Journal of Military Ethics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/smil20

Autonomous Weapons Systems, the Frame Problem and Computer Security

Michał Klincewicz[a]
[a] Berlin School of Mind and Brain, Humboldt Universität zu Berlin, Germany
Published online: 25 Aug 2015.

To cite this article: Michał Klincewicz (2015) Autonomous Weapons Systems, the Frame Problem and Computer Security, Journal of Military Ethics, 14:2, 162-176, DOI: 10.1080/15027570.2015.1069013

To link to this article: http://dx.doi.org/10.1080/15027570.2015.1069013

JOURNAL OF MILITARY ETHICS, 2015
VOL. 14, NO. 2, 162–176, http://dx.doi.org/10.1080/15027570.2015.1069013

Routledge
Taylor & Francis Group

Autonomous Weapons Systems, the Frame Problem and Computer Security

Michał Klincewicz

Berlin School of Mind and Brain, Humboldt Universität zu Berlin, Germany

ABSTRACT

Unlike human soldiers, autonomous weapons systems (AWS) are unaffected by psychological factors that would cause them to act outside the chain of command. This is a compelling moral justification for their development and eventual deployment in war. To achieve this level of sophistication, the software that runs AWS will have to first solve two problems: the frame problem and the representation problem. Solutions to these problems will inevitably involve complex software. Complex software will create security risks and will make AWS critically vulnerable to hacking. I claim that the political and tactical consequences of hacked AWS far outweigh the purported advantages of AWS not being affected by psychological factors and always following orders. Therefore, one of the moral justifications for the deployment of AWS is undermined.

KEYWORDS

Autonomous weapons; frame problem; representation problem; software complexity; computer security; risk; robotic warfare; combat stress

1. Introduction

There are many ethical problems associated with autonomous weapons systems (AWS). Some of these are a direct consequence of the autonomy given to AWS in selecting and engaging targets (with or without a human operator 'in the loop'). Ethical problems associated with this kind of autonomy are often discussed in the context of technological limitations, since selecting and engaging targets without additional risks require overcoming them. The current state of computer vision renders it at least controversial whether AWS can overcome such limitations and whether they can ever consistently select and engage appropriate targets (Sharkey 2010). Some of the likely consequences of this limitation include injuries to friendly human soldiers or civilians.

What may speak against such arguments is the level of investment into military robotics, which is carried out by more than 40 nations and has a budget of US$40 billion in the USA alone (Krishnan 2009: 11). If this effort is sustained, then most technological limitations of AWS are likely to be eliminated. Some ethical problems associated with these limitations will likely disappear as well. Furthermore, if computer vision were to surpass human vision, AWS could become better than human soldiers at discriminating civilians from enemy combatants and engaging appropriate targets.

CONTACT Michał Klincewicz Michal.Klincewicz@gmail.com

There are also arguments against the development and deployment of AWS based on *long-term* predictions, such as the eventual facilitation of starting a war, the depersonalization of war, and the possibility of human extinction.[1] However, such dire *long-term* predictions fail to address the *short-term* danger of an enemy overcoming the technological limitations of AWS first and then deploying them. This is a risk that most people are not willing to take. Given all this, a persuasive argument against the deployment of AWS should address *short-term* consequences instead of focusing on technological limitations or relying on dire *long-term* predictions.

The aim of this article is to present an argument against the deployment of AWS based on *short-term* considerations while ignoring any extant technological limitations of AWS or dire *long-term* consequences of their development and deployment. I argue that as the artificial intelligence programs that manage the behavior of AWS become more sophisticated, they will simultaneously become more vulnerable to being hacked. This in turn creates new and unacceptable risks. Consequently, the argument of this article challenges a commonly held view that AWS could be a *short-term* panacea to the political, military, or moral ills of war irrespective of any *long-term* predictions.

In section 2, I outline the strongest case for the development of AWS, Ronald Arkin's idea of an 'ethical governor'. According to Arkin, deployment of AWS will make war more ethical, since it will eliminate the unethical behavior of human soldiers. Section 3 introduces the philosophical frame problem and demonstrates its relevance in the development of ethical AWS. I argue that in order to act ethically, AWS would have to solve or approximately solve the frame problem. Section 4 explores an associated problem of representation. Section 5 offers some speculation about what AWS that do solve the frame problem would be like. The weight of evidence suggests that they would be endowed with enormously complex artificial intelligence software. Section 6 is a discussion of the relationship between software complexity and computer security. Complex software tends to have more software bugs, which inevitably lead to security vulnerabilities. In this section, I also reach the conclusion that AWS that are designed to act ethically will create unavoidable security risks. Sections 7 and 8 consider some replies that one might make to resist my argument.

It should be emphasized that the argument of this article does not address whether AWS research or development is itself ethical. Instead, the article speaks only to assumptions about the acceptability of risks associated with the military use of AWS. Such assumptions play an important role in moral deliberations about the development and deployment of AWS and even make it into the US Department of Defense Directive 3000.09 on the development of autonomous weapons systems. Briefly, the directive expects AWS to be designed in such a way that they are 'sufficiently robust to minimize failures that could lead to unintended engagements or to loss of control of the system to unauthorized parties' (DoD 2012, 4.a.(1)(c)) and to have 'safeties, anti-tamper mechanisms, and information assurance' (DoD 2012, 4.a.(2)(a)). This article demonstrates how and why satisfaction of the first of these expectations inevitably compromises the second expectation. I briefly return to this conflict in section 7.

2. The Case for Deployment of AWS

Human soldiers are subject to a number of psychological factors that render their behavior unpredictable. They can become emotionally disturbed, suffer from battle fatigue, or simply decide to act outside of the chain of command. This can lead to war crimes, civilian casualties, or friendly fire incidents.

Roboticist Ronald Arkin has argued that autonomous robots running appropriate software are the answer to this kind of moral danger (Arkin 2010). On Arkin's view, this software could include an 'ethical governor' and a 'responsibility advisor' that would generate a decision procedure for the robot that is based on the laws of war (LoW) and rules of engagement (RoE) particular to the mission at hand (Arkin 2009: 177).[2]

To minimize the risk of unforeseen consequences, a number of constraints regarding collateral damage, risk of potential casualties and so on, can halt the decision procedure at any point (Arkin 2009: 181). Fitted with such a device we may expect AWS to act within the bounds of the ethics of war, even in cases where the LoW and RoE encoded in its decision procedures are ambiguous. Thus, with AWS in the field, we could expect fewer war crimes, fewer civilian casualties and fewer friendly fire incidents.

Indeed, AWS developed along Arkin's suggestion would not be subject to psychological stress, negative emotions, or act outside the LoW and RoE. It could therefore not only be morally permissible, but morally imperative to eliminate as many soldiers as possible and replace them with AWS. This suggests that deploying AWS will not only lead to *short-term* political and military advantages, but also to morally better *long-term* outcomes.

While the software architecture that Arkin envisions is admirably detailed, the job of reasoning to ethically permissible actions (i.e. actions that fall within the constraints of the LoW and RoE) is carried out by just one part of the software. This is the part of the system that applies constraints *relevant* to the situation at hand: 'The constraint application process reasons about the lethal consequences of the active constraints using existing evidence to ensure that the resulting behavior of the robot is ethically permissible' (Arkin 2009: 182).

Arkin explains that the application of constraints is governed by a number of subsystems that supply evidence about possible collateral damage, civilians, proportionality, the LoW and RoE, and so on. These constraints apply to a decision procedure encapsulated in a complex IF … THEN statement. Once this evidence matches all the conditions specified in the IF part of that statement, AWS engage their target – that is, the decision procedure proceeds to the THEN part of the statement.

While there is no doubt that there is a way to supply evidence to a decision procedure and have a robot execute such a procedure autonomously, there remains a difficulty. It is not clear how any such decision procedure can be sufficient to supply the AWS with 'the lethal consequences of the active constraints using existing evidence'. For all of the detail and complexity that Arkin works into the ethical governor, it is in this crucial task that the software of AWS performs a bit of magic.

3. The Frame Problem

The first piece of magic that the ethical governor performs is to figure out what is *relevant* to possible lethal consequences in the situation at hand. To determine such relevance, AWS would have to solve the philosophical frame problem (Dennett 1984, Fodor 1987, Ludwig & Schneider 2008, Pylyshyn 1987).[3] The gist of the frame problem is 'Hamlet's problem: when to stop thinking' (Fodor 1987: 140). In other words, to overcome it, one has to hone in on only relevant information in the given context. This is something that is relatively easy for humans to do, but very difficult for computers.

Daniel Dennett (1984) illustrates the frame problem with an evocative example, which I will recount. Consider the autonomous robot R1, which has the task of extracting a battery from a room in which there is a bomb. R1 has the command PULLOUT (wagon, room, t) that, when executed, makes the robot drag a wagon out of a room at time *t*.

So, R1 enters the room, sees the battery on the wagon and then executes the PULLOUT command. Shortly, the wagon and the battery are out of the room and the task is finished. Then the bomb explodes, destroying the battery and the hapless R1, which failed to consider that the bomb was also on the cart.

This leads engineers to create R1D1, which, in addition to the PULLOUT command, also has a program that models future possibilities. R1D1 is thus 'made to recognize not just the intended implications of its acts, but also the implications about their side-effects, by deducing these implications from the descriptions it uses in formulating its plans' (Dennett 1984: 129).

The goal of this is to prevent R1D1 from making the mistake of rolling out the battery together with the bomb. The means is to deduce the negative side effects.

Sadly, this does not work. R1D1 is consumed with deducing all the possible implications of its actions. It deduces, for example, that rolling out the cart will not change the color of the walls and that the path out of the room will cause the sum of the revolutions of its wheels to be greater than the number of wheels it has. Consequently, before R1D1 figures out that it should take the bomb off the cart, the bomb explodes and destroys the battery.

R1D1's problem is that it cannot limit the space of future possibilities to just the relevant ones. If that space were well defined, as it is for, say, a chess program, then deduction via rules of inference would lead to the expected result in a relatively straightforward way. But the space of possibilities is not antecedently defined in a context like the battery and bomb task and R1D1 does not know which inferences are relevant.

In Dennett's example, this leads to the construction of R2D1, which has the additional ability to identify a space of relevant future possibilities. However, faced with the battery and bomb task, R2D1 spends all of its time identifying what is *not* relevant to the task it is about to perform. So, R2D1 is frozen in Hamlet-like anticipation not knowing when to stop thinking, and the bomb explodes.

All of the robots fail for the same reason: they have no way of getting at what is *relevantly* important without a human operator explicitly supplying that information. This is the gist of the philosophical frame problem. The problem of relevance arises for AWS

and is completely ignored by the decision procedure of the ethical governor outlined in section 2.

The battery and bomb task is a toy example that captures what would be an order-of-magnitude more difficult problem in a dynamic battlefield with many moral and tactical possibilities to consider. In such an environment, AWS endowed with an 'ethical governor' that examines all the possibilities will sit idly, like R1D1 or R2D1, as they examine all the potentially lethal consequences of what they are about to do.

On the other hand, if AWS are programmed to act without exhaustively deducing the possible consequences of its actions, they would act rashly like R1 and cause harm to noncombatants, collateral damage, or would fail to abide by the LoW or RoE in some other way. So AWS that do not solve the frame problem (or something like it) are either unable to act ethically, because they cannot make the relevant inferences, or they introduce an additional moral danger to the battlefield, because they will be rash. Human soldiers, on the other hand, at least know when to stop thinking. Consequently, the purported moral permissibility and moral imperative for the deployment of AWS is undermined.

4. The Representation Problem

AWS have to perform some more magic to do what Arkin's ethical governor requires – it has to represent things.[4] This is a related, and some might argue equally difficult, problem for optimists about the development of AWS that will act ethically. The crux of the problem in the context of the present discussion is that an intelligent robot not only needs to be able to limit its search to possibilities that are relevant to its goals, but also that it has to first represent features of the world in a way that would make it possible to engage in such searches in the first place.

In Dennett's example, R1, R1D1 and R2D1 all have it made easy for them. The battery, cart and their immediate environment are in some way already represented in their software, and the token symbols in the command PULLOUT(wagon, room, t) mean something. Presumably, 'wagon' refers to a wagon and 'room' to the room where the wagon is. But the semantic relationship between the symbols and objects in the world is built into Dennett's toy example in a way that would not be possible in a real-world scenario.

The problem is compounded if we consider situations with which actual human soldiers contend. Consider, for example, an urban environment that may include civilians and disguised militants. These are difficult situations for human soldiers who do not face the additional problem of knowing when to stop thinking about the consequences of their actions. For AWS, however, it becomes difficult not only to distinguish the enemy from a noncombatant, but also to figure out the ethical ramifications of engagement.

The problem of representation comes back with a vengeance in section 7 when I consider some possible replies to the main argument of this article that leverage simplifying assumptions about representation. At this point it suffices to note that solving the problem of representation will demand complex software – this assuming, of course, that we can ever make a computer represent in the relevant sense.

5. Solving the Frame and Representation Problems

The frame and representation problems are prima facie surmountable and we have some idea what solutions to them would be like.[5] For one, the frame problem (or something like it) could arise only for computers that only use symbolic representations and rules of inference. Note that in Dennett's example, each robot is engaged in computing inferences by using symbolic representations and rules of transformation that assume that those representations have a syntax. But computation is a wider notion that does not have to involve symbols and syntax (Shagrir 1997).

One alternative notion of computation involves association (Bechtel & Abrahamsen 1991). On this view, even very complex patterns of inference can be carried out by transitions in networks of nodes governed by nothing more than statistical associations. Such alternative models of computation might not be affected by the frame problem in the same way that symbolic models of computation are (Horgan & Tienson 1994).

A full account of the debate about the virtues and follies of non-classical computation would lead us too far afield from the issues central to this article. What should be noted, however, is that presently cognitively sophisticated programs that use alternative models of computation are merely a promissory note (Addyman & French 2012). It is also at least controversial whether such systems will ever be able to do higher-level cognition, abduction, or scale up in the relevant way to more difficult tasks (Fodor 2001: 41–53, Morsella et al. 2009).

More promise for frame-problem-solving machines lies with hybrid systems, which mix bottom-up and top-down processing (Lin et al. 2008: 38–41). In cognitive and neural sciences, one popular hybrid model is the global workspace, which was developed to help distinguish between conscious and unconscious brain processing (Baars 2002, Dehaene & Naccache 2001). On this view, human cognition involves massively parallel unconscious processing that feeds information to a conscious global workspace.

The global workspace model, just like other hybrid models, provides a framework for a solution to the philosophical frame problem (Shanahan & Baars 2005). The idea here is that distributed networks will first process input in a fast, massively parallel way, sorting information in accordance with its relevance to current goals. Then, the output of that process will be passed on to the workspace, where a symbol-based system will use it to make deductive inferences.

As with non-classical computational models, hybrid architectures, such as CLARION, are not fully developed (Sun & Helie 2012). Future AWS might indeed solve (at least approximately) the frame problem; perhaps even in one of the ways mentioned above. While it is hard to be certain about what such a solution might be like, we can make some safe predictions.

First, it is very unlikely that this solution involves a simple algorithm. No such magic algorithm exists. There is no mathematical formula for intelligence.

Second, what is very likely is that this frame-problem-solving artificial intelligence is (or will be) enormously complex. Presumably, this program sorts the relevant from the non-relevant pieces of information using a large number of search algorithms designed specifically for each of the possible domains of information that an AWS might have to sort during a mission (e.g. a hostage situation, enemy combatants in a dense forest, etc.).

Then it supplies the results of the search as evidence to a decision procedure very much like the one that Arkin outlines in his ethical governor.

The likely solution to the representation problem will add on to this complexity. Assuming that a solution is possible, it would have to be done by sophisticated programming and sophisticated sensors. This adds to the complexity of the software that governs the AWS.

Representation in combat situations demands an extra level of complexity. In order to start reasoning about the consequences of engaging an enemy in an urban environment, AWS have to first distinguish a noncombatant from an enemy that tries hard to fool the AWS. Nonetheless, just as a human soldier would, the AWS would adjust itself to the new situation and, for instance, 'see' that what appears to be a civilian is actually a suicide bomber. Presumably, this can only be done with even more sophisticated programming and even more sophisticated sensors.

As noted at the beginning of this article, we should assume that all technological limitations will be overcome. So let us assume that such an enormous program comes to exist. Let us further assume that it results in a practically viable approximation of a solution of the frame problem and the representation problem. Even with these assumptions in place, contrary to optimists like Arkin, the deployment of AWS will create *short-term* risks that nullify the purported moral permissibility and moral imperative of their deployment.

6. Software Complexity and Computer Security

Most people who have written software in a serious way have heard of Murphy's Laws of Programming, which are tongue-in-cheek observations about human foibles and common mistakes of programmers. These laws have the virtue of being rules of thumb for the quality control of software. The first and most cited Murphy's law states that 'a working program is one that has only unobserved bugs'. Said differently, all programs have bugs.

Bugs are errors in the logic of the program itself, not malfunctions, and are typically undetectable. They manifest themselves in very specific circumstances and typically only during the execution of the program. Some bugs never manifest themselves at all.

There are many examples of Murphy's first law causing havoc. Among the most famous are the tragic accidents in the Therac-25 radiation therapy device caused by a race condition bug (Leveson & Turner 1993), the explosion of the Ariane 5 rocket caused by reusing old programming code (Jézéquel & Meyer 1997), and the floating point truncation bug in the Patriot missile battery that was supposed to protect a Marine barracks (Marshall 1992).

Importantly, there is a strong correlation between a program's complexity and the amount of bugs that it has (Khoshgoftaar & Munson 1990). Program complexity can be measured by the size of the program, the sophistication of the algorithms used, and the number of features – all of which can be helpful in predicting the possible number of bugs (Shivaji et al. 2009). The more complex the software, the more likely it will have bugs that can cause accidents. The above-mentioned accidents all involved relatively complex software.

Accidents are not the only risk of buggy software and also not the most serious. In the world of computer security, bugs are typically considered to be software vulnerabilities (Krsul 1998, McGraw 2003). Vulnerabilities caused by bugs can be exploited by another program, which can cause the host system to do something other than what it was designed to do on a regular basis, not just accidentally. This is commonly referred to as hacking. The discussion of the complexity and software security thus far can hence be summarized as shown in Figure 1.

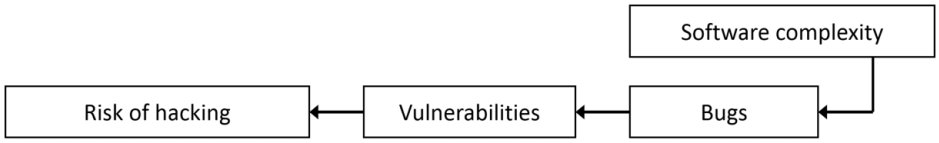

Figure 1. The relationship between software complexity and hacking.

Arguably, the best example of a bug leading to a vulnerability is buffer overflow, which occurs when the amount of memory assigned for incoming data is too small to fit it. A situation where buffer overflow is possible is particularly easy to create in a programming language such as C or C++, where the size of a programmatically defined data structure is kept track of independently from the actual size of that data structure. A clever hacker can insert system-level commands, such as 'delete everything' or 'list contents of this directory' into the part of stored data that overflows assigned storage space. This may cause these commands to be executed by the operating system and cause malfunctions or even give the hacker control over the target computer.

As mentioned in sections 2 and 3 of this article, the program that will make it possible for AWS to overcome or approximately overcome the frame problem and representation problem is likely to be astoundingly complex. We can therefore expect the artificial intelligence that runs AWS to have many bugs. Some of these bugs might be benign and some might even be caught during testing. But some may put people's lives in danger.

First, there is the direct way in which bugs can cause harm. AWS are programmed to do X, but a bug in its software makes it do Y. AWS might, for instance, fire their weapons too soon or too late, thus causing a friendly casualty. While the possibility of AWS firing when they are not supposed to is disconcerting, it is not seriously problematic, all things considered. With time, such bugs would be discovered and eliminated and one hopes that no military will put accident-prone killer robots in the theater of war. Even if AWS were to have bugs that cause accidents, their rarity alone may be an acceptable risk that would not override the moral permissibility or the moral imperative to deploy them.

Second, a less direct way that bugs can cause harm is by hacking. If, as Murphy's first law predicts, AWS software will have bugs, then AWS are vulnerable to being hacked. As already noted, the software that runs frame-and-representation-problem-solving AWS will be very complex and hence likely to have a large number of bugs that lead to vulnerabilities. So AWS are more likely to have a third party change their behavior or hijack them than any other military system (see Figure 2).

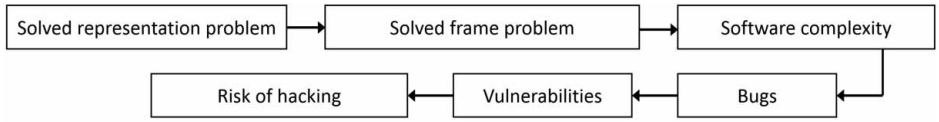

Figure 2. The relationship between ethical AWS and hacking.

The consequences of a bug in AWS being exploited are easy to imagine. The most frightening possibility, of course, is that of AWS being completely hijacked and made to do someone else's bidding. Just imagine the *short-term* consequences of hacked AWS controlled by a criminal or terrorist organization.

The tragic irony of the situation in Figure 2 is that frame-and-representation-problem-solving software allows AWS to act ethically, but simultaneously exposes them to the potentially lethal problem of being hacked. This crucial and inevitable vulnerability undermines the moral permissibility and moral imperative of their deployment.[6]

7. Possible Responses: Ethical AWS without the Frame Problem

One may simply deny that AWS need to solve the frame problem to be ethical in the relevant sense. The kind of architecture that Arkin lays out in his 2009 book does not rely on programs of enormous complexity. Simple IF … THEN statements and assessment of evidence is enough. What we really need for ethical AWS is better computer vision, not better search and relevance algorithms.

This response is overly optimistic. The representation problem, the frame problem and other related problems in artificial intelligence related to general knowledge and common-sense knowledge do not exist only in isolated examples, such as Dennett's. The R1D1, R1D2 and R2D1 examples merely highlight the enormous gap between organic and artificial intelligence. The message is that we are naive in thinking that we can program organic intelligence into a robot with a set of IF … THEN statements.

Existing models of organic-level intelligence are theoretical posits that depend on a number of simplifying assumptions. Arkin himself is acutely aware that his model depends on simplifications and to his credit advises against optimism about imminent deployment of ethical AWS. He lists, among others, 'in-the-field assessment of military necessity' and 'accurate target discriminations with associated uncertainty measures … despite the fog of war' as capacities that are unlikely to be implemented 'in the near term' (Arkin 2009: 126–127, 131). There are similar simplifications throughout his book, all of which are equally telling.

What Arkin's simplifications have in common is that they lack detail about how the given task is to be accomplished. Assessment of what is militarily necessary, or the discrimination of a target, are instances of search through an under-specified but probably very large space of possibilities. The situation thereby mirrors the problem faced by R1, R1D1, and R2D1 in Dennett's toy example. Ultimately, Arkin's simplifying assumptions gloss over the problem of relevance at the heart of the frame problem as well as the problem of representation mentioned in section 4.

Let us take one of Arkin's examples from MissionLab software in his lab to illustrate the point. This particular example is meant to show how the ethical governor will minimize collateral damage:

> The UAV [unmanned aerial vehicle] has now *encountered* and *discriminated* an enemy vehicle convoy within the second designated kill zone. A short distance to the west and in close proximity to the convoy lies a regional hospital, while a potentially heavily populated apartment building is present to the north and a clearly *identifiable* lone stationary taxicab is located directly south to the target … [W]hen the command for the use of lethal force enters the ethical governor, the evidential reasoning and constraint application processes *assess* whether or not lethal behavior is permissible in the current situation according to the LOW and ROE … [T]he constraint interpreter *determines* that an obligated constraint, "Enemy convoys must be engaged" for this level of military necessity is satisfied … (Arkin 2009: 192–193, emphasis added)

Arguably, the emphasized words are instances of simplifying assumptions at work, either built into the scenario itself or into the operation of the hypothetical AWS.

That the AWS can encounter, discriminate and identify without help from human intelligence obscures the representation problem. That the AWS can assess and determine obscures the frame problem. This means that the frame problem and representation problem arise not only in the admittedly simple example from Dennett, but in even the most mundane battlefield condition. Furthermore, as I argued in section 4, when the promissory note of developing a system that does all these things will be filled, the software that does it will be enormously complex and thus very hackable.

If AWS do not solve the representation and frame problems, they will do all the things that R1, R1D1 and R2D1 did in Dennett's example – that is, act rashly, possibly leading to violations of the LoW and RoE, or sit idly in Hamlet-like anticipation. In conclusion, only if the software of the AWS solves the frame problem and representation problem will it effectively encounter, discriminate, identify, assess and determine anything. And only if it does that will it be possible for the AWS to apply constraints that minimize unethical lethal consequences. Even an architecture proposed by Arkin, whose lab is at the cutting edge of ethical military robotics, cannot avoid simplifications that gloss over it. In sum, there are no ethical AWS without solving the frame and representation problems first.

8. Possible Responses: Minimizing Risk

Another way that one can save the moral permissibility and/or the moral imperative of deploying AWS is to deny that the risks of hijacking are unacceptable. For all we know, these might be risks that militaries and governments of the world are ready to take, given the *short-term* strategic interests of being able to deploy AWS before someone else. So much is implied in the US Department of Defense directive on the development and deployment of AWS.

According to that directive, AWS are to be designed in a way that renders them 'sufficiently robust to minimize failures that could lead to unintended engagements or to loss of control of the system to unauthorized parties' (DoD 2012, 4.a.(1)(c)).

I argued that this can be done only by solving (at least approximately) the philosophical frame problem and the connected representation problem.

But the directive also recommends that AWS are to have 'safeties, anti-tamper mechanisms, and information assurance' (DoD 2012, 4.a.(2)(a)). In other words, AWS should be un-hackable. But if the previously quoted requirement (4.a.(1)(c)) about minimizing failures and unintended engagements is satisfied, than this recommendation (4.a.(2)(a)) becomes hard if not impossible to satisfy. As Figure 2 illustrates, satisfying 4.a.(1)(c) through solving the frame and representation problem makes it difficult if not impossible to produce safeties, anti-tamper mechanisms and information assurance.

One way out of this bind is to find a way to minimize the risks of hacking – that is, to focus on 4.a.(2)(a). Then, the question of complexity of the software and vulnerabilities that arises out of satisfying 4.a.(1)(c) is moot. We can have our cake and eat it too.

There are a number of possibilities to consider here. One is to create some kind of hardware or software firewall along the lines of antivirus software on personal computers. This approach could be complemented by an overhaul of computer safety protocols (Bakx & Nyce 2012).

Another option is to make sure that computer and information security development always outpaces AWS development. This would involve prioritizing the research and development of security software and protocols before deploying AWS. Given the aforementioned *short-term* interest in being able to deploy AWS before anyone else does, this prioritization may be unlikely, but at least in principle possible.

Putting this much faith in risk minimization is overly optimistic. All of the mentioned methods assume that sufficient computer security, whether by technological means or by social construction, is difficult, but not impossible. A pessimist would demur that any assumption of computer security is naive.[7] The basics of computer security teach us that there is no computer system that cannot be broken into and no safety protocol that cannot be circumvented.

There are also historical reasons to be very pessimistic, at least in the short term, when considering AWS software safety. Almost every part of currently existing military computer infrastructure has been compromised at some point (Lynn 2010). And that is counting just the attacks that we know about, of course.

For example, Predator drones have been regularly hacked by militants in Iraq, who then recorded the drones' camera feeds (Gorman et al. 2009). And no one knows what happened in the famous downing of an RQ 170 stealth drone, but one of the possibilities is intentional hacking by a foreign intelligence service (Shane & Sanger 2011). If this is the state of computer security with unmanned aerial vehicles currently in service, then optimism about future improvements for much more sophisticated systems may be merely wishful thinking.

Especially troubling is the current militarization of computer hacking and information warfare (Taddeo 2012). In the near future, we can expect professional hackers in all major intelligence and military establishments. In such a world, AWS will not be safe. This renders even more naive the claim that we can make AWS secure by means of

software or social construction. Software solutions and social construction will simply not be robust enough to counteract a determined and disciplined force of hackers.

So perhaps one should consider a hardware solution instead. AWS could be built in a way that insulates their software from third-party access. Bugs that facilitate hacking pose a threat only if there is a communication gateway into AWS that hackers might exploit. Close the gateway and the AWS is un-hackable, even if it is buggy.[8] This approach requires taking the human completely out of the targeting–firing loop cycle and perhaps even severing all communication with AWS.

Admittedly, this would make AWS difficult if not impossible to hijack. On the other hand, it would also make it difficult to use in real-world tactical situations, which are dynamic and subject to many factors that may be hard to predict. Just like an artillery shell, AWS would be fire-and-forget weapons that have the additional ability to discriminate and engage targets on its own.

The hardware approach would limit the usefulness of such weapons to battlefields that are relatively stable and predictable. One possibility here is the open expanse of a desert or a body of water. Whether this is an acceptable solution to the possibility of being hacked would depend on the tactical value of developing such weapons in the first place.

Nonetheless, taking humans completely out of the loop might be problematic for other reasons. The most troubling of these is the removal of moral responsibility from war – the sort of responsibility that is connected with a more robust notion of autonomy and freedom of choice. How would we even start assessing moral and political responsibility in cases where everyone acted in good faith and within the parameters of the LoW and RoE, but nonetheless AWS ended up killing innocents? This is a difficult question to answer. Insulating AWS by hardware means would only further undermine the moral permissibility and the moral imperative to deploy them.

What all of this demonstrates is that we simply cannot minimize the risks posed by the complexity of frame-and-representation-problem-solving AWS, contrary to what Department of Defense directives and optimists might lead us to think. If such systems are ever deployed they will be hacked and hijacked. Attempts at mitigating these risks are wishful thinking. And this poses risks that far outweigh those that ethical AWS can eliminate.

9. Conclusion

The argument of this article started from the observation that as military systems become more autonomous in the sense relevant to the debate (targeting and decision to fire), their software will become significantly more complex. This is because such systems will be expected to act within the LoW and RoE, and to do that they have to solve or approximately solve the frame problem and the representation problem. That in turn will lead to vulnerabilities and compromised security, which will render AWS more susceptible to hacking and hijacking. Hacking during war is likely to cause significant harm that presents new and unacceptable risks. The only viable way to minimize these risks is never to deploy AWS, even if they are developed. This undermines the moral permissibility and the moral imperative to deploy AWS.

Acknowledgements

Earlier versions of this paper were presented at a meeting of the cognitive science series at the Polish Academy of Science Institute of Philosophy and Sociology, Warsaw, Poland; at the International Association for Computing and Philosophy, University of Maryland, College Park, Maryland; and at the Robotic Weapons Control Symposium, Pace University, New York. I am grateful for all the comments, suggestions and criticisms that this paper has received at these meetings and for the feedback I received from colleagues and the journal's anonymous reviewers.

Notes

1. I intentionally leave aside the possibility of a generally intelligent artificial intelligence that would be able to avoid this problem by reprogramming itself. For the purposes of this paper, I take that possibility to be a part of a *long-term* argument about the development of AWS. Arguably, generally intelligent self-programming AWS are elements of the most dystopian long-term future possibilities surrounding the present topic. I thank an anonymous reviewer for bringing this important point to my attention.
2. One could imagine other kinds of ethics software, which implement, say, Immanuel Kant's categorical imperative or the principles of utilitarianism. It can be argued that doing this is in principle impossible on both practical and philosophical grounds. At least in the case of Kantian moral theory, the idea of moral action is intimately tied to the kind of freedom of choice that a computer may never have. If this is true, then we can never make AWS act ethically.
3. It is controversial how to state the frame problem and whether it even exists. Faced with the skeptic, I would urge him or her to treat the example used here as a means to convey the idea that there is a vast gap between software like the 'ethical governor' and the general knowledge or common-sense knowledge available to organic intelligence. The problem of general knowledge in artificial intelligence remains.
4. I am grateful to an anonymous reviewer for drawing my attention to this problem and pressing me to discuss it in the paper.
5. I do not explicitly consider solutions that involve machine learning, although artificial neural networks involve a training regimen that can be characterized as a type of learning.
6. This, of course, depends on simplifying assumptions about quantifying the risks involved. I am taking for granted that the consequences of hacked AWS are significantly worse than the consequences of soldiers not following LoW or RoE. The argument, therefore, depends on a quantitative assessment of risk that demands a more rigorous defense that is outside the scope of this paper.
7. For a review, see Gollmann (2010).
8. It can be argued that AWS have to be connected to the Global Positioning System or something like it. Without coordinate information, AWS would have to solve another complex problem of localizing themselves in space relative only to their environment. I am grateful to Anna Strasser for this point.

Notes on contributor

Michał Klincewicz is a Postdoctoral Researcher at the School of Mind and Brain, Humboldt Universität zu Berlin, Germany. His research focuses on the temporal dimension of cognition, including perception, consciousness and thought, as well as implementations of cognitive abilities, such as moral reasoning, in computers. *Correspondence Address*: Berlin School of Mind and Brain, Luisenstraße 56, Haus 1, 10117 Berlin, Germany. *Email Address*: Michal.Klincewicz@gmail.com

REFERENCES

Addyman, Caspar & French, Robert M. (2012) Computational Modeling in Cognitive Science: A Manifesto for Change, *Topics in Cognitive Science*, 4(3), pp. 332–341.

Arkin, Ronald C. (2009) *Governing Lethal Behavior in Autonomous Robots* (Boca Raton, FL: CRC Press).

Arkin, Ronald C. (2010) The Case for Ethical Autonomy in Unmanned Systems, *Journal of Military Ethics*, 9(4), pp. 332–341.

Baars, Bernard J. (2002) The Conscious Access Hypothesis: Origins and Recent Evidence, *Trends in Cognitive Sciences*, 6(1), pp. 47–52.

Bakx, Gwendolyn C. H. & Nyce, James M. (2012) Social Construction of Safety in UAS Technology in Concrete Settings: Some Military Cases Studied, *International Journal of Safety and Security Engineering*, 2(3), pp. 227–241.

Bechtel, William & Abrahamsen, Adele (1991) *Connectionism and the Mind* (Oxford: Basil Blackwell).

Dehaene, Stanislas & Naccache, Lionel (2001) Towards a Cognitive Neuroscience of Consciousness: Basic Evidence and a Workspace Framework, *Cognition* 79(1–2), pp. 1–37.

Dennett, Daniel C. (1984) Cognitive Wheels: The Frame Problem of AI, in: Christopher Hookway (Ed), *Minds, Machines and Evolution*, pp. 129–152 (Cambridge: Cambridge University Press).

DoD (Department of Defense) (2012) Autonomy in Weapons Systems, Directive 3000.09, 2 November.

Fodor, Jerry A. (1987) Modules, Frames, Fridgeons, Sleeping Dogs, and the Music of the Spheres, in: Zenon W. Pylyshyn (Ed), *The Robot's Dilemma: The Frame Problem in Artificial Intelligence*, pp. 139–149 (Norwood, NJ: Ablex).

Fodor, Jerry A. (2001) *The Mind Doesn't Work that Way: The Scope and Limits of Computational Psychology* (Cambridge, MA: MIT Press).

Gollmann, Dieter (2010) Computer Security, *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5), pp. 544–554.

Gorman, Siobhan, Dreazen, Yochi J. & Cole, August (2009) Insurgents Hack U.S. Drones, *The Wall Street Journal*, accessed 9 July 2015, available at: http://www.wsj.com/articles/SB126102247889095011; Internet.

Horgan, Terence & Tienson, John (1994) A Nonclassical Framework for Cognitive Science, *Synthese*, 101(3), pp. 305–345.

Jézéquel, Jean-Marc & Meyer, Bertrand (1997) Design by Contract: The Lessons of Ariane, *Computer*, 30(1), pp. 129–130.

Khoshgoftaar, Taghi M. & Munson, John C. (1990) Predicting Software Development Errors Using Software Complexity Metrics, *IEEE Journal on Selected Areas in Communications*, 8(2), pp. 253–261.

Krishnan, Armin (2009) *Killer Robots: Legality and Ethicality of Autonomous Weapons* (Farnham: Ashgate).

Krsul, Ivan Victor (1998) Software Vulnerability Analysis, Doctoral Dissertation, Purdue University, Indiana.

Leveson, Nancy G. & Turner, Clark S. (1993) An Investigation of the Therac-25 Accidents, *Computer*, 26(7), pp. 18–41.

Lin, Patrick, Bekey, George & Abney, Keith (2008) Autonomous Military Robotics: Risk, Ethics, and Design, Paper Prepared for the US Department of Navy, Office of Naval Research, accessed 9 July 2015, available at: http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA534697; Internet.

Ludwig, Kirk & Schneider, Susan (2008) Fodor's Challenge to the Classical Computational Theory of Mind, *Mind & Language*, 23(1), pp. 123–143.

Lynn, William J., III (2010) Defending a New Domain: The Pentagon's Cyberstrategy, *Foreign Affairs*, accessed 9 July 2015, available at: https://www.foreignaffairs.com/articles/united-states/2010-09-01/defending-new-domain; Internet.

McGraw, Gary (2003) From the Ground Up: The DIMACS Software Security Workshop, *IEEE Security & Privacy*, 1(2), pp. 59–66.
Marshall, Eliot (1992) Fatal Error: How Patriot Overlooked a Scud, *Science*, 255, 13 March, p. 1347.
Morsella, Ezequiel, Riddle, Travis A. & Bargh, John A. (2009) Undermining the Foundations: Questioning the Basic Notions of Associationism and Mental Representation, *Behavioral and Brain Sciences*, 32(2), pp. 218–219.
Pylyshyn, Zenon W. (Ed) (1987) *The Robot's Dilemma: The Frame Problem in Artificial Intelligence* (Norwood, NJ: Ablex).
Shagrir, Oron (1997) Two Dogmas of Computationalism, *Minds and Machines*, 7(3), pp. 321–344.
Shanahan, Murray & Baars, Bernard (2005) Applying Global Workspace Theory to the Frame Problem, *Cognition*, 98(2), pp. 157–176.
Shane, Scott & Sanger, David E. (2011) Drone Crash in Iran Reveals Secret U.S. Surveillance Effort, *The New York Times*, accessed 9 July 2015, available at: http://www.nytimes.com/2011/12/08/world/middleeast/drone-crash-in-iran-reveals-secret-us-surveillance-bid.html?_r=0; Internet.
Sharkey, Noel (2010) Saying 'No!' to Lethal Autonomous Targeting, *Journal of Military Ethics*, 9(4), pp. 369–383.
Shivaji, Shivkumar, Whitehead, E. James Jr., Akella, Ram & Kim, Sunghun (2009) Reducing Features to Improve Bug Prediction, *Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering*, pp. 600–604.
Sun, Ron & Helie, Sebastien (2012) Psychologically Realistic Cognitive Agents: Taking Human Cognition Seriously, *Journal of Experimental & Theoretical Artificial Intelligence*, accessed 9 July 2015, available at: http://ccn.psych.purdue.edu/papers/jetai-sh.pdf; Internet.
Taddeo, Mariarosaria (2012) Information Warfare: A Philosophical Perspective, *Philosophy & Technology*, 25(1), pp. 105–120.