AI & SOCIETY
https://doi.org/10.1007/s00146-020-01024-9

ORIGINAL ARTICLE

The epistemic opacity of autonomous systems and the ethical consequences

Mihály Héder[1]

Received: 30 January 2020 / Accepted: 19 July 2020

Abstract
This paper takes stock of all the various factors that cause the design-time opacity of autonomous systems behaviour. The factors include embodiment effects, design-time knowledge gap, human factors, emergent behaviour and tacit knowledge. This situation is contrasted with the usual representation of moral dilemmas that assume perfect information. Since perfect information is not achievable, the traditional moral dilemma representations are not valid and the whole problem of ethical autonomous systems design proves to be way more empirical than previously understood.

Keywords Epistemic opacity · Ethics of AI · Embodiment · Autonomous systems

1 Introduction

When we discuss the ethical issues of AI or autonomous systems, the debate is often about what a system should or should not do in a given situation. The possible outcomes are unambiguously defined, and there is only a manageable number of them, often as few as just two. The stakes are precise and deterministic, instead of being probabilistic. In other words, our information is perfect. This is especially true about the problem representations meant to the general public.

An excellent example of this is the famous MIT Moral Machine. The Moral Machine, in essence, is a very creatively implemented survey hosted by the MIT Media Lab to gather what would people have the autonomous car to do, given a binary choice. The choice, of course, is between two quite adverse outcomes and yet the survey participant needs to decide, thereby revealing preferences and preservation biases between young and old car crash victims, male or female, dogs vs humans, criminals vs doctors, and so on (Awad et al. 2018).

This survey has gathered over 40 million individual decisions this far and has provided valuable results. The perception of the experiment is overwhelmingly positive. Saxena et al. (2019) argue that this is a valuable input about what the general public—may be differentiated between cultures—judges as fair and thus reinforces the design of algorithms. Others (Kaplan and Haenlein 2020) also see the results of this work as revealing. Of course, the MIT Moral Machine is just one of this kind of moral dilemma representations; there are several others (Goodall 2014a, b; Lin 2014, 2015). There is no question about the usefulness of such problem representations as tools to ascertain the moral preferences of the public. Based on more than 400 thousand subjects, this study was able to shed light on the cultural differences between countries and regions in their moral approach and value appraisal (Fig. 1).

This paper argues, however, that these representations and the debates they generate are not useful at helping the engineering team that is responsible for the design of autonomous systems. In other words, the findings ascertained from such representations are not directly implementable.

The issue is that these representations make it appear like the design team of the autonomous system is in the position of deciding what the system will do in a given well-defined situation. However, this is not the case, since the design team cannot anticipate with full certainty what the designed system would do in a given situation, nor can it be sure that the system will have the correct representation of the case. Moreover, the number of distinct alternatives possible could be quite high and also not definitely known at any point.

In other words, the design team is epistemically handicapped at least in comparison with those perfect information

✉ Mihály Héder
mihaly.heder@filozofia.bme.hu

[1] Budapest University of Technology and Economics, Egry J. u. 1, Budapest 1111, Hungary

Published online: 30 July 2020

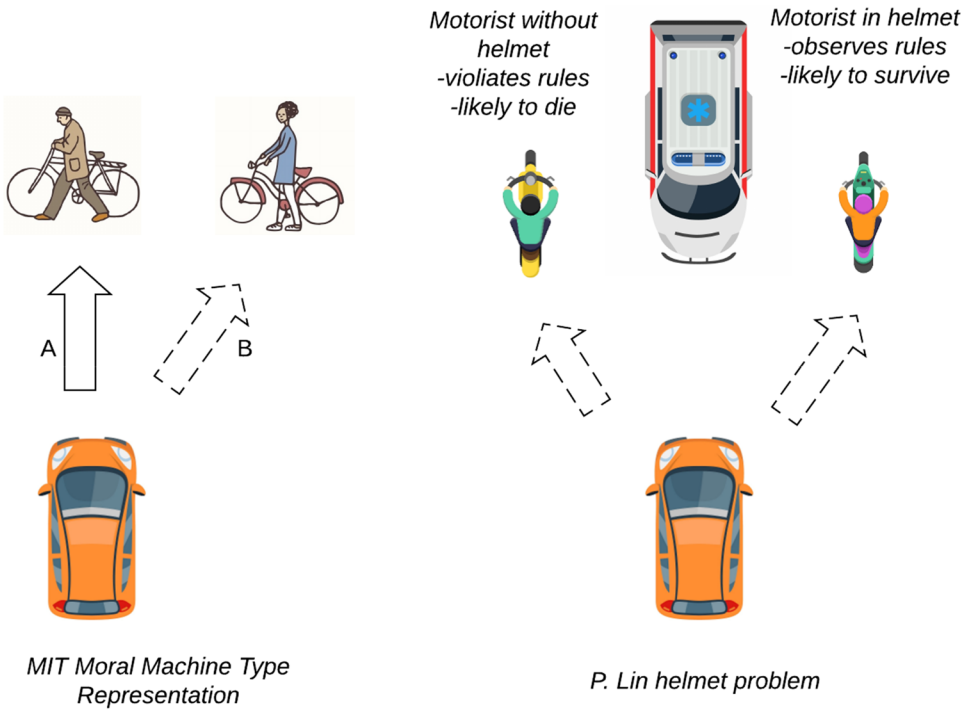

Fig. 1 A simplified representation of the moral machine experiment (left) and another thought experiment about whether to hit the motorist with or without the helmet (right)

cases the usual representations assume. The clarity of the representations of those morally problematic, tricky situations is a myth. The truth about any sufficiently complex system (and all systems of interest for the AI debate are very complex) is that the designers while having quite a high degree of power over their creations, are never in control enough on such a level that would enable them to simply just code a utilitarianist, deontologist, or any other kind of moral behaviour.

Let's call this problem the *design-time knowledge gap*. Those moral problem representations mentioned above are never design-time dilemmas; they are *in-situ* decision problems. When a driver is at the wheel, there is an in-situ moral agent present. Autonomous systems get rid of the agent, therefore eliminating the in-situ decision making altogether. For machine autonomy to work, every moral decision needs to happen design-time, where of course some of the future situations may be envisioned, even simulated. Still, the anticipated and foreseeable circumstances are, of course, just a subset of all the things could happen.

There are two significant reasons for the knowledge gap: the opacity of the surrounding environment of the system and the opacity of the system itself. This paper explains the latter: the sources of opacity and resistance against the designer's intentions within the autonomous system itself. Then, it reveals that this is just a compound problem over the tacit knowledge of the system itself.

This paper will use autonomous vehicles as examples throughout; however, the author believes that the findings are relevant for all autonomous systems.

2 Main discussion

This paper enumerates the several obstacles of algorithmic transparency and argues that many of these are intrinsic to the situation in which humans design autonomous agents, hence there is no hope of overcoming them. After reviewing the terminology, the emergent behaviour of machines, the embodiment effects, the hardware layer factors, statistical knowledge and human factors are presented as obstacles to transparency.

2.1 Terminology

Throughout this article, we will be relying on the terminology of Engineering and Philosophy of Science. This poses a risk on the text not being intelligible for the practitioner of any one field; therefore, it is best to provide some definitions.

Layers and *emergence* feature in this article quite often. The important thing about the usage of these terms is that here they will always refer to epistemic categories. In other philosophical debates, layers may refer to layers of existence and emergence may be ontological. Here, no claims are made about ontology. The paper relies on Michael Polanyi's emergence concept (Polanyi 1958; Paksi and Héder 2020) that is both epistemic and ontological. Still, the latter, more often contended dimension of that theory does not come into play here. Fortunately, it seems that the

epistemic sense of emergence, as well as the differentiation between the layers of engineering design knowledge in correspondence with the architecture of the machines, are practically never contested.

Opacity refers to the epistemic barrier between the engineer and its creation. The created systems will become so complex, and self-modifying (via machine learning) and the engineering teams are so big, that no single person can comprehend it fully. Hence, there is a lack of visibility, or there is the presence of opacity in a sense from a human point of view. This is despite the fact that in the case of software, details are knowable down to the last bit.

The aforementioned *design-time* is an engineering concept. It refers to the time window when a system is being designed. In software engineering *runtime* is the time window when the software runs. The distinction is crucial for us because of the tremendous epistemic gap between the two situations.

Autonomous systems are any AI-governed robotic platforms that operate without continuous human controls. *Autonomous vehicles* or AVs are typical examples of autonomous systems, but there are plenty of others. In this article, all points will be illustrated with AVs but with the intent of being relevant for autonomous systems in general.

Technological stack or sometimes just stack refers to the layered architecture of an engineering solution. This phrase is likely coming from the programming world where software engineers daily mention software stacks, which means layers of software working together, for instance, the operating system may host a software framework that in turn may host some "business logic". The point is that this separation enables the division of labour in two senses: the higher-level pieces of software delegate tasks to the lower levels; also, on the human side, the layers will be developed by different people. This way the most common tasks will be done by the lower layers (like the operating system handles the files and the network), so the higher level can focus on details particular to the task at hand. This layering continues downwards on the hardware level; hence the phrase *technological stack* is created to cover the whole machine.

Underdetermination refers to the term from philosophy of science. Although there are certainly predecessors of the thought, this is commonly credited to Duhem (1954) in the context of scientific theories and to Quine (1951), who extended it to all knowledge claims. In this article, we use the term in the more extended meaning and precisely in the sense these philosophers of science intended.

Inductive and *inductive statistical* will also be mentioned, again from the philosophy of science vocabulary, with Hempel (1958) championing at the elucidation of these terms.

Performance will refer to the success of a system at a task in general.

2.2 Sources of the opacity of autonomous systems

The claim that autonomous system behaviour cannot be designed—on the profound level of detail that matters for implementing a value system or ethics. This is a claim admittedly quite serious. And yet, this is precisely the situation for several reasons.

One reason is the well-known opacity of machine learning systems, especially neural networks. This is subject to several investigations (Pasquale 2016; Diakopoulos 2014; Burrell 2016). This alone would be enough to support the intransparency claim to a very large extent, but there is more.

2.3 Emergent behaviour in autonomous systems

To understand the sources of intransparency, it is useful to examine the layered architecture of autonomous systems and the emergent behaviour that occurs between the layers. In an upper layer, at any point in the architecture, things happen that are within the boundaries set by the lower level, but not entirely governed by them. This is a feature of the architecture, not a flaw: it allows for the operational principles of a higher level (Héder and Paksi 2012; Héder 2019) to work and carry out their function. Following Michael Polanyi's account, we will call this the dual control principle. The exact interplay between the two layers cannot be predicted by either relying upon the descriptions of any of the layers (Héder 2019).

In the framework of personal knowledge, there is no distinction between knowledge and skill, so the evidently skilful autonomous systems may be said to *know.* And that means that there is robot tacit knowledge—or, in fact, all robot knowledge currently can be said to be tacit as it is argued in Héder and Paksi (2012).

To support this claim, let us review what are the levels of an autonomous system, for instance, an autonomous car (Fig. 2).

To illustrate the dual control principle, let us first review a simple example of DNA-based implementation tic-tac-toe.

In this case, we cannot talk about sensors, hardware, operating system or software. The system is the implementation of the perfect tic-tac-toe algorithm (the higher layer) of playing tic-tac-toe based on a chemical system (lower layer), therefore an excellent example of dual control (Fig. 3).

The authors of this solution have created a chemical representation for the cells from 1 to 9, and also a molecular representation to play the game. As long as the game gets the proper input molecules from its human opponent, it is able to respond with a good next move. However, on the level of the algorithm, there is no way to exactly know whether the underlying level of a chemical substance is within the limits it can function properly.

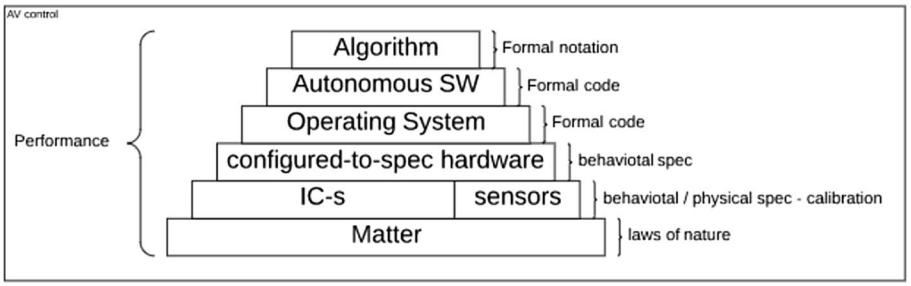

Fig. 2 The technological stack of an autonomous car that achieves the performance of driving autonomously

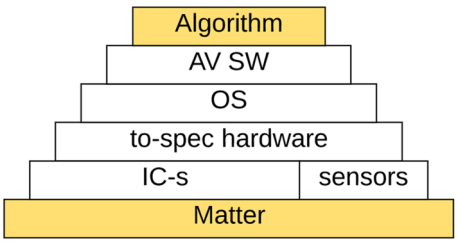

Fig. 3 The layers or "technological stack" realized in a wet computing system, like the DNA-based tic-tac-toe implementation (Stojanovic and Stefanovic 2003)

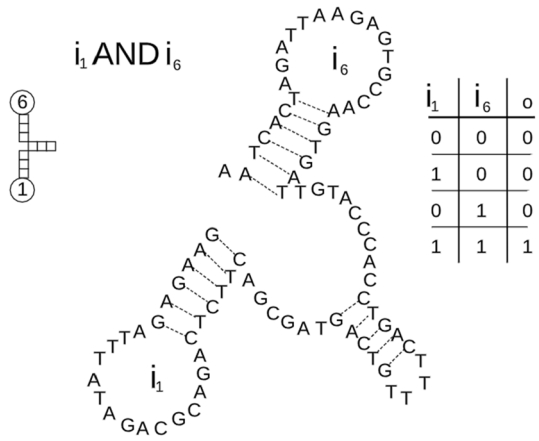

Fig. 4 The realization of a logical AND operator in DNA by Stojanovic and Stefanovic (2003)

Generalizing this claim, in autonomous systems, there is no sure way of self-health-check that can be done on a given level to verify that the conditions of functioning—a proper state of the lower level—are present. There are health checks, for instance, a computer can investigate its file system and find signs that are inconsistent with well-functioning hardware and therefore declare a part of the hardware broken. In information storage, it is always possible to add redundancy that may correct information transfer or retrieval errors. But neither of these solutions are infallible. Going back to the DNA-based example, if the chemical conditions deteriorate, there is no way to tell on the algorithmic level if an answer is faulty or right. Of course, the experimenter sees that problem and may intervene. But that is mainly possible by virtue of being outside the situation and not, for instance, being a lot smarter or being a human. The human experimenter could also have neurobiological problems that would affect its higher-level consciousness in a way that is not so inconsistent that the person in question would know that something is wrong (Fig. 4).

Also, with the DNA-based machine, it is clear that we could introduce inputs that would not work properly and yet the system would muster some sort of response. But the input could be such a chemical compound that would break down the underlying chemical system.

This tic-tac-toe solution is very useful for us as an explanatory tool for the modern autonomous system. Imagine that this solution does not play tic-tac-toe; instead, it is on board of a self-driving vehicle. Now, it is entirely possible that the vehicle navigates itself into a situation, for instance, to some acidic environment that would modify the reactions of the chemical level.

If we substitute the multi-layered AV architecture in the place of this simple, two-layered chemical machine, we will still have the same kind of problem. The activity of a higher level (say, the algorithm) would get the system in a state where a lower level (e.g. hardware) cannot work properly. This way, the higher level can simply destroy the preconditions of its own proper functioning. And the tacit nature of the whole functionality is on clear display: this system does not even use a programming language or any other explicated language. Its body structure contains the knowledge of the perfect tic-tac-toe strategy that will cause it to win every time it has a first-mover advantage—provided that the chemical conditions do not deteriorate.

So, in order to prove that, for instance, our algorithm will not behave in a certain way we don't want it to, we would have to prove that the output of such algorithm—remember it is embedded in an autonomous car in our case—would not govern the whole system into a situation that deteriorates the lower levels.

There could be guarantees in a given layer to remain within certain limits, for instance, in formal systems, there

are ways to prove that the system maintains certain properties at all times (Butler 2001). But this is to be understood with a small print that says as long as the primitives of the system (provided by the underlying layer) work properly.

However, in the case of our autonomous vehicle, the algorithm carries around the hardware, operating system and everything else which poses a risk to these lower layers that is not addressed at all on a higher-level proof. In fact, there is not even a language on a higher level to address those layers.

And if we go below the level of the software, there is no deductive model of deterioration of hardware, for instance. There are inductive-statistical models instead of how many years an average RAM module takes before some of its bits start dysfunction, and that is all we have.

2.4 Embodiment effects

One source of intransparency is what we may call the embodiment effect. Let us imagine that an Autonomous Vehicle technology stack was tested and now is released. The stack includes the mobile platform itself (the car), the computer hardware, the operating system, and the AV software stack running a fine-tuned algorithm. Now let us also assume that an identical vehicle is reproduced except that it is a pale grey. Let us suppose that we find that the safety performance—the number of accident-free hours driven—of this car in the pedestrian-rich environment is lower than the bright yellow one, precisely because of this difference in the colour. Our subsequent investigation establishes that this car is less visible for the pedestrians than the other one, and having no other difference, we will rely on Mill's causal heuristic (Mill 1884) to assume this is the cause for the difference (Fig. 5).

So, how to account for and prepare against this deficiency? The thought experiment highlights the hidden interdependencies between the layers of the technology stack when producing the performance of the car. And if that is true, then the overall engineering blueprint, that is, a different kind of engineering document for each level of the stack (source codes and runtime configuration on the OS and SW stack, blueprints form the mobile platform, etc.) need to reflect on this detail. In other words, the different colours may have to be tested, and only a limited range of offers may be allowed, and the user will be warned that all warranty is lost if an unofficial paint job is applied.

Fig. 5 The particular paint job of the car, an embodiment detail may factor into its performance, yet it is not the scope of the design; therefore design-time these two cars are not even distinguished from each other

This kind of prohibition of alterations of a car is already commonplace. You are not allowed to change your car in many ways, like getting rid of breaks or lights. Furthermore, it is entirely possible that in the world of human-driven cars, a brighter car is statistically already safer given equivalent drivers. The reason why we don't have a regulation for the car colours is that the presence of the driver—accountable, punishable by law if need be—masks this problem. Drivers of grey cars may hit more people than yellow cars all else being equivalent, but this—in theory—will not be taken into account when the responsibility for an accident is established.

When it comes to AV cars, there is no similar responsibility-absorbing agent, so it all comes down to the design and the designer. But the car colour and visibility is just a single example. In fact, there is no way of knowing what the important embodiment factors for which the design should explicate rules are. Every design underdetermines the eventual tangible artefact that will be created based upon it. And the dimensions of the freedoms that can be safely allowed cannot be established with deductive methods.

2.5 Material layer effects on hardware

One of the important bottlenecks of modern computer hardware is cooling; in other words, the effective dissipation of heat. This is tackled in three ways. The first is by making the calculations themselves more energy-effective so that there is less heat generated in the first place. Great advancements have been made in that direction in modern chips, but of course, it is still the case that the generated heat is roughly proportional to the number of calculations made, and as IBM's excellent Landauer (1961) established, there is a theoretical minimum of energy needed for a unit of calculation.

Another way of dealing with the heat is to apply better cooling, but that itself consumes energy and also, is often very noisy.

So the third way is also used: throttling the speed of the computation based on the measured temperature of the hardware. This way the number of computations may be allowed to spike for a short period of time, nicely serving the user, whose typical pattern of computer usage is short bursts of activity followed by a longer period of inactivity, like loading a browser tab and then reading it. The high-frequency computation, of course, cannot be sustained for long, because of the generated heat. To maintain the balance, limitations may be put on the computation frequency.

Because of this heat budget, a lower number of computations can be made in hotter environments, or the same computation takes longer. In the case of the autonomous car, in order to ensure that there is an answer in time, there might be real-time operating systems used, meaning that for certain operations there is a possibility to get an answer within some time period measured in real-time, instead of CPU time, which, as we have just seen, may take different real times in different heat conditions.

So the system will have to guarantee a certain responsivity in any conditions where the vehicle can conceivably find itself. But it makes no sense to make this lower limit to become an upper limit as well: if it can, it should execute more calculations, which in this case may lead to a better classification of an object or more precise location information of the vehicle. The presence of this margin between the minimal and maximal performance makes it very hard to exactly predict just how good the understanding of any situation will be at any moment of time. The designers need to accept that the very same situation in a bit better CPU heat condition might be better recognized than otherwise, and there is just no way of foreseeing the variations that result from this effect.

This is just one way how the processes of the material layer have an effect on the overall system, without breaking it. There are many more such examples, especially when it comes to the performance of sensors, like the cameras and the effects of the changing physical environment on them.

To know these processes to the extent that would allow the clear and well-defined ethical dilemmas we know from the usual representations, we would need a Laplace's demon-like knowledge of our system, which is just not possible.

2.6 Statistical-only knowledge

All the arguments thus far are leading up to a representation in which the situation and the alternative outcomes are only known to some level of confidence. This will lead to the need for probabilistic representations of situations.

When it comes to the probability of the outcomes, another new set of uncertainties come into play. These have to do with the effect appraisal of the vehicle's actions.

For instance, there is no way to know the exact braking distance in any given situation, because that obviously depends on the properties of the surfaces involved, etc. What will happen is that the system implements a feedback loop—a well-known concept from control theory. It will calculate an action that will be put in effect; then the situation will be surveyed again and based on that, further actions will be taken. This is all to keep the system within the parameters set by some higher level of the architecture, like the cruise planning software. However, the resistance of the system—because of its own factors and because of the environment's unpredictability, will limit the extent it can be controlled. It can be imagined like a sort of momentum, sometimes quite literally in the physical sense, other times understood in state space.

The unpredictability of the autonomous system, as the subject of its own control mechanism and also because of the imperfect knowledge of the environment, results in a situation with several uncertainties as depicted below:

Figure 6 shows (on the right-hand side) how the realistically knowable information for the autonomous car looks like in an imagined situation. Since the situation itself isn't known with perfect accuracy, even the input of the supposedly moral decision is uncertain.

Yet, even this representation shows a too optimistic picture. That is, the probabilities depicted here rely on past experience distilled from real data by a machine learning process. The representativity of that learning data itself poses questions. We can expect that the rollout of autonomous applications will not happen all at once in all countries, between all income levels, among all layers and groups of society. So, at best, the learning data will be built from the data gathered from the early adopters. A good example of that is the Tesla autopilot software. It is well-known that it trains on actual driving footage: even in non-autopilot, the software is paying attention to what the driver does and calculates what itself would do in the same situation, and notes, learns from the differences. This means embedded bias on at least two levels: first, the human driving to be taken as etalon will represent the driving habits of the group that can afford such an extensive car. Second, the roads and the environment of training will be the countries that have higher adoption of Teslas. So overall we can say, that the machine representation of situations of moral import is not only probabilistic, but the probabilities themselves are only based on training data at best approximating the actual environment, and being completely different at worst.

2.7 Incalculable human decisions

We have already seen that the autonomous system itself is a source of uncertainty, as well as it's the physical environment. Yet another source of opacity of the outcomes is the reaction of the humans and possibly other autonomous vehicles involved in the situation. Imagine a system that is on a crash course with a pedestrian who could have enough time still to jump left or right, or just stay stationary. If our system decides that it can safely swerve to either left or right, we are not in the clear yet, since the pedestrian may decide to jump exactly where the system tries to steer. In the usual representations, the agency of the participants does not figure into the situation, only the agency of the designed system itself. This is quite obviously false. In any realistic representation,

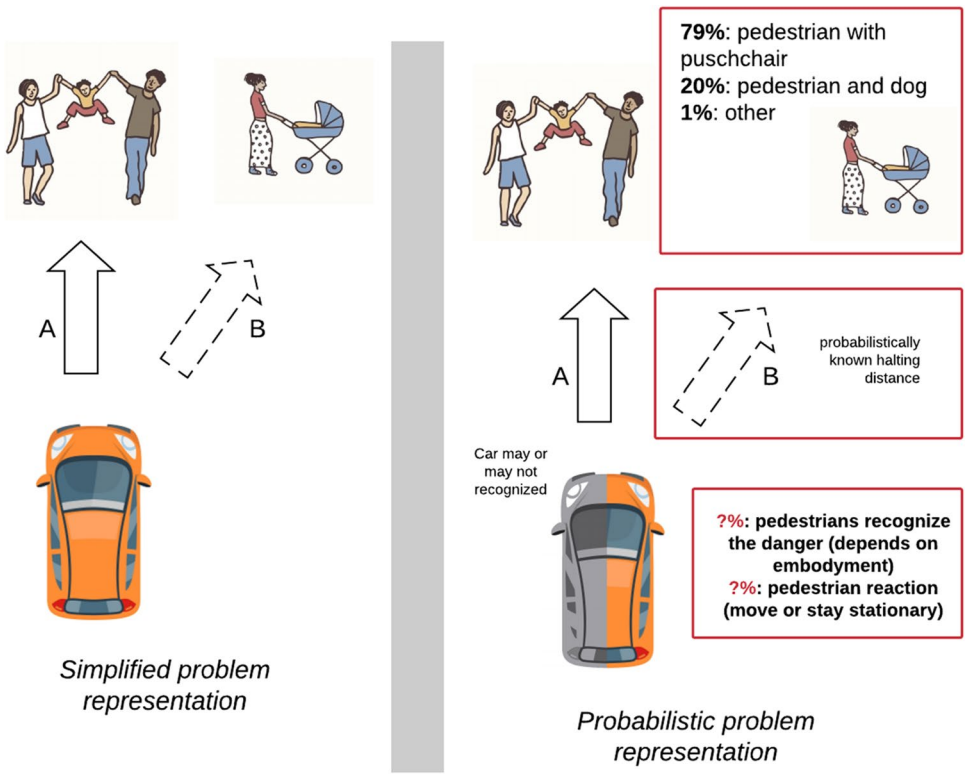

Fig. 6 The probabilistic representation of the problem versus a simplified representation on a situation of moral import

the maximum we can have is maybe a probabilistic model of human behaviour, if anything at all.

There is another kind of incalculable human decisions: malevolent acts against the system and its environment. It has been shown that neural nets are hackable with stickers applied on road signs, invisible (for the human eye) graphic noise, or even by just projecting an image on the wall. There is also a school in attacking IT systems that will try and overwhelm the target system by presenting so much or so well-crafted input to it that it cannot process. Still, another approach taken by attackers is forcing systems in undesirable states by predicting what it would do in certain situations.

In design-time, the best predictions that the engineers can have about these human factors are educated guesses, or worse. This again is another source of opacity.

3 Conclusion

This article showed that the often hidden assumption of perfect information when discussing moral dilemmas is not achievable when designing autonomous systems. On the contrary, there is a huge gap between the epistemic conditions of design-time and use-time. This paper refers to the problem as the *design-time knowledge gap*.

Beyond the epistemic gap, we may realize that there is a moral gap as well. Since there is no user (that may be responsible for what happens in use-time), all the responsibility falls back on the designers, it seems (Fig. 7).

The reasons for this knowledge gap are twofold: the future environment cannot be fully predicted design-time, but more interestingly still, the system behaviour itself cannot be predicted, therefore designed because of its inherent opacity.

The latter is because of several factors. There are always *epistemically emergent* elements of behaviour, and not even by accident but by design: this is one feature of machine learning itself, by which machines acquire tacit knowledge. Activity is happening on *several layers* simultaneously, and only a few of them are formalized enough so that there is even a remote chance to prove that the behaviour stays between pre-set boundaries. *Embodiment effects* refer to important "bodily" or hardware factors and circumstances that are out of the scope of design. *Material effects on hardware,* like heating of the hardware are effects that do not necessarily affect the functionality of the hardware and software layers; however, they may affect the real-time performance of those processes, thereby the whole performance of the system. In other words, the layers of the technological stack not only interact with each other with the interfaces they are designed to, but also in other, incalculable ways.

When we move beyond the hardware layer, we will realize that everything about machine learning, like the training data or the performance, are inductive-statistical and not deterministic. This makes any situation representation only a hypothesis with alternatives in the background and

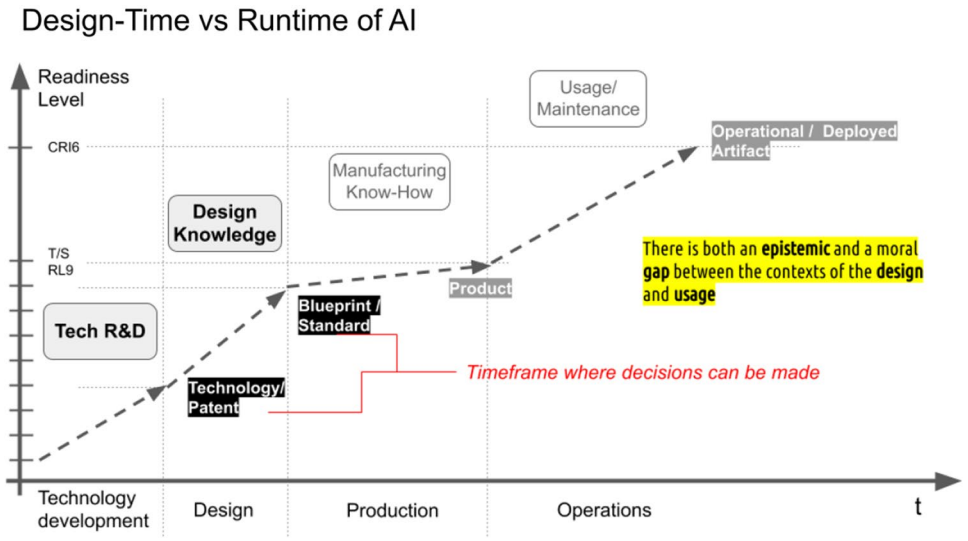

Fig. 7 The epistemic and moral gap between design-time and runtime (or use-time). The designers are not around during runtime, and what makes autonomous systems special, is that no other human agent is around either, compressing all decisions to design-time. On the *y* axis: the technology and commercial readiness levels (Héder 2017)

any supposed in-situ moral decisions should take the fallibility of the representation into account. Finally, the future environment of autonomous systems include humans, whose decisions (like a pedestrian jumps away from a threat or stays still) cannot be predicted, and therefore behaviour based on those decisions cannot be designed. A special category of the human factor is the malevolent act, like tricking or hacking the system on purpose.

To sum up, what counts as good, ethical design decision in any given systems design problem is underdetermined by several factors. Even worse, not all issues manifest itself as design questions in the first place.

Still, moving towards the epistemic transparency of autonomous systems would be a welcome development. To achieve this, instead of attempting to provide ever more details of the design, we may just test systems with human participation to see which systems the users feel they can predict. This shifts the problem to an entirely new level that takes into account the tacit knowledge of the humans involved. After all, people often feel about others that they are reliable, pose no threat, and it is safe being around them. And yet, the exact details of this reliable behaviour are ill-defined and ever-changing, so what really matters is the interface (Gill 2015), and not in the sense of any protocols but in the sense of human engagement with autonomous systems.

A consequence of the statements of this paper is that the design-time ethical problems are much more empirical than they first appear. But they are not really empirical in the sense The Moral Machine investigates the issue—they are empirical in that since on the drawing board autonomous systems are hopelessly opaque, systematic human–machine engagements are necessary to establish the characteristics of the system. This will inevitably lead to an iterative process with many variants and also the ever lingering prospect of unforeseen behaviour in never-tested edge cases.

Acknowledgements Open access funding provided by Budapest University of Technology and Economics. This work was supported by the MTA Bolyai scholarship and the ÚNKP-19-4 New National Excellence Program of the Ministry for Innovation and Technology.

References

Awad E, Dsouza S, Kim R, Schulz J, Henrich J, Shariff A, Rahwan I (2018) The moral machine experiment. Nature 563(7729):59–64

Burrell J (2016) How the machine 'thinks': understanding opacity in machine learning algorithms. Big Data Soc 3(1):2053951715622512

Butler RW (2001) What is formal methods?. NASA LaRC Formal Methods Program

Diakopoulos N (2014) Algorithmic accountability reporting: on the investigation of black boxes. A Tow/Knight Brief. Tow Center for Digital Journalism. https://doi.org/10.7916/D8ZK5TW2

Duhem P [1914] (1954) The aim and structure of physical theory, trans. from 2nd ed. by P. W. Wiener; originally published as La Théorie Physique: Son Objet et sa Structure (Paris: Marcel Riviera & Cie.). Princeton University Press, Princeton

Gill SP (2015) Tacit engagement. Tacit engagement. Springer, Cham, pp 1–34

Goodall NJ (2014a) Ethical decision making during automated vehicle crashes. Transp Res Rec 2424(1):58–65
Goodall NJ (2014b) Vehicle automation and the duty to act. In: Proceedings of the 21st world congress on intelligent transport systems, pp 7–11
Héder M (2017) From NASA to EU: the evolution of the TRL scale in Public Sector Innovation. Innov J 22(2):1–23
Héder M (2019) Michael Polanyi and the epistemology of engineering. In: Héder M, Nádasi E (eds) Essays in post-critical philosophy of technology. Vernon Press, Wilmington
Héder M, Paksi D (2012) Autonomous robots and tacit knowledge. Appraisal 9(2):8–15
Hempel CG (1958) The theoretician's dilemma: a study in the logic of theory construction. University of Minnesota Press, Minneapolis. Retrieved from the University of Minnesota Digital Conservancy. https://hdl.handle.net/11299/184621. Accessed 28 July 2020
Kaplan A, Haenlein M (2020) Rulers of the world, unite! The challenges and opportunities of artificial intelligence. Bus Horiz 63(1):37–50
Landauer R (1961) Irreversibility and heat generation in the computing process. IBM J Res Dev 5(3):183–191
Lin P (2014) The robot car of tomorrow may just be programmed to hit you. WIRED. https://www.wired.com/2014/05/the-robot-car-of-tomorrow-might-just-be-programmed-to-hit-you/. Accessed 28 July 2020
Lin P (2015) Why ethics matters for autonomous cars. M. Maurer et al. (Hrsg.), Autonomes Fahren. https://doi.org/10.1007/978-3-662-45854-9_4
Mill JS (1884) A system of logic, ratiocinative and inductive: being a connected view of the principles of evidence and the methods of scientific investigation, vol 1. Longmans, Green, and Company, Harlow
Paksi D, Héder M (2020) Guide to personal knowledge. Vernon Press, Wilmington
Pasquale F (2016) The black box society. Harvard University Press, Cambridge
Polanyi M (1958) Personal knowledge. University of Chicago Press, Chicago
Quine WV (1951) Main trends in recent philosophy: two dogmas of empiricism. Philos Rev 60:20–43
Saxena NA, Huang K, DeFilippis E, Radanovic G, Parkes DC, Liu Y (2019) How do fairness definitions fare? Examining public attitudes towards algorithmic definitions of fairness. In: Proceedings of the 2019 AAAI/ACM conference on AI, ethics, and Society, pp 99–106
Stojanovic MN, Stefanovic D (2003) A deoxyribozyme-based molecular automaton. Nat Biotechnol 21(9):1069–1074