the earth, but it is viewed as a relic and only lightly used; whatever a person needs, it is brought to her individual room. Forster adds a large number of hints at the ways of the society, which suggest much societal engineering prepared the population as the Machine was developed—our focus all along is the Machine. A "Committee of the Machine" exists, also called "Central Committee;" the two may be the same—it is mentioned as a body that enforces and rarely changes policies.

Forster mentions in passing some measures, presumably put in place by the Central Committee, that we would find intolerable. Space is assigned, rather than selected; some form of eugenics maintained; freedom of movement limited. Values could change too, he suggests, and describe how originality is deprecated.

Maintenance of the Machine is mentioned as a set of rote tasks that were performed, as time went on, with less and less understanding. It is said no one really knew much about its operation and stated that nobody understood it as a whole. The machine was built with a "mending apparatus," which also protected it from attacks. The place of the Machine in the world view of the population was such that the phrase "the Machine stops" was meaningless to most, when used by one of its denizens. Yet as the Machine started failing, interruptions during the streaming of music, or bad food delivered, the people learned to live with it and expected the machine to heal itself. The deterioration is relatively quick, and the entire system crumbles, as various operations fail.

The story was an impressive tour de force by the author at the time, yet we can see it as an account of the demise of a society who trusted its engineers, but did not understand the nature of engineered systems, the need for maintenance—not just of the machinery but also of the connection and interaction, between the machine and the population—so that a dynamic develops, in which the routine operation of the Machine depends on human intervention. It is a riveting exercise for the reader to imagine what circumstances forced that society underground and made them choose such a Machine as their solution, an improbable duress? Possibly global warming getting out of hand… ? And why were the interfaces needed to keep the system alive so poorly designed? (Naturally, because Forster was not an expert in systems engineering and none was available for him to consult.)

Would our current society have done better if such extreme pressure were forced on us? With technology much evolved beyond what was available, or even imaginable in 1910, could our solution be similarly superior to that Machine and our engineering skills up to it? Designing for long-term survival is unlike any challenges humans have faced… !

34.3.5 Artificial Intelligence and Its Ethical Aspects

> As the internet and increased computing power have facilitated the accumulation and analysis of vast data, unprecedented vistas for human understanding have emerged …. [We may have] generated a potentially dominating technology in search of a guiding philosophy.
>
> —Henry A. Kissinger: *How the Enlightenment Ends,* 2018, [27].

Artificial intelligence (AI) was declared in 1956 as the "next big field" in computer science and kept exactly this designation for better than half a century. The field was not idle: multiple techniques were developed, such as *genetic algorithms* and *Bayesian networks* or *simulated annealing*; various vogues had their day, such as enormous *world ontologies* or *expert systems*, but successes were narrow and far between. The situation started changing in the first decade of the twenty-first century, with the growth and then explosion of *machine learning*. The growth was facilitated by the significant, continuing decline in costs of computer power and data storage space, on one hand, and by the enormous growth of data available on the Internet, from blogs to social networks to databases of many government agencies and corporations who conduct their business in cyberspace, on the other hand. AI continued to employ a variety of techniques, but soon machine learning (ML), based on (artificial) neural networks, became the much-preferred paradigm. When Kissinger, an experienced historian and statesman, was perchance exposed to these facts, followed by dialogues with technically aware colleagues, he was moved to write [27], where the above instructive quotation is found. What the (neural) network learns may be seen as the ability to classify items—which can be, for example, strings of characters or images—and decide whether a given item belongs to a specified class; with much ingenuity this capability has been leveraged to any number of applications, from driving a car to recognizing faces in a crowd.

While most of the ethical attention raised by AI is similar to that which any software modality entails, AI, especially its recent developments—machine learning and deep learning—merits a special note, as more and more areas in our life and affairs and the automatic tools we build are impacted by it. In this it is not different from other theory-rich areas of computer science, such as database management or compilation—but unlike those, little in the theory of AI is currently settled. The recent survey [25] provides detailed summary and a wealth of references. In Sect. 34.3.2 we presented the three "laws" of robotics invented by Isaac Asimov and referred briefly to their inadequacy to prevent a robot from making wrong moves. The 2017 Montreal declaration for a responsible development of artificial intelligence [26] can be seen as a careful, disciplined "call to arms," for this difficult task. It replaces those three laws by ten principles, which are further developed into several directives each.

Opacity of Machine Learning AI

There is a special property of AI systems developed by training a neural network; even as such systems are shown to be very proficient, in a remarkable range of fields and applications, where they provide a wealth of good or useful decisions—they cannot provide an explanation of how, and why, any of their decisions was reached, and the system owners (its users), as well as its developers, cannot do so either—such systems function as black boxes. The concept of a subsystem which is a *black box* is commonplace, dating to middle of the twentieth century; the traditional black box refers to a mere transfer function, which can be used with no need to know its operational details. It is however coupled with the understanding that if the need for the details arises, the box can be opened and inspected and any information about its internals be available. That is not the case with the neural network machine: it is an opaque classifier of clumps of data—a classification which may be put to further use, in a following part of an algorithm—but it is not responsive to any attempt to probe it for enlightenment about its *modus operandi*, which includes its successes, as well as the failures. If it is wished, its internal structure and the weights associated with each node and other elements can be listed, but this sheds no light or provides a meaningful explanation. This was not the case with previous generations of AI, which can be described as rule-based; their operations can be explained, at any required depth; but alas, they were not as capable!

This property is alarming for three reasons beyond the chill it casts (nobody likes to be thwarted in this way):

(a) **Brittleness:** Even an excellent ML classifier, is reliable on a part of its possible input space only, a part which is effectively impossible to determine there is no guarantee the system responds correctly, in any particular case, and in fact such systems are notorious for being occasionally derailed, by trivial features in their input [31]. Researchers have found many ways of tricking such systems, by changing subtly the system input—in a manner that is unobtrusive to humans—yet entirely confused the system responses. It is considered open to hackers to "game" and twiddle the systems at will. Since there is no understanding, there is no defense.

(b) **Bias:** The selection of data used to train the system affects, naturally, the way different characteristics of the input get translated to outcomes. A voice recognition system can be more likely to misunderstand speech in a regional accent it was not trained with, and thereby not provide the desired service or help. A loan-application scanner program in a bank is likely to mishandle an application of a person with unusual life trajectory. Just as the AI cannot be asked to verify its response, the opacity means the impossibility of critiquing or appealing effectively the system determinations. This type of AI systems especially the extremely large "deep learning" kind, which is discussed further below. Different system architectures, training regimes, and approximation criteria are attempted. Some observers of the scene are confident the opacity can be dissolved in some manner,

maybe by added complexity; others see this AI approach as doomed, due to its being "Brittle, Greedy, Opaque, and Shallow," and their progress likely to hit a wall any time soon. Even our vocabulary for cognitive processes does not fit well such AI systems. One result is the difficulty of pursuing any ethical analysis of such systems—they are opaque! Is there hope that rule-based AI can be combined with the neural network-based recognizer, making its responses explicable? The difficulty seems similar to comprehending the deep learning AI directly. It is an open question—which is pursued with great vigor.

(c) **Errors in input:** Only recently have researchers turned to examine the quality of the labeling in several of the most popular labeled datasets used to train *and test* such networks [33]; the results were surprising. They found that labeling errors averaged 3.4% of the data set items. They also found that the impact on the performance over test-sets was more nuanced and depended on the model used and the size of the training set.

A word after a word after a word, is power
—Margaret Atwood.

Recently a particular type of large Deep Learning model was introduced by several companies, generally called Large Language Model. The most famous, probably since it came first, in 2020, is GPT-3, the third in a sequence of *Generative Pre-trained Transformers*, created by OpenAI in San Francisco. Its native purpose is to generate text when given a prompt, producing one token (a word, usually) at a time, recursively. For its input the creators collected texts amounting to 175 billion tokens (by skimming the web, and adding digitized books); a measure of its complexity is indicated by the size of its context, its "status descriptor," that determines the generated output: it is a 2048-token-long string.

The generated text is usually of a high, human-created quality. While clearly 'there is no there there,' in such a text beyond the initial prompt (which can be as short as a word), people have reported that they enjoyed reading the output, finding it of substance, interest, etc. **Note:** Because this type of automata is currently under feverish development, the main reference recommended is the Wikipedia article "GPT-3." It is likely to be kept current, and lists many of the original sources.

This model has already found numerous uses, leading to serious questions, such as about the (im)possibility to trust the authenticity of the claimed authorship of submitted written work, such as essays, in any imaginable context. A recent article describes getting GPT-3 to write a scientific paper about itself, [41], with minimal tweaks by humans. GPT-3 ends the paper (which as of this writing, September 2022, is still in review) with "Overall, we believe that the benefits of letting GPT-3 write about itself outweigh the risks. However, we recommend that any such writing be closely monitored by researchers in order to mitigate any potential negative consequences."

One of the developments based on GPT-3 was DALL-E, now followed by DALL-E 2. The tokens these programs generate are pixels rather than words. They create images instead of text when prompted by verbal cues. They can draw and paint in any style which was digitally available when their input was collected, and do it very effectively; they can also produce photo-realistic images.

The ethical implications of unleashing such systems are of a different order than anything we have witnessed or described before, for two main reasons, and both pertain to issues of automation:

(a) These tools automate activities hitherto considered the domain of high-order human creativity as suggested by the quotation of Margaret Atwood. Their release was received with howls of protest by artists and others who are affronted by the perceived loss of human exceptionality. These denunciations were quite similar to the reactions two centuries earlier, when the invention of photography alarmed painters and art lovers. Interestingly, that invention drove painters to develop, and art-lovers to appreciate, a large number of new styles, genres, materials, and techniques, as ways to distinguish their creations from the "mechanical" ones.

(b) The purpose of the designer of any automation tool has been to create a system with certain properties. The so-called "world models" described here were designed to have specific capabilities — generate scads of tokens, emulating the ways they are seen to be used by humans. The properties of the tools, or rather, their output, were not designed for: they are *emergent*, not yet known, and the task of the engineer is then to discover them, rather than verify the existence of any target behavior. The properties then need to be evaluated; the questions to be asked are about the potential of such output, possibly in large quantity with such properties, to benefit or harm society. Such analyses are analogous to the work of biologists who investigate the nature of novel creatures caught in the wild, or sometimes grown in the lab by a process akin to genetic engineering.

From the point of view of public platforms (social media, open blogs) such systems would be seen as content-creators, on par with humans. Users of the media can even have automata of this type post their output without prior inspection. A question of ethical, and possibly legal, nature that needs to be settled is whether to require such postings to be marked as of other-than-human source?

34.4 Protocols for Ethical Analysis

> You're not going to find an exceptionless rule …Sometimes there isn't an answer in the moral domain, and sometimes we have to agree to disagree, and come together and arrive at a good solution about what we will live with.
>
> —Patricia Churchland, [32].

The knowledge of ethical principles, coupled with the understanding of ethical theories and their relation to our society, does not yet mean that whenever we need to make a decision that has an ethical dimension, it is immediately available. The application of this knowledge to a specific case at hand requires what has been called *ethical protocols* or more fully—as the section is named. There is a very large literature about these protocols; see, for example [34, 35].

As Maner says in [34], numerous protocols have been prepared, reflecting the different domains which were the areas of concern of the protocol designers. We show now a skeletal protocol and defer to Appendix B a much more detailed one, adapted form [36]. Each is seen as a sequence of questions or tasks:

1. What is the question? What needs to be decided?
2. Who are the stakeholders?
3. What are the conflicting issues?
4. Which values are involved?
5. How are stakeholders, issues, and values related?
6. What (effective) actions exist? Which of them are available to us?
7. Are the following ethical tests satisfied by the actions we consider:
   (a) Does it conflict with accepted principles? (Ten commandments, criminal law …)
   (b) Is it in agreement with the Golden Rule?
   (c) Does it agree with the Rule of Universality (what if everyone acted this way?)
   (d) Does it agree with the Rule of Consistency (what if we always acted this way?)
   (e) Does it agree with the Rules of Disclosure (what if our action is known to everybody?)
   (f) Does it satisfy the Rule of Best Outcome?

Question 7 invokes nearly all decision aids we saw in this chapter.

34.5 Codes of Ethics

Nearly all professional societies create and advertise a code of ethics adapted to the profession in question (other terms often used are *code of behavior* and *code of conduct*). There are several roles, or functions of such a code, internal and external to the professional society:

- Inspire the members to professionalism
- Educate about the profession (both the members and society)
- Guide professional actions
- Inform about accountability (both the members and society)
- Provide enforcement information (for the licensed professions)

The code needs to be read by the professionals as they find their place in the profession. Appendix A exhibits the code of ethics for software engineers. It was prepared by a joint committee of the ACM and the Computer Society of the IEEE, the two major organizations for computing professionals in the United States. A survey and an interesting evaluation of the impact of the code of ethics of the ACM, which is similar to the Software Engineering code in Appendix A, are at [37]. Most engineering societies adapted for their needs the *Code of Conduct* of the National Academy of Engineering, available at [38].

The code of conduct of IFAC, *International Federation of Automatic Control*, also provided in Appendix A, is starkly different, since IFAC has no individuals as members, but national member organizations; also, other control-oriented organizations can be affiliates, and those organizations are the immediate audience of the code.

34.6 Online Resources for Ethics of Automation

> If it is not on the 'net, it does not exist.
>
> —Street scuttlebutt.

The street needs not be taken too literally, yet the claim expresses much that is true. In addition to the "specialized" sites in the list below, each of the professional societies that touches on automation has parts of its website which deal with professional ethics: ACM, IEEE, ASEE, IFAC, and the arching societies AAAS and AAES. Online is where we shall find which is new, and in the field of automation, "new" is a key word. The list below is a rich one, but while they all are active at the time of writing, many (most?) web sites are short-lived species; you may need to look for more. Use links in those sources, to delve farther and deeper, using your favorite web search engine. Finally, the Stanford Encyclopedia of Philosophy, SEP, is impressive in its selection and

quality of ethics-related entries. The Internet Encyclopedia of Philosophy, IEP, has a more practical mien and is highly recommended as well.

http://www.bsa.org	Global software industry advocate
http://catless.ncl.ac.uk/risks	The Risks Digest; an ACM moderated forum
http://www.cerias.purdue.edu	Center for Information Security at Purdue Univ
http://www.dhs.gov/dhspublic	Bulletin board of the DHS
http://ethics.iit.edu	Center for research in professional ethics
http://privacyrights.org	A privacy rights clearing house
https://plato.stanford.edu/	Stanford Encyclopedia of Philosophy
https://iep.utm.edu/	Internet Encyclopedia of Philosophy
https://esc.umich.edu/	Center for ethics, society, and computing at the Univ. of Michigan
https://cssh.northeastern.edu/informationethics/	Multiple centers for computing ethics reseach at Northeastern Univ
https://www.nationalacademies.org/our-work/responsible-computing-research-ethics-and-governance-of-computing-research-and-its-applications	Responsible Computing research at the national academies
https://ocean.sagepub.com/blog/10-organizations-leading-the-way-in-ethical-ai	A roster of organizations for ethical AI research

34.7 Sources for Automation and Ethics

Comments About Sources for This Chapter

The bibliography lists as usual the specific publications which have been used in preparing this chapter, in order of citation. We wish to draw attention to three sources. One is available online, *The Stanford Encyclopedia of Philosophy*; some of its articles recur in the bibliography. The articles are by recognized authorities and are revised for currency every several (5 to 10) years. Its editors take a remarkably expansive view of their domain, possibly of eighteenth-century vintage, and include many articles relevant to the questions we consider, for example, [28].

Two other sources are print books [39,40] and anthologies of wide-ranging articles, many of which deal with topics that are important to the interactions of ethics with automation. They are not new, appearing in 1995 and 2011, respectively, yet remain very much of interest.

Appendix A: Code of Ethics Examples

Software Engineering Code of Ethics and Professional Practice

This code is maintained on the site of the ACM at SE CODE (https://ethics.acm.org/code-of-ethics/software-engineering-code/).

Short Version

PREAMBLE

The short version of the code summarizes aspirations at a high level of abstraction. The clauses that are included in the full version give examples and details of how these aspirations change the way we act as software engineering professionals. Without the aspirations, the details can become legalistic and tedious; without the details, the aspirations can become high sounding but empty; together, the aspirations and the details form a cohesive code.

Software engineers shall commit themselves to making the analysis, specification, design, development, testing, and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety, and welfare of the public, software engineers shall adhere to the following Eight Principles:

1. Public
   Software engineers shall act consistently with the public interest.
2. Client and employer
   Software engineers shall act in a manner that is in the best interests of their client and employer, consistent with the public interest.
3. Product
   Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
4. Judgment
   Software engineers shall maintain integrity and independence in their professional judgment.
5. Management
   Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6. Profession
   Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7. Colleagues
   Software engineers shall be fair to and supportive of their colleagues.
8. Self

Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Full Version

PREAMBLE

Computers have a central and growing role in commerce, industry, government, medicine, education, entertainment, and society at large. Software engineers are those who contribute by direct participation or by teaching, to the analysis, specification, design, development, certification, maintenance, and testing of software systems. Because of their roles in developing software systems, software engineers have significant opportunities to do good or cause harm, to enable others to do good or cause harm, or to influence others to do good or cause harm. To ensure, as much as possible, that their efforts will be used for good, software engineers must commit themselves to making software engineering a beneficial and respected profession. In accordance with that commitment, software engineers shall adhere to the following Code of Ethics and Professional Practice.

The Code contains eight Principles related to the behavior of and decisions made by professional software engineers, including practitioners, educators, managers, supervisors, and policy-makers, as well as trainees and students of the profession. The Principles identify the ethically responsible relationships in which individuals, groups, and organizations participate and the primary obligations within these relationships. The Clauses of each Principle are illustrations of some of the obligations included in these relationships. These obligations are founded in the software engineer's humanity, in special care owed to people affected by the work of software engineers and in the unique elements of the practice of software engineering. The Code prescribes these as obligations of anyone claiming to be or aspiring to be a software engineer.

It is not intended that the individual parts of the Code be used in isolation to justify errors of omission or commission. The list of Principles and Clauses is not exhaustive. The Clauses should not be read as separating the acceptable from the unacceptable in professional conduct in all practical situations. The Code is not a simple ethical algorithm that generates ethical decisions. In some situations, standards may be in tension with each other or with standards from other sources. These situations require the software engineer to use ethical judgment to act in a manner which is most consistent with the spirit of the Code of Ethics and Professional Practice, given the circumstances.

Ethical tensions can best be addressed by thoughtful consideration of fundamental principles, rather than blind reliance on detailed regulations. These Principles should influence software engineers to consider broadly who is affected by their work; to examine if they and their colleagues are treating other human beings with due respect; to consider how the public, if reasonably well informed, would view their decisions; to analyze how the least empowered will be affected by their decisions; and to consider whether their acts would be judged worthy of the ideal professional working as a software engineer. In all these judgments, concern for the health, safety, and welfare of the public is primary; that is, the "Public Interest" is central to this Code.

The dynamic and demanding context of software engineering requires a code that is adaptable and relevant to new situations as they occur. However, even in this generality, the Code provides support for software engineers and managers of software engineers who need to take positive action in a specific case by documenting the ethical stance of the profession. The Code provides an ethical foundation to which individuals within teams and the team as a whole can appeal. The Code helps to define those actions that are ethically improper to request of a software engineer or teams of software engineers.

The Code is not simply for adjudicating the nature of questionable acts; it also has an important educational function. As this Code expresses the consensus of the profession on ethical issues, it is a means to educate both the public and aspiring professionals about the ethical obligations of all software engineers.

Principles

Principle 1: Public

Software engineers shall act consistently with the public interest. In particular, software engineers shall, as appropriate:

1.01. Accept full responsibility for their own work.
1.02. Moderate the interests of the software engineer, the employer, the client, and the users with the public good.
1.03. Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy, or harm the environment. The ultimate effect of the work should be to the public good.
1.04. Disclose to appropriate persons or authorities any actual or potential danger to the user, the public, or the environment, that they reasonably believe to be associated with software or related documents.
1.05. Cooperate in efforts to address matters of grave public concern caused by software, its installation, maintenance, support, or documentation.
1.06. Be fair and avoid deception in all statements, particularly public ones, concerning software or related documents, methods, and tools.

1.07. Consider issues of physical disabilities, allocation of resources, economic disadvantage, and other factors that can diminish access to the benefits of software.
1.08. Be encouraged to volunteer professional skills to good causes and to contribute to public education concerning the discipline.

Principle 2: Client and employer
Software engineers shall act in a manner that is in the best interests of their client and employer, consistent with the public interest. In particular, software engineers shall, as appropriate:

2.01. Provide service in their areas of competence, being honest and forthright about any limitations of their experience and education.
2.02. Not knowingly use software that is obtained or retained either illegally or unethically.
2.03. Use the property of a client or employer only in ways properly authorized and with the client's or employer's knowledge and consent.
2.04. Ensure that any document upon which they rely has been approved, when required, by someone authorized to approve it.
2.05. Keep private any confidential information gained in their professional work, where such confidentiality is consistent with the public interest and consistent with the law.
2.06. Identify, document, collect evidence, and report to the client or the employer promptly if, in their opinion, a project is likely to fail, to prove too expensive, to violate intellectual property law, or otherwise to be problematic.
2.07. Identify, document, and report significant issues of social concern, of which they are aware, in software or related documents, to the employer or the client.
2.08. Accept no outside work detrimental to the work they perform for their primary employer.
2.09. Promote no interest adverse to their employer or client, unless a higher ethical concern is being compromised; in that case, inform the employer or another appropriate authority of the ethical concern.

Principle 3: Product
Software engineers shall ensure that their products and related modifications meet the highest professional standards possible. In particular, software engineers shall, as appropriate:

3.01. Strive for high quality, acceptable cost, and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by the employer and the client and are available for consideration by the user and the public.
3.02. Ensure proper and achievable goals and objectives for any project on which they work or propose.
3.03. Identify, define, and address ethical, economic, cultural, legal, and environmental issues related to work projects.
3.04. Ensure that they are qualified for any project on which they work or propose to work, by an appropriate combination of education, training, and experience.
3.05. Ensure that an appropriate method is used for any project on which they work or propose to work.
3.06. Work to follow professional standards, when available, that are most appropriate for the task at hand, departing from these only when ethically or technically justified.
3.07. Strive to fully understand the specifications for software on which they work.
3.08. Ensure that specifications for software on which they work have been well documented, satisfy the users' requirements, and have the appropriate approvals.
3.09. Ensure realistic quantitative estimates of cost, scheduling, personnel, quality, and outcomes on any project on which they work or propose to work and provide an uncertainty assessment of these estimates.
3.10. Ensure adequate testing, debugging, and review of software and related documents on which they work.
3.11. Ensure adequate documentation, including significant problems discovered and solutions adopted, for any project on which they work.
3.12. Work to develop software and related documents that respect the privacy of those who will be affected by that software.
3.13. Be careful to use only accurate data derived by ethical and lawful means and use it only in ways properly authorized.
3.14. Maintain the integrity of data, being sensitive to outdated or flawed occurrences.
3.15. Treat all forms of software maintenance with the same professionalism as new development.

Principle 4: Judgment
Software engineers shall maintain integrity and independence in their professional judgment. In particular, software engineers shall, as appropriate:

4.01. Temper all technical judgments by the need to support and maintain human values.
4.02. Only endorse documents either prepared under their supervision or within their areas of competence and with which they are in agreement.
4.03. Maintain professional objectivity with respect to any software or related documents they are asked to evaluate.

4.04. Not engage in deceptive financial practices such as bribery, double billing, or other improper financial practices.
4.05. Disclose to all concerned parties those conflicts of interest that cannot reasonably be avoided or escaped.
4.06. Refuse to participate, as members or advisors, in a private, governmental, or professional body concerned with software-related issues, in which they, their employers, or their clients have undisclosed potential conflicts of interest.

Principle 5: Management
Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance. In particular, those managing or leading software engineers shall, as appropriate:

5.01. Ensure good management for any project on which they work, including effective procedures for promotion of quality and reduction of risk.
5.02. Ensure that software engineers are informed of standards before being held to them.
5.03. Ensure that software engineers know the employer's policies and procedures for protecting passwords, files, and information that is confidential to the employer or confidential to others.
5.04. Assign work only after taking into account appropriate contributions of education and experience tempered with a desire to further that education and experience.
5.05. Ensure realistic quantitative estimates of cost, scheduling, personnel, quality, and outcomes on any project on which they work or propose to work, and provide an uncertainty assessment of these estimates.
5.06. Attract potential software engineers only by full and accurate description of the conditions of employment.
5.07. Offer fair and just remuneration.
5.08. Not unjustly prevent someone from taking a position for which that person is suitably qualified.
5.09. Ensure that there is a fair agreement concerning ownership of any software, processes, research, writing, or other intellectual property to which a software engineer has contributed.
5.10. Provide for due process in hearing charges of violation of an employer's policy or of this Code.
5.11. Not ask a software engineer to do anything inconsistent with this Code.
5.12. Not punish anyone for expressing ethical concerns about a project.

Principle 6: Profession
Software engineers shall advance the integrity and reputation of the profession consistent with the public interest. In particular, software engineers shall, as appropriate:

6.01. Help develop an organizational environment favorable to acting ethically.
6.02. Promote public knowledge of software engineering.
6.03. Extend software engineering knowledge by appropriate participation in professional organizations, meetings, and publications.
6.04. Support, as members of a profession, other software engineers striving to follow this Code.
6.05. Not promote their own interest at the expense of the profession, client, or employer.
6.06. Obey all laws governing their work, unless, in exceptional circumstances; such compliance is inconsistent with the public interest.
6.07. Be accurate in stating the characteristics of software on which they work, avoiding not only false claims but also claims that might reasonably be supposed to be speculative, vacuous, deceptive, misleading, or doubtful.
6.08. Take responsibility for detecting, correcting, and reporting errors in software and associated documents on which they work.
6.09. Ensure that clients, employers, and supervisors know of the software engineer's commitment to this Code of ethics and the subsequent ramifications of such commitment.
6.10. Avoid associations with businesses and organizations which are in conflict with this Code.
6.11. Recognize that violations of this Code are inconsistent with being a professional software engineer.
6.12. Express concerns to the people involved when significant violations of this Code are detected unless this is impossible, counter-productive, or dangerous.
6.13. Report significant violations of this Code to appropriate authorities when it is clear that consultation with people involved in these significant violations is impossible, counter-productive, or dangerous.

Principle 7: Colleagues
Software engineers shall be fair to and supportive of their colleagues. In particular, software engineers shall, as appropriate:

7.01. Encourage colleagues to adhere to this Code.
7.02. Assist colleagues in professional development.
7.03. Credit fully the work of others and refrain from taking undue credit.
7.04. Review the work of others in an objective, candid, and properly documented way.

7.05. Give a fair hearing to the opinions, concerns, or complaints of a colleague.
7.06. Assist colleagues in being fully aware of current standard work practices including policies and procedures for protecting passwords, files and other confidential information, and security measures in general.
7.07. Not unfairly intervene in the career of any colleague; however, concern for the employer, the client, or public interest may compel software engineers, in good faith, to question the competence of a colleague.
7.08. In situations outside of their own areas of competence, call upon the opinions of other professionals who have competence in that area.

Principle 8: Self
Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession. In particular, software engineers shall continually endeavor to:

8.01. Further their knowledge of developments in the analysis, specification, design, development, maintenance, and testing of software and related documents, together with the management of the development process.
8.02. Improve their ability to create safe, reliable, and useful quality software at reasonable cost and within a reasonable time.
8.03. Improve their ability to produce accurate, informative, and well-written documentation.
8.04. Improve their understanding of the software and related documents on which they work and of the environment in which they will be used.
8.05. Improve their knowledge of relevant standards and the law governing the software and related documents on which they work.
8.06. Improve their knowledge of this Code, its interpretation, and its application to their work.
8.07. Not give unfair treatment to anyone because of any irrelevant prejudices.
8.08. Not influence others to undertake any action that involves a breach of this Code.
8.09. Recognize that personal violations of this Code are inconsistent with being a professional software engineer.

This Code was developed by the IEEE-CS/ACM joint task force on Software Engineering Ethics and Professional Practices (SEEPP):
Executive Committee: Donald Gotterbarn (Chair), Keith Miller, and Simon Rogerson
Members: Steve Barber, Peter Barnes, Ilene Burnstein, Michael Davis, Amr El-Kadi, N. Ben Fairweather, Milton Fulghum, N. Jayaram, Tom Jewett, Mark Kanko, Ernie Kallman, Duncan Langford, Joyce Currie Little, Ed Mechler, Manuel J. Norman, Douglas Phillips, Peter Ron Prinzivalli, Patrick Sullivan, John Weckert, Vivian Weil, S. Weisband, and Laurie Honour Werth

International Federation of Automatic Control—Code of Conduct

IFAC recognizes its role as a worldwide federation for promoting automatic control for the benefit of humankind. In agreement with and in implementation of the approved IFAC—Mission and Vision—this document summarizes the commitment and obligation of IFAC to maintain ethical and professional standards in its academic and industrial activities. All activities within IFAC as well as volunteers acting on behalf of or for IFAC are to act in accordance with this Code of Conduct.

1. Honesty and Integrity
Activities conducted by IFAC shall always be fair, honest, transparent, and in accordance with the IFAC—Mission and Vision. That is, their main goal is to contribute to the promotion of the science and technology of control in the broadest sense. IFAC disapproves any actions which are in conflict with existing laws, are motivated by criminal intentions, or include scientifically dishonest practices such as plagiarism, infringement, or falsification of results. IFAC will not only retaliate against any person who reports violations of this principle but rather encourage such reporting.

2. Excellence and Relevance
IFAC recognizes its responsibility to promote the science and technology of automatic control through technical meetings, publications, and other means consistent with the goals and values defined in the IFAC—Mission and Vision. Further, IFAC has the responsibility to be a trusted source of publication material on automatic control renowned for its technical excellence. IFAC acknowledges its professional obligation toward employees, volunteers, cooperating or member organizations and companies, and further partners.

3. Sustainability
A major challenge in future automatic control is the development of modern techniques which reduce the ecological damage caused by technology to a minimum. IFAC acknowledges this fact and contributes to a solution by promoting the importance of automatic control and its impact on the society and by advancing the knowledge in automatic control and

its applications. IFAC disapproves any actions which are in conflict with the above philosophy, in particular those which have a negative impact on the environment.

4. Diversity and Inclusivity
IFAC is a diverse, global organization with the goal to create a fruitful environment for people from different cultures dealing with automatic control in theory and practical applications. People shall be treated fairly, respectfully, and their human rights shall be protected. IFAC is committed to the highest principles of equality, diversity, and inclusion without boundaries. IFAC disapproves of any harassment, bullying, or discrimination.

5. Compliance of Laws
The purpose of any action conducted by IFAC is to further the goals defined in IFAC's constitution and consequences thereof. Activities on behalf of IFAC cannot be in conflict with ethical principles or laws existing in countries where IFAC operates. This includes but is not limited to any form of bribery, corruption, or fraud. IFAC disapproves unethical or illegal business practices which restrain competition such as price fixing or other kinds of market manipulation. Conflicts of interest are to be prevented if possible and revealed immediately whenever they occur. IFAC assures the protection of confidential information belonging to its member organizations and further partners.

Appendix B: Steps of the Ethical Decision-Making

The following has been closely adapted from [36].

While the process is presented here as a sequence of actions, in practice the decision-maker may have to return to an earlier stage and fill up details the need for was revealed later:

1. Gather all relevant facts.
   - Don't jump to conclusions without the facts.
   - Questions to ask: Who, what, where, when, how, and why. However, facts may be difficult to find because of the uncertainty often found around ethical issues.
   - Some facts are not available.
   - Assemble as many facts as possible before proceeding.
   - Clarify what assumptions you are making!
2. Define the ethical issue(s)
   - Don't jump to solutions without first identifying the ethical issue(s) in the situation.
   - Define the ethical basis for the issue you want to focus on.
   - There may be multiple ethical issues—focus on one major one at a time.
3. Identify the affected parties.
   - Identify all stakeholders, and then determine:
   - Who are the primary or direct stakeholders?
   - Who are the secondary or indirect stakeholders?
   - Why are they stakeholders for the issue?
   - Perspective-taking—try to see the situation through the eyes of those affected; interview them if possible.
4. Identify the consequences of possible actions.
   - Think about potential positive and negative consequences for affected parties by the decision. Focus on primary stakeholders initially.
   - Estimate the magnitude of the consequences and the probability that the consequences will happen.
   - Short-term vs. long-term consequences—will decision be valid over time.
   - Broader systemic consequences—tied to symbolic and secrecy as follows
   - Symbolic consequences—each decision sends a message.
   - Secrecy consequences—what are the consequences if the decision or action becomes public?
   - Did you consider relevant cognitive barriers/biases?
   - Consider what your decision would be based only on consequences—then move on and see if it is similar given other considerations.
1. Identify the relevant principles, rights, and justice issues.
   - Obligations should be thought of in terms of principles and rights involved:
     (A) What obligations are created because of particular ethical principles you might use in the situation? Examples: Do no harm; do unto others as you would have them do unto you; do what you would have anyone in your position do in the given context.
     (B) What obligations are created because of the specific rights of the stakeholders? What rights are more basic vs. secondary in nature? Which help protect an individual's basic autonomy? What types of rights are involved—negative or positive?
     (C) What concepts of justice (fairness) are relevant—distributive or procedural justice?
   - Did you consider any relevant cognitive barriers/biases? Formulate the appropriate decision or action based solely on the above analysis of these obligations.
2. Consider your character and integrity
   - Consider what your relevant community members would consider to be the kind of decision that an individual of integrity would make in this situation.
   - What specific virtues are relevant in the situation?

- Disclosure rule—what would you do if various media channels reported your action and everyone was to read it.
- Think about how your decision will be remembered when you are gone.
- Did you consider any relevant cognitive biases/barriers?
- What decision would you come to based solely on character considerations?

3. Think creatively about potential actions.
   - Be sure you have not been unnecessarily forced into a corner.
   - You may have some choices or alternatives that have not been considered.
   - If you have come up with solutions "a" and "b," try to brainstorm, and come up with a "c" solution that might satisfy the interests of the primary parties involved in the situation.
4. Check your gut.
   - Even though the prior steps have argued for a highly rational process, it is always good to check your moral sense, as observed by Kant.
   - Intuition is gaining credibility as a source for good decision-making; feeling something is not "right" is a useful trigger.
     —Particularly relevant if you have a lot of experience in the area and are considered an expert decision-maker.
5. Decide on your course of action, and prepare responses to those who may oppose your position.
   - Consider potential actions based on the consequences, obligations, and character approaches.
   - Do you come up with similar answers from the different perspectives?
   - Do your obligations and character help you evaluate the consequentialist preferred action?
   - How can you protect the rights of those involved (or your own character) while still maximizing the overall good for all of the stakeholders?
   - What arguments are most compelling to you to justify the action ethically? How will you respond to those with opposing viewpoints?

References

1. Sayre-McCord, G.: Metaethics, The Stanford Encyclopedia of Philosophy (Summer 2014 Edition). Edward N. Zalta (ed.). SEP
2. Tavani, H.: Ethics and Technology, 3rd edn. Wiley, Hoboken NJ (2011)
3. Cudd, A., Eftekhari, S.: Contractarianism, *The Stanford Encyclopedia of Philosophy* (Winter 2021 Edition), Edward N. Zalta (ed.), SEP
4. Gowans, C.: Moral Relativism, The Stanford Encyclopedia of Philosophy (Summer 2019 Edition). Edward N. Zalta (ed.). SEP
5. D'Agostino, F., Gaus, G., Thrasher, J.: Contemporary Approaches to the Social Contract, The Stanford Encyclopedia of Philosophy (Fall 2019 Edition). Edward N. Zalta (ed.). SEP
6. Sandel, M.J.: Justice: What Is the Right Things to Do. Farrar, Straus and Giroux, New York (2009)
7. de Mori, B.: What moral theory for human rights? Naturalization vs. denaturalization. Etica e Politica **2**(1) (2000)
8. Quinn, M.J.: Ethics for the Information Age, 7th edn. Pearson Education Inc. (2017)
9. Sinnott-Armstrong, W.: Consequentialism, The Stanford Encyclopedia of Philosophy (Summer 2019 Edition). Edward N. Zalta (ed.). SEP
10. Gilboa, I.: Theory of Decision Under Uncertainty. Cambridge University Press, Cambridge (2009)
11. Benton, R.J.: Political expediency and lying: Kant vs Benjamin constant. J. Hist. Ideas **43**(1), 135–144 (1982)
12. Ross, D.: The Right and the Good. Oxford University press (1930)
13. Asimov, I.: I, Robot. Doubleday, New York (1950)
14. Asimov, I.: The Rest of the Robots. Doubleday, New York (1964)
15. Clarke, R.: Asimov's laws of robotics: Implications for information technology. IEEE Computer **26**(12), 53–61 (1993), and **27**(1), 57–66 (1994). Reprinted as Chapter 15 in [40]
16. Anderson, S.L.: The Unacceptability of Asimov's Three Laws of robotics as a Basis for Machine Ethics, Chapter 16 in [40]
17. Spinello, R.A.: Cybernetics: Morality and Law in Cyberspace. Jones & Bartlett Learning, Burlington, Massachusetts (2021)
18. Lützhöft, M.H., Dekker, S.W.A.: On your watch: Automation on the bridge. J. Navigat. **55**, 83–96 (2002)
19. Travis, G.: How the Boeing 737 Max disaster looks to a software developer. IEEE Spectrum (18 April 2019), 19:49 GMT
20. Forster, A.M.: The Machine Stops. In: (the final chapter) [39] (1910)
21. NTSB.: Grounding of the Panamanian Passenger Ship Royal Majesty on Rose and Crown Shoal near Nantucket Massachusetts, June 10, 1995. (NTSB/MAR-97/01). National Transportation Safety Board, Washington DC (1997)
22. Nancy, L.: Safeware: System Safety and Computers. Appendix A: Medical Devices: The Therac-25. Addison-Wesley (1995). This report is separately available at Leveson
23. Baase, S.: A Gift of Fire: Social, Legal, and Ethical Issues for Computing Technology, 4th edn. Pearson, Boston (2013)
24. International Federation of Robotics (IFR).: World Robotics 2020 Edition
25. Müller, V.C.: Ethics of Artificial Intelligence and Robotics, The Stanford Encyclopedia of Philosophy (Winter 2020 Edition). Edward N. Zalta (ed.). SEP
26. https://www.montrealdeclaration-responsibleai.com/the-declaration Montreal Declaration for a responsible development of AI, 2017
27. https://www.theatlantic.com/magazine/archive/2018/06/henry-kissinger-ai-could-mean-the-end-of-human-history/559124/ Henry Kissinger: How the Enlightenment Ends, The Atlantic, June 2018
28. Franssen, M., Lokhorst, G.-J., van de Poel, I.: Philosophy of Technology, The Stanford Encyclopedia of Philosophy (Fall 2018 Edition). Edward N. Zalta (ed.). SEP
29. IEEE Standard for System, Software, and Hardware Verification and Validation.: In: IEEE Std 1012-2016 (Revision of IEEE Std 1012-2012/ Incorporates IEEE Std 1012-2016/Cor1-2017), pp.1–260, 29 Sept. 2017. https://doi.org/10.1109/IEEESTD.2017.8055462
30. Leitner, A., Daniel, W., Javier, I.-G. (eds.): Validation and Verification of Automated Systems, Results of the ENABLE-S3 Project (Springer, 2020)
31. Heaven, D.: Why deep-learning AIs are so easy to fool. Nature **574**, 163–166 (2019). https://doi.org/10.1038/d41586-019-03013-5

32. Churchland, P.: The biology of ethics. The Chronicle of Higher Education (June 12, 2011). Available at Rule Breaker
33. Northcutt, C.G., Athalye, A., Mueller, J.: Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks. arXiv:2103.14749v2[stat.ML]. Summarised by Kyle Wiggers, MIT study finds 'systematic' labeling errors in popular AI benchmark datasets. VentureBeat
34. Maner, W.: Heuristic methods for computer ethics. Metaphilosophy **33**(3), 339–365 (2002). Available at Maner
35. Liffick, B.W.: Analyzing ethical scenarios. Ethicomp. J. **1** (2004)
36. May, D.R.: Steps of the Ethical Decision–Making Process. Accessed at research.ku.edu, November 24, 2020
37. Peslak, A.R.: A review of the impact of ACM code of conduct on information technology moral judgment and intent. J. Comput. Inf. Syst. **47**(3), 1–10 (2007)
38. https://www.nae.edu/225756/Code_of_Conduct NAE Code of Conduct
39. Johnson, D.G., Nissenbaum, H. (eds.): Computers, Ethics & Social Values. Prentice Hall Upper Saddle River, NJ (1995)
40. Anderson, M., Anderson, S.L. (eds.): Machine Ethics. Cambridge University Press, New York, NY (2011)
41. Thunström, A.O., We Asked GPT-3 to Write an Academic Paper about Itself- Then We tried to Get It Published. *Scientific American*, **327**(3), September (2022)

Micha Hofri earned his first two degrees in the department of Physics, at the Technion (IIT) in Haifa, Israel. For his PhD he climbed to a higher hill in the same campus, for the department of management and industrial engineering. His dissertation, for the D.Sc. degree in Industrial engineering and Operations Research, concerned performance evaluation of computer systems. He has been on the faculty at the Technion, Purdue, University of Houston and Rice, and came to WPI as the CS department chairman. He has taught several different courses about operating systems, analysis of algorithms, probability and combinatorics in computing; and more recently about the societal impact of computing and communications. Professor emeritus Hofri continues to offer this recent course. His latest book *Algorithmics of Nonuniformity: Tools and Paradigms,* authored with Hosam Mahmoud, was published in 2019 by CRC Press.