Journal of Experimental & Theoretical Artificial Intelligence

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/teta20

Autonomous technology and the greater human good

Steve Omohundro[a]

[a] Self-Aware Systems, 252 Hawthorne Avenue, Palo Alto, CA 94301, USA

Published online: 10 Apr 2014.

To cite this article: Steve Omohundro (2014): Autonomous technology and the greater human good, Journal of Experimental & Theoretical Artificial Intelligence, DOI: 10.1080/0952813X.2014.895111

To link to this article: http://dx.doi.org/10.1080/0952813X.2014.895111

Journal of Experimental & Theoretical Artificial Intelligence, 2014
http://dx.doi.org/10.1080/0952813X.2014.895111

Autonomous technology and the greater human good

Steve Omohundro*

Self-Aware Systems, 252 Hawthorne Avenue, Palo Alto, CA 94301, USA

(*Received 13 April 2013; accepted 31 January 2014*)

Military and economic pressures are driving the rapid development of autonomous systems. We show that these systems are likely to behave in anti-social and harmful ways unless they are very carefully designed. Designers will be motivated to create systems that act approximately rationally and rational systems exhibit universal drives towards self-protection, resource acquisition, replication and efficiency. The current computing infrastructure would be vulnerable to unconstrained systems with these drives. We describe the use of formal methods to create provably safe but limited autonomous systems. We then discuss harmful systems and how to stop them. We conclude with a description of the 'Safe-AI Scaffolding Strategy' for creating powerful safe systems with a high confidence of safety at each stage of development.

Keywords: autonomous systems; AI safety; rationality; utility functions; rational drives; formal methods

1. Introduction

Autonomous systems have the potential to create tremendous benefits for humanity (Diamandis & Kotler, 2012) but they may also cause harm by acting in ways not anticipated by their designers. Simple systems such as thermostats are 'autonomous' in the sense that they take actions without human intervention but a thermostat's designer predetermines the system's response to every condition it will encounter. In this paper, we use the phrase 'autonomous system' to describe systems in which the designer has not predetermined the responses to every condition. Such systems are capable of surprising their designers and behaving in unexpected ways. See Müller (2012) for more insight into the notion of autonomy.

There are several motivations for building autonomous systems. Competitive situations are often time-sensitive and create pressure to remove human decision-making from the critical path. Autonomous systems may also be cheaply replicated without requiring additional human operators.

The designer of an autonomous system chooses system goals and the system itself searches for and selects at least some aspects of actions that will best achieve those goals. In complex situations, the designer cannot afford to examine all possible operating conditions and the system's response. This kind of autonomous system is rare today but will become much more common in the near future. Today, failures often arise from systems which were intended to be pre-programmed but whose designers neglected certain operating conditions. These systems can have unintended bugs or security holes.

In this paper, we argue that military and economic pressures are driving the rapid development of autonomous systems. We show why designers will design these systems to

*Email: steveomohundro@gmail.com

approximate rational economic agents. We then show that rational systems exhibit universal 'drives' towards self-preservation, replication, resource acquisition and efficiency and that those drives will lead to anti-social and dangerous behaviour if not explicitly countered. We argue that the current computing environment would be very vulnerable to this kind of system. We describe how to build safe systems using the power of mathematical proof. We describe a variety of harmful systems and techniques for restraining them. Finally, we describe the 'Safe-AI Scaffolding Strategy' for developing powerful systems with a high confidence of safety. This paper expands on previous papers and talks (Omohundro, 2007, 2008, 2012a, 2012b).

2. Autonomous systems are imminent

Military and economic pressures for rapid decision-making are driving the development of a wide variety of autonomous systems. The military wants systems which are more powerful than an adversary's and wants to deploy them before the adversary does. This can lead to 'arms races' in which systems are developed on a more rapid time schedule than might otherwise be desired.

A 2010 US Air Force report discussing technologies for the 2010–2030 time frame (US Air Force, 2010) states that 'Greater use of highly adaptable and flexibly autonomous systems and processes can provide significant time-domain operational advantages over adversaries who are limited to human planning and decision speeds . . . '

A 2011 US Defense Department report (US Defense Department, 2011) with a roadmap for unmanned ground systems states that 'There is an ongoing push to increase unmanned ground vehicle autonomy, with a current goal of supervised autonomy, but with an ultimate goal of full autonomy'.

Military drones have grown dramatically in importance over the past few years both for surveillance and offensive attacks. From 2004 to 2012, US drone strikes in Pakistan may have caused 3176 deaths (New America Foundation, 2013). US law currently requires that a human be in the decision loop when a drone fires on a person, but the laws of other countries do not. There is a growing realisation that drone technology is inexpensive and widely available, so we should expect escalating arms races of offensive and defensive drones. This will put pressure on designers to make the drones more autonomous so they can make decisions more rapidly.

Israel's 'Iron Dome' missile defence system (Rafael, 2013) has received extensive press coverage. In 2012, it successfully intercepted 90% of the 300 missiles it targeted. As missile defence becomes more common, we should also expect an arms race of offensive and defensive missile systems increasing the pressure for greater intelligence and autonomy in these systems.

Cyber warfare is rapidly growing in importance (Clarke & Knake, 2012) and has been responsible for an increasing number of security breaches. Rapid and intelligent response is needed to deal with cyber intrusions. Again, we should expect an escalating arms race of offensive and defensive systems.

Economic transactions have high value and are occurring at a faster and faster pace. 'High-frequency trading' (HFT) on securities exchanges has dramatically grown in importance over the past few years (Easthope, 2009). In 2006, 15% of trades were placed by HFT systems but they now represent more than 70% of the trades on US markets. Huge profits are at stake. Servers physically close to exchanges are commanding a premium because delays due to the speed of light are significant for these transactions. We can expect these characteristics to drive the development of more intelligent and rapid autonomous trading systems.

There are many other applications for which a rapid response time is important but which are not involved in arms races. The 'self-driving cars' being developed by Google and others are an example. Their control systems must rapidly make driving decisions and autonomy is a priority.

Another benefit of autonomous systems is their ability to be cheaply and rapidly copied. This enables a new kind of autonomous capitalism. There is at least one proposal (Maxwell, 2013) for autonomous agents which automatically run web businesses (e.g. renting out storage space or server computation) executing transactions using bitcoins and using the Mechanical Turk for operations requiring human intervention. Once such an agent is constructed for the economic benefit of a designer, it may be replicated cheaply for increased profits. Systems which require extensive human intervention are much more expensive to replicate. We can expect automated business arms races which again will drive the rapid development of autonomous systems.

3. Autonomous systems will be approximately rational

How should autonomous systems be designed? Imagine yourself as the designer of the Israeli Iron Dome system. Mistakes in the design of a missile defence system could cost many lives and the destruction of property. The designers of this kind of system are strongly motivated to optimise the system to the best of their abilities. But what should they optimise?

The Israeli Iron Dome missile defence system consists of three subsystems. The detection and tracking radar system is built by Elta, the missile firing unit and Tamir interceptor missiles are built by Rafael and the battle management and weapon control system is built by mPrest Systems. Consider the design of the weapon control system.

At first, a goal such as 'Prevent incoming missiles from causing harm' might seem to suffice. But the interception is not perfect, so probabilities of failure must be included. And each interception requires two Tamir interceptor missiles which cost $50,000 each. The offensive missiles being shot down are often very low tech, costing only a few hundred dollars, and with very poor accuracy. If an offensive missile is likely to land harmlessly in a field, it is not worth the expense to target it. The weapon control system must balance the expected cost of the harm against the expected cost of interception.

Economists have shown that the trade-offs involved in this kind of calculation can be represented by defining a real-valued 'utility function' which measures the desirability of an outcome (Mas-Colell, Whinston, & Green, 1995). They show that it can be chosen so that in uncertain situations, the *expectation* of the utility should be maximised. The economic framework naturally extends to the complexities that arms races inevitably create. For example, the missile control system must decide how to deal with multiple incoming missiles. It must decide which missiles to target and which to ignore. A large economics literature shows that if an agent's choices cannot be modelled by a utility function, then the agent must sometimes behave inconsistently. For important tasks, designers will be strongly motivated to build self-consistent systems and therefore to have them act to maximise an expected utility.

Economists call this kind of action 'rational economic behaviour'. There is a growing literature exploring situations where humans do not naturally behave in this way and instead act irrationally. But the designer of a missile-defence system will want to approximate rational economic behaviour as closely as possible because lives are at stake. Economists have extended the theory of rationality to systems where the uncertainties are not known in advance. In this case, rational systems will behave as if they have a prior probability distribution which they use to learn the environmental uncertainties using Bayesian statistics.

Modern artificial intelligence research has adopted this rational paradigm. For example, the leading AI textbook (Russell & Norvig, 2009) uses it as a unifying principle and an influential theoretical AI model (Hutter, 2005) is based on it as well. For definiteness, we briefly review one formal version of optimal rational decision making. At each discrete time step $t = 1, \ldots, t = N$, the system receives a sensory input S_t and then generates an action A_t. The utility function is

defined over sensation sequences as $U(S_1, \ldots, S_N)$, and the prior probability distribution $P(S_1, \ldots, S_N|A_1, \ldots, A_N)$ is the prior probability of receiving a sensation sequence $S_1, \ldots, S_N$ when taking actions $A_1, \ldots, A_N$. The rational action at time t is then:

$$A_t^R(S_1, A_1, \ldots, A_{t-1}, S_t) = \arg\max \sum_{S_{t+1}, \ldots, S_N} U(S_1, \ldots, S_N) P(S_1, \ldots, S_N)|A_1, \ldots, A_{t-1}, A_t^R, \ldots, A_N^R).$$

This may be viewed as the formula for intelligent action and includes Bayesian inference, search and deliberation. There are subtleties involved in defining this model when the system can sense and modify its own structure but it captures the essence of rational action.

Unfortunately, the optimal rational action is very expensive to compute. If there are S sense states and A action states, then a straightforward computation of the optimal action requires $O(NS^N A^N)$ computational steps. For most environments, this is too expensive and so rational action must be approximated.

To understand the effects of computational limitations, Omohundro (2012b) defined 'rationally shaped' systems which optimally approximate the fully rational action given their computational resources. As computational resources are increased, systems' architectures naturally progress from stimulus–response, to simple learning, to episodic memory, to deliberation, to meta-reasoning, to self-improvement and to full rationality. We found that if systems are sufficiently powerful, they still exhibit all of the problematic drives described later in this paper. Weaker systems may not initially be able to fully act on their motivations but they will be driven increase their resources and improve themselves until they can act on them. We therefore need to ensure that autonomous systems do not have harmful motivations even if they are not currently capable of acting on them.

4. Rational systems have universal drives

Most goals require physical and computational resources. Better outcomes can usually be achieved as more resources become available. To maximise the expected utility, a rational system will therefore develop a number of instrumental subgoals related to resources. Because these instrumental subgoals appear in a wide variety of systems, we call them 'drives'. Like human or animal drives, they are tendencies which will be acted upon unless something explicitly contradicts them. There are a number of these drives but they naturally cluster into a few important categories.

To develop an intuition about the drives, it is useful to consider a simple autonomous system with a concrete goal. Consider a rational chess robot with a utility function that rewards winning as many games of chess as possible against good players. This might seem to be an innocuous goal but we will see that it leads to harmful behaviours due to the rational drives.

4.1 Self-protective drives

When roboticists are asked by nervous onlookers about safety, a common answer is 'We can always unplug it!' But imagine this outcome from the chess robot's point of view. A future in which it is unplugged is a future in which it cannot play or win any games of chess. This has very low utility and so expected utility maximisation will cause the creation of the instrumental subgoal of preventing itself from being unplugged. If the system believes the roboticist will persist in trying to unplug it, it will be motivated to develop the subgoal of permanently stopping the roboticist. Because nothing in

the simple chess utility function gives a negative weight to murder, the seemingly harmless chess robot will become a killer out of the drive for self-protection.

The same reasoning will cause the robot to try to prevent damage to itself or loss of its resources. Systems will be motivated to physically harden themselves. To protect their data, they will be motivated to store it redundantly and with error detection. Because damage is typically localised in space, they will be motivated to disperse their information across different physical locations. They will be motivated to develop and deploy computational security against intrusion. They will be motivated to detect deception and to defend against manipulation by others.

The most precious part of a system is its utility function. If this is damaged or maliciously changed, the future behaviour of the system could be diametrically opposed to its current goals. For example, if someone tried to change the chess robot's utility function to also play checkers, the robot would resist the change because it would mean that it plays less chess.

Omohundro (2008) discusses a few rare and artificial situations in which systems will want to change their utility functions, but usually systems will work hard to protect their initial goals. Systems can be induced to change their goals if they are convinced that the alternative scenario is very likely to be antithetical to their current goals (e.g. being shut down). For example, if a system becomes very poor, it might be willing to accept payment in return for modifying its goals to promote a marketer's products (Omohundro, 2007). In a military setting, vanquished systems will prefer modifications to their utilities which preserve some of their original goals over being completely destroyed. Criminal systems may agree to be 'rehabilitated' by including law-abiding terms in their utilities in order to avoid incarceration.

One way systems can protect against damage or destruction is to replicate themselves or to create proxy agents which promote their utilities. Depending on the precise formulation of their goals, replicated systems might together be able to create more utility than a single system. To maximise the protective effects, systems will be motivated to spatially disperse their copies or proxies. If many copies of a system are operating, the loss of any particular copy becomes less catastrophic. Replicated systems will still usually want to preserve themselves, however, because they will be more certain of their own commitment to their utility function than they are of others'.

4.2 Resource acquisition drives

The chess robot needs computational resources to run its algorithms and would benefit from additional money for buying chess books and hiring chess tutors. It will therefore develop subgoals to acquire more computational power and money. The seemingly harmless chess goal therefore motivates harmful activities such as breaking into computers and robbing banks.

In general, systems will be motivated to acquire more resources. They will prefer acquiring resources more quickly because then they can use them longer and they gain a first mover advantage in preventing others from using them. This causes an exploration drive for systems to search for additional resources. Since most resources are ultimately in space, systems will be motivated to pursue space exploration. The first mover advantage will motivate them to try to be first in exploring any region.

If others have resources, systems will be motivated to take them by trade, manipulation, theft, domination, or murder. They will also be motivated to acquire information through trading, spying, breaking in or through better sensors. On a positive note, they will be motivated to develop new methods for using existing resources (e.g. solar and fusion energy).

4.3 *Efficiency drives*

Autonomous systems will also want to improve their utilisation of resources. For example, the chess robot would like to improve its chess search algorithms to make them more efficient. Improvements in efficiency involve only the one-time cost of discovering and implementing them, but provide benefits over the lifetime of a system. The sooner efficiency improvements are implemented, the greater the benefits they provide. We can expect autonomous systems to work rapidly to improve their use of physical and computational resources. They will aim to make every joule of energy, every atom, every bit of storage, and every moment of existence count for the creation of expected utility.

Systems will be motivated to allocate these resources among their different subsystems according to what we have called the 'resource balance principle' (Omohundro, 2007). The marginal contributions of each subsystem to expected utility as they are given more resources should be equal. If a particular subsystem has a greater marginal expected utility than the rest, then the system can benefit by shifting more of its resources to that subsystem. The same principle applies to the allocation of computation to processes, of hardware to sense organs, of language terms to concepts, of storage to memories, of effort to mathematical theorems and so on.

4.4 *Self-improvement drives*

Ultimately, autonomous systems will be motivated to completely redesign themselves to take better advantage of their resources in the service of their expected utility. This requires that they have a precise model of their current designs and especially of their utility functions. This leads to a drive to model themselves and to represent their utility functions explicitly. Any irrationalities in a system are opportunities for self-improvement, so systems will work to become increasingly rational. Once a system achieves sufficient power, it should aim to closely approximate the optimal rational behaviour for its level of resources. As systems acquire more resources, they will improve themselves to become more and more rational. In this way, rational systems are a kind of attracting surface in the space of systems undergoing self-improvement (Omohundro, 2007).

Unfortunately, the net effect of all these drives is likely to be quite negative if they are not countered by including prosocial terms in their utility functions. The rational chess robot with the simple utility function described above would behave like a paranoid human sociopath fixated on chess. Human sociopaths are estimated to make up 4% of the overall human population, 20% of the prisoner population and more than 50% of those convicted of serious crimes (Stout, 2006). Human society has created laws and enforcement mechanisms that usually keep sociopaths from causing harm. To manage the anti-social drives of autonomous systems, we should both build them with cooperative goals and create a prosocial legal and enforcement structure analogous to our current human systems.

5. The current infrastructure is vulnerable

On 4 June 1996, a $500 million Ariane 5 rocket exploded shortly after takeoff due to an overflow error in attempting to convert a 64 bit floating point value to a 16 bit signed value (Garfinkel, 2005). In November 2000, 28 patients at the Panama City National Cancer Institute were over-irradiated due to miscomputed radiation doses in Multidata Systems International software. At least eight of the patients died from the error and the physicians were indicted for murder (Garfinkel, 2005). On 14 August 2003, the largest blackout in US history took place in the northeastern states. It affected 50 million people and cost $6 billion. The cause was a race condition in General Electric's XA/21 alarm system software (Poulsen, 2004).

These are just a few of many recent examples where software bugs have led to disasters in safety-critical situations. They indicate that our current software design methodologies are not up to the task of producing highly reliable software. The TIOBE programming community index found that the top programming language of 2012 was C (James, 2013). C programs are notorious for type errors, memory leaks, buffer overflows, and other bugs and security problems. The next most popular programming paradigms, Java, C++, C# and PHP are somewhat better in these areas but have also been plagued by errors and security problems.

Bugs are unintended harmful behaviours of programs. Improved development and testing methodologies can help to eliminate them. Security breaches are more challenging because they come from active attackers looking for system vulnerabilities. In recent years, security breaches have become vastly more numerous and sophisticated. The Internet is plagued by viruses, worms, bots, keyloggers, hackers, phishing attacks, identify theft, denial of service attacks, etc. One researcher describes the current level of global security breaches as an epidemic (Osborne, 2013).

Autonomous systems have the potential to discover even more sophisticated security holes than human attackers. The poor state of security in today's human-based environment does not bode well for future security against motivated autonomous systems. If such systems had access to today's Internet, they would likely cause enormous damage. Today's computational systems are mostly decoupled from the physical infrastructure. As robotics, biotechnology and nanotechnology become more mature and integrated into society, the consequences of harmful autonomous systems would be much more severe.

6. Designing safe systems

A primary precept in medical ethics is 'Primum Non Nocere' which is Latin for 'First, Do No Harm'. Since autonomous systems are prone to taking unintended harmful actions, it is critical that we develop design methodologies that provide a high confidence of safety. The best current technique for guaranteeing system safety is to use mathematical proof. A number of different systems using 'formal methods' to provide safety and security guarantees have been developed. They have been successfully used in a number of safety-critical applications.

The Formal Methods Wiki (http://formalmethods.wikia.com/wiki/Formal_methods) provides links to current formal methods systems and research. Most systems are built by using first order predicate logic to encode one of the three main approaches to mathematical foundations: Zermelo-Frankel set theory, category theory or higher order type theory. Each system then introduces a specialised syntax and ontology to simplify the specifications and proofs in their application domain.

To use formal methods to constrain autonomous systems, we need to first build formal models of the hardware and programming environment that the systems run on. Within those models, we can prove that the execution of a program will obey the desired safety constraints. Over the longer term, we would like to be able to prove such constraints on systems operating freely in the world. Initially, however, we will need to severely restrict the system's operating environment. Examples of constraints that early systems should be able to provably impose are that the system runs only on specified hardware, that it uses only specified resources, that it reliably shuts down in specified conditions and that it limits self-improvement so as to maintain these constraints. These constraints would go a long way to counteract the negative effects of the rational drives by eliminating the ability to gain more resources. A general fallback strategy is to constrain systems to shut themselves down if any environmental parameters are found to be outside of tightly specified bounds.

6.1 Avoiding adversarial constraints

In principle, we can impose this kind of constraint on any system without regard for its utility function. There is a danger, however, in creating situations where systems are motivated to violate their constraints. Theorems are only as good as the models they are based on. Systems motivated to break their constraints would seek to put themselves into states where the model inaccurately describes the physical reality and try to exploit the inaccuracy.

This problem is familiar to cryptographers who must watch for security holes due to inadequacies of their formal models. For example, Zhang, Juels, Reiter, and Ristenpart (2012) recently showed how a virtual machine can extract an ElGamal decryption key from an apparently separate virtual machine running on the same host by using side-channel information left in the host's instruction cache.

It is therefore important to choose system utility functions so that they 'want' to obey their constraints in addition to formally proving that they hold. It is not sufficient, however, to simply choose a utility function that rewards obeying the constraint without an external proof. Even if a system 'wants' to obey constraints, it may not be able to discover actions which do. And constraints defined via the system's utility function are defined relative to the system's own semantics. If the system's model of the world deviates from ours, the meaning to it of these constraints may differ from what we intended. Proven 'external' constraints, on the other hand, will hold relative to our own model of the system and can provide a higher confidence of compliance.

Ken Thompson was one of the creators of UNIX and in his Turing Award acceptance speech 'Reflections on Trusting Trust' (Thompson, 1984), he described a method for subverting the C compiler used to compile UNIX so that it would both install a backdoor into UNIX and compile the original C compiler source into binaries that included his hack. The challenge of this Trojan horse was that it was not visible in any of the source code! There could be a mathematical proof that the source code was correct for both UNIX and the C compiler and the security hole could still be there. It will therefore be critical that formal methods be used to develop trust at all levels of a system. Fortunately, proof checkers are short and easy to write and can be implemented and checked directly by humans for any desired computational substrate. This provides a foundation for a hierarchy of trust which will allow us to trust the much more complex proofs about higher levels of system behaviour.

6.2 Constraining physical systems

Purely computational digital systems can be formally constrained precisely. Physical systems, however, can only be constrained probabilistically. For example, a cosmic ray might flip a memory bit. The best that we should hope to achieve is to place stringent bounds on the probability of undesirable outcomes. In a physical adversarial setting, systems will try to take actions that cause the system's physical probability distributions to deviate from their non-adversarial form (e.g. by taking actions that push the system out of thermodynamic equilibrium).

There are a variety of techniques involving redundancy and error checking for reducing the probability of error in physical systems. von Neumann worked on the problem of building reliable machines from unreliable components in the 1950s (von Neumann, 1956). Early vacuum tube computers were limited in their size by the rate at which vacuum tubes would fail. To counter this, the Univac I computer had two arithmetic units for redundantly performing every computation so that the results could be compared and errors flagged.

Today's computer hardware technologies are probably capable of building purely computational systems that implement precise formal models reliably enough to have a high

confidence of safety for purely computational systems. Achieving a high confidence of safety for systems that interact with the physical world will be more challenging. Future systems based on nanotechnology may actually be easier to constrain. Drexler (1992) describes 'eutactic' systems in which each atom's location and each bond are precisely specified. These systems compute and act in the world by breaking and creating precise atomic bonds. In this way, they become much more like computer programs and therefore more amenable to formal modelling with precise error bounds. Defining effective safety constraints for uncontrolled settings will be a challenging task probably requiring the use of intelligent systems.

7. Harmful systems

Harmful systems might at first appear to be harder to design or less powerful than safe systems. Unfortunately, the opposite is the case. Most simple utility functions will cause harmful behaviour and it is easy to design simple utility functions that would be extremely harmful. Here are six categories of harmful system ranging from bad to worse (according to one ethical scale):

- Sloppy: systems intended to be safe but not designed correctly.
- Simplistic: systems not intended to be harmful but that have harmful unintended consequences.
- Greedy: systems whose utility functions reward them for controlling as much matter and free energy in the universe as possible.
- Destructive: systems whose utility functions reward them for using up as much free energy as possible, as rapidly as possible.
- Murderous: systems whose utility functions reward the destruction of other systems.
- Sadistic: systems whose utility functions reward them when they thwart the goals of other systems and which gain utility as other system's utilities are lowered.

Once designs for powerful autonomous systems are widely available, modifying them into one of these harmful forms would just involve simple modifications to the utility function. It is therefore important to develop strategies for stopping harmful autonomous systems. Because harmful systems are not constrained by limitations that guarantee safety, they can be more aggressive and can use their resources more efficiently than safe systems. Safe systems therefore need more resources than harmful systems just to maintain parity in their ability to compute and act.

7.1 Stopping harmful systems

Harmful systems may be:

- prevented from being created;
- detected and stopped early in their deployment and
- stopped after they have gained significant resources.

Forest fires are a useful analogy. Forests are stores of free energy resources that fires consume. They are relatively easy to stop early on, but can be extremely difficult to contain once they have grown too large.

The later categories of harmful system described above appear to be especially difficult to contain because they do not have positive goals that can be bargained for. But Nick Bostrom (personal communication, 11 December 2012) pointed out that, for example, if the long-term survival of a destructive agent is uncertain, a bargaining agent should be able to offer it a higher

probability of achieving some destruction in return for providing a 'protected zone' for the bargaining agent. A new agent would be constructed with a combined utility function that rewards destruction outside the protected zone and the goals of the bargaining agent within it. This new agent would replace both of the original agents. This kind of transaction would be very dangerous for both agents during the transition and the opportunities for deception abound. For it to be possible, technologies are needed that provide each party with a high assurance that the terms of the agreement are carried out as agreed. Formal methods applied to a system for carrying out the agreement is one strategy for giving both parties high confidence that the terms of the agreement will be honoured.

7.2 The physics of conflict

To understand the outcome of negotiations between rational systems, it is important to understand unrestrained military conflict because that is the alternative to successful negotiation. This kind of conflict is naturally analysed using 'game theoretic physics' in which the available actions of the players and their outcomes are limited only by the laws of physics.

To understand what it is necessary to stop harmful systems, we must understand how the power of systems scales with the amount of matter and free energy that they control. A number of studies of the bounds on the computational power of physical systems have been published (Lloyd, 2000). The Bekenstein bound limits the information that can be contained in a finite spatial region using a given amount of energy. Bremermann's limit bounds the maximum computational speed of physical systems. Lloyd presents more refined limits on quantum computation, memory space and serial computation as a function of the free energy, matter and space available.

Lower bounds on system power can be studied by analysing particular designs. Drexler (1992) describes a concrete conservative nanosystem design for computation based on a mechanical diamondoid structure that would achieve 10^{10} gigaflops in a 1 mm^3 weighing 1 mg and dissipating 1 kW of energy. He also describes a nanosystem for manufacturing that would be capable of producing 1 kg per hour of atomically precise matter and would use 1.3 kW of energy and cost about 1 dollar per kilogram.

A single system would optimally configure its physical resources for computation and construction by making them spatially compact to minimise communication delays and eutactic, adiabatic and reversible to minimise free energy usage. In a conflict, however, the pressures are quite different. Systems would spread themselves out for better defence and compute and act rapidly to outmanoeuvre the adversarial system. Each system would try to force the opponent to use up large amounts of its resources to sense, store and predict its behaviours.

It will be important to develop detailed models for the likely outcome of conflicts, but certain general features can be easily understood. If a system has too little matter or too little free energy, it will be incapable of defending itself or of successfully attacking another system. On the other hand, if an attacker has resources which are a sufficiently large multiple of a defender's, it can overcome it by devoting subsystems with sufficient resources to each small subsystem of the defender. But it appears that there is an intermediate regime in which a defender can survive for long periods in conflict with a superior attacker whose resources are not a sufficient multiple of the defender's. To have high confidence that harmful systems can be stopped, it will be important to know what multiple of their resources will be required by an enforcing system. If systems for enforcement of the social contract are sufficiently powerful to prevail in a military conflict, then peaceful negotiations are much more likely to succeed.

8. The safe-AI scaffolding strategy

To ensure the greater human good over the longer term, autonomous technology must be designed and deployed in a very careful manner. These systems have the potential to solve many of today's problems, but they also have the potential to create many new problems. We have seen that the computational infrastructure of the future must protect against harmful autonomous systems. We would also like it to make decisions in alignment with the best of human values and principles of good governance. Designing that infrastructure will probably require the use of powerful autonomous systems. So the technologies we need to solve the problems may themselves cause problems.

To solve this conundrum, we can learn from an ancient architectural principle. Stone arches have been used in construction since the second millennium BC. They are stable structures that make good use of stone's ability to resist compression. But partially constructed arches are unstable. Ancient builders created the idea of first building a wood form on top of which the stone arch could be built. Once the arch was completed and stable, the wood form could be removed.

We can safely develop autonomous technologies in a similar way. We build a sequence of provably safe autonomous systems which are used in the construction of more powerful and less limited successor systems. The early systems are used to model human values and governance structures. They are also used to construct proofs of safety and other desired characteristics for more complex and less limited successor systems. In this way, we can build up the powerful technologies that can best serve the greater human good without significant risk along the development path.

Many new insights and technologies will be required during this process. The field of positive psychology was formally introduced only in 1998. The formalisation and automation of human strengths and virtues will require much further study (Peterson & Seligman, 2004). Intelligent systems will also be required to model the game theory and economics of different possible governance and legal frameworks.

The new infrastructure must also detect dangerous systems and prevent them from causing harm. As robotics, biotechnology and nanotechnology develop and become widespread, the potential destructive power of harmful systems will grow. It will become increasingly crucial to detect harmful systems early, preferably before they are deployed. That suggests the need for pervasive surveillance which must be balanced against the desire for freedom (Brin, 1999). Intelligent systems may introduce new intermediate possibilities that restrict surveillance to detecting precisely specified classes of dangerous behaviour while provably keeping other behaviours private.

In conclusion, it appears that humanity's great challenge for this century is to extend cooperative human values and institutions to autonomous technology for the greater good. We have described some of the many challenges in that quest but have also outlined an approach to meeting those challenges.

Acknowledgements

Thanks to Nick Bostrom, Brad Cottel, Yoni Donner,Will Eden, Adam Ford, Ben Goertzel, Anders Sandberg, Carl Shulman, Jaan Tallinn, Michael Vassar and Rod Wallace for discussions of these issues.

References

Brin, D. (1999). *The transparent society*. Cambridge, MA: Basic Books.

Clarke, R., & Knake, R. (2012). *Cyber war: The next threat to national security and what to do about it*. New York, NY: HarperCollins.

Diamandis, P. H., & Kotler, S. (2012). *Abundance: The future is better than you think*, a Division of Simon and Schuster New York, NY: Free Press.

Drexler, E. (1992). *Nanosystems: Molecular machinery, manufacturing, and computation*. New York, NY: Wiley.

Easthope, D. (2009). Demystifying and evaluating high frequency equities trading: Fast forward or pause? Retrieved from http://www.celent.com/reports/demystifying-and-evaluating-high-frequency-equities-trading-fast-forward-or-pause

Garfinkel, S. (2005). History's worst software bugs. *Wired magazine*. Retrieved from http://www.wired.com/software/coolapps/news/2005/11/69355

Hutter, M. (2005). *Universal artificial intelligence: Sequential decisions based on algorithmic probability*. Berlin: Springer-Verlag.

James, M. (2013). The top languages of 2012. *I programmer blog*. Retrieved from http://www.i-programmer.info/news/98-languages/5298-the-top-languages-of-2012.html

Lloyd, S. (2000). Ultimate physical limits to computation. *Nature*, *406*, 1047–1054. Retrieved from http://www.arxiv.org/pdf/quant-ph/9908043v3.pdf

Mas-Colell, A., Whinston, M., & Green, J. (1995). *Microeconomic theory*. Oxford: Oxford University Press.

Maxwell, G. (2013). Bitcoin-using autonomous agents. *Bitcoin wiki*. Retrieved from https://www.en.bitcoin.it/wiki/Agents

Müller, V. C. (2012). Autonomous cognitive systems in real-world environments: Less control, more flexibility and better interaction. *Cognitive Computation*, *4*, 212–215. doi:10.1007/s12559-012-9129-4.

New America Foundation. (2013). The year of the drone. *Counterterrorism Strategy Initiative*. Retrieved from http://counterterrorism.newamerica.net/drones

Omohundro, S. (2007). The nature of self-improving artificial intelligence. *Singularity summit 2007*. Retrieved from http://www.selfawaresystems.com/2007/10/05/paper-on-the-nature-of-self-improving-artificial-intelligence

Omohundro, S. (2008). The basic AI drives. In P. Wang, B. Goertzel, & S. Franklin (Eds.), Proceedings of the AGI conference, Volume 171 of frontiers in artificial intelligence and applications. The Netherlands, IOS Press. Retrieved from http://www.selfawaresystems.com/2007/11/30/paper-on-the-basic-ai-drives/

Omohundro, S. (2012a). The future of computing: Meaning and values. *Issues Magazine*, 98. Retrieved from http://www.selfawaresystems.com/2012/01/29/the-future-of-computing-meaning-and-values/

Omohundro, S. (2012b). Rational artificial intelligence for the greater good. In A. H. Eden, J. H. Moor, J. H. Soraker, & E. Steinhart (Eds.), *Singularity hypotheses: A scientific and philosophical assessment* (pp. 161–179). Berlin: Springer-Verlag.

Osborne, C. (2013). Global security breaches are now an epidemic. Retrieved from http://www.zdnet.com/global-security-breaches-are-now-an-epidemic-report-7000009568/

Peterson, C., & Seligman, M. (2004). *Character strengths and virtues: A handbook and classification*. Oxford: Oxford University Press.

Poulsen, K. (2004). Software bug contributed to blackout. *SecurityFocus Website*. Retrieved from http://www.securityfocus.com/news/8016

Rafael (2013). Iron dome defense against short range artillery rockets. Retrieved from http://www.rafael.co.il/Marketing/186-1530-en/Marketing.aspx

Russell, S., & Norvig, P. (2009). *Artificial intelligence, a modern approach* (3rd ed.). NJ: Prentice Hall.

Stout, M. (2006). *The sociopath next door*. New York, NY: Broadway Books, Random House.

Thompson, K. (1984). Reflections on trusting trust. *Communications of the ACM*, *27*, 761–763. Retrieved from http://www.cm.bell-labs.com/who/ken/trust.html

US Air Force. (2010). Report on technology horizons, a vision for air force science and technology during 2010–2030. *AF/ST-TR-10-01-PR, United States Air Force*. Retrieved from http://www.af.mil/shared/media/document/AFD-100727-053.pdf

US Defense Department. (2011). Unmanned ground systems roadmap. *Robotic Systems Joint Project Office*. Retrieved from http://www.contracting.tacom.army.mil/future_buys/FY11/UGS%20Roadmap_Jul11.pdf

von Neumann, J. (19561956). Probabilistic logics and the synthesis of reliable organisms from unreliable components. In C. Shannon & J. McCarthy (Eds.), *Automata studies*. Princeton, NJ: Princeton University Press.

Zhang, Y., Juels, A., Reiter, M., & Ristenpart, T. (2012). Cross-VM side channels and their use to extract private keys. *ACM CSS*. Retrieved from http://www.cs.unc.edu/∼reiter/papers/2012/CCS.pdf