1
Introduction

While the idea of an 'Autonomous System', a system that can make complex decisions without human intervention, is both appealing and powerful, actually developing such a system to be safe, reliable, and trustworthy, is far from straightforward. An important aspect of this development involves being able to verify the decision-making that forms the core of many truly autonomous systems. In this book, we will introduce a particular approach to the formal verification of agent-based autonomous systems, leading the reader through autonomous systems architectures, agent programming languages, formal verification, agent model-checking, and the practical analysis of autonomous systems.

In this introductory chapter we will address the following aspects.

- What is an *Autonomous System*?
  $\longrightarrow$ from automatic, to adaptive, then on to autonomous systems
- Why are *Autonomous Systems* used?
  $\longrightarrow$ with increased flexibility, wider applicability, and significant future potential
- Why apply *Formal* Verification?
  $\longrightarrow$ provides a strong mathematical basis and increased confidence in systems
- What it means in *Practice*?
  $\longrightarrow$ with an impact on safer decisions, ethical behaviour, certification, and so on

https://doi.org/10.1017/9781108755023.001 Published online by Cambridge University Press

1.1 What is an Autonomous System?

1.1.1 From Automatic, through Adaptive, on to Autonomous

The concept of *Autonomy* can be characterised as

the ability of a system to make its own decisions and to act on its own, and to do both without direct human intervention.

We want to distinguish this from both automatic and adaptive systems. An **automatic** system follows a pre-scripted series of activities, with very little (if any) potential for deviation. An **adaptive** system will modify its behaviour, but does so rapidly in order to adapt to its environment (Sastry and Bodson, 1994). This means that its behaviour is tightly based on the inputs or stimuli from its environment and adaptation is typically (especially in adaptive control systems) achieved through continuous feedback loops usually described using differential equations. These are common features of adaptive systems, with continuous feedback control responding to changes in the environment.

Example. Consider a legged robot walking in a straight line across some ground that varies between a hard tarmac surface, a smooth icy surface, and a soft sandy surface. If the robot is controlled by an adaptive system, then as the properties of the ground change, the system can adapt the control of the legs: taking smaller, slower steps on the icy surface, or lifting the legs higher on the sandy surface.

While the behaviour of this system varies flexibly based on the environment it encounters, we would tend not to consider this adaptation to be decision-making and would not describe the system as autonomous unless it also had the capacity to turn aside from its straight line in order to, for instance, examine some object of interest.

By contrast, an **autonomous** system does more than simply react and adapt to its environment. It may have many different reasons for making a choice, and often these are not at all apparent to an external observer. The important thing is that an autonomous system might not be directly driven by immediate factors in its environment. It is often natural to talk of an autonomous system having goals that it is trying to achieve and seeing its behaviour as influenced both by its goals and its current environment.

It should be noted that the distinction between a system that makes its own decisions and one that is purely automatic or adaptive is often difficult to draw,

https://doi.org/10.1017/9781108755023.001 Published online by Cambridge University Press

particularly if you do not want the definition to depend upon the specifics of how the system is implemented. In general, the greater the ability of system to behave flexibly in dynamic, uncertain environments and cope well with scenarios that may not have been conceived of, or only partially specified, when the system was designed, the more autonomous the system is usually considered to be. This book concerns itself with one particular methodology for providing greater autonomy – agent programming – and how the autonomy provided in this way may be verified.

1.1.2 Variable Autonomy

In practical applications there are many levels of autonomy. While fully autonomous systems still remain quite rare, there are many systems that involve a mixture of human and system control. These can range from direct human (e.g., operator/pilot/driver) control of *all* actions all the way through to the system controlling decision-making and action with only very limited (if any) human intervention. This spectrum of *variable autonomy* is so common that several taxonomies have been developed; below is one such classification, called 'PACT' (Bonner et al., 2004), often used in aerospace scenarios.

Level 0: 'No Autonomy'
→ *Whole task is carried out by the human except for the actual operation*

Level 1: 'Advice only if requested'
→ *Human asks system to suggest options and then human makes selection*

Level 2: 'Advice' → *System suggests options to human*

Level 3: 'Advice, and if authorised, action'
→ *System suggests options and also proposes one of them*

Level 4: 'Action unless revoked'
- 4a: *System chooses an action and performs it if the human approves*
- 4b: *System chooses an action and performs it unless the human disapproves*

Level 5: 'Full Autonomy'
- 5a: *System chooses action, performs it, and informs the human*
- 5b: *System does everything autonomously*

An interesting aspect of this, and a current research topic, concerns the mechanism by which a system changes between these levels. Not only when can the operator/pilot/driver give the system more control, but when can the system relinquish some/all control back to the human?

https://doi.org/10.1017/9781108755023.001 Published online by Cambridge University Press

Aside. Consider a convoy (or 'road train') of cars on a motorway. The driver chooses to relinquish control to his/her vehicle and sits back as the car coordinates with other vehicles in the convoy. Some time later, the driver decides to take back control of the vehicle. In principle the vehicle should let him/her do this, but what if the vehicle assesses the situation and works out that allowing the driver to take control in this way will very likely lead to an accident. Should the car refuse to let the driver have control back? Should the car only let the driver have *partial* control back? And what are the legal/ethical considerations?

1.2 Why Autonomy?

Higher levels of autonomy are increasingly appearing in practical systems. But why? There are traditionally several reasons for this trend, and we begin with *distant* or *dangerous* environments. If a system needs to be deployed in a *remote* environment, then direct human control is likely to be infeasible. For example, communications to planetary rovers take a *long* time, while communications to deep sea vehicles can be prone to failure. Perhaps more importantly, the remote control of an autonomous system is notoriously difficult. Even with unmanned aircraft, as soon as the vehicle goes out of sight, its direct control by a human operator on the ground is problematic. Similarly, there are many *dangerous* situations where humans cannot be nearby, and so cannot easily assess the possibilities and confidently control the system. Such environments include space and deep sea, as above, but can also include closer environments such as those involving nuclear or chemical hazards.

In some cases, the environment is neither distant nor dangerous, yet a human is not able to effectively control the system as his/her reactions are just not quick enough. Imagine an automated trading system within a stock market – here the speed of interactions is at the millisecond level, which is beyond human capabilities. In some scenarios there are just too many things happening at once. Possibly a human can remotely control a single unmanned air vehicle, as long as it remains in the controller's line of sight. But what if there are two, or twenty, or two hundred such vehicles? A single human controller cannot hope to manage all their possible interactions.

There are increasingly many cases where a human *could* carry out various tasks, but finds them just too dull. In the case of a robot vacuum cleaner, we could clearly sweep the floor ourselves but might find the task boring and

https://doi.org/10.1017/9781108755023.001 Published online by Cambridge University Press

mundane. Instead, we utilise the robot vacuum cleaner and use the time to tackle something more interesting.

Finally, it may well be that using an autonomous system is actually *cheaper* than using a human-controlled one. With training, safety regulations, and ongoing monitoring required for human pilots, drivers, or operators, possibly an autonomous solution is more cost effective.

Applications

Unsurprisingly, there are very many potential applications for autonomous systems. Few have yet made it to reality and, those that have, rarely employ full autonomy. So, below, we explore some of the possibilities, both existing and future.

Before doing that, however, we note that there is clearly a whole class of purely software applications that have autonomy at their heart. Typically, these are embedded within internet algorithms, e-commerce applications, stock trading systems, and so on. However, for the rest of this section, we will ignore such applications, focussing on those that have more of a physical embodiment.

Embedded Applications. While many applications are explicitly autonomous (e.g., robots, unmanned aircraft) where it is clear to users as well as programmers that the system makes its own decisions, there are perhaps more that are implicit, with autonomous behaviour being embedded within some other system and not necessarily obvious to users. Particularly prominent are the range of *pervasive* or *ubiquitous* systems, typically characterised by multiple computational entities and multiple sensors all situated within an open communications framework. Examples include *communications networks* where some form of autonomy is used for reconfiguration or re-routing or *autonomous sensor networks* where the sensing task is autonomously organised by the sensor nodes. Embedded autonomy also appears within smart cities, smart homes, and so on (see Figure 1.1), where the monitoring aspects are linked to decisions about the system, for example, controlling traffic flow, controlling the environment, deciding what to do in exceptional situations, and so on. More generally, these examples are all varieties of *pervasive* or *ubiquitous* systems.

Autonomous Vehicles. Vehicles of various forms (automotive, air, space, underwater, etc.) increasingly incorporate at least adaptive behaviour and sometimes autonomous behaviour. The 'driver-less car' is one, particularly high profile, example. While both driving in a single lane and obstacle avoidance are

https://doi.org/10.1017/9781108755023.001 Published online by Cambridge University Press

Figure 1.1 'Smart' home
Source: DrAfter123/DigitalVisionVectors via Getty Images. Used with permission

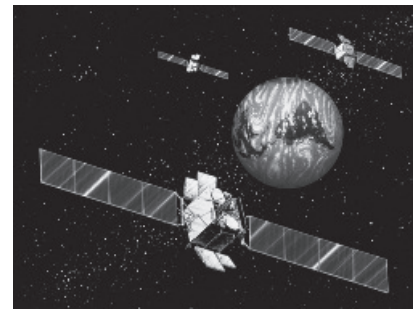
essentially adaptive, the role of the human driver in terms of high-level decisions is increasingly being carried out by software on the vehicle that can make decisions about which route to take and could even foreseeably choose destinations such as petrol stations, supermarkets, and restaurants. While, at the time of writing, there are a number of technological and regulatory obstacles in the way of fully autonomous vehicles, automotive manufacturers are quickly moving towards convoying or 'road train' technology. However, as described earlier, there are clearly some legal/regulatory and ethical questions surrounding even this limited form of autonomous behaviour, especially when the system must decide whether to allow control to be given back to the driver or not.

Moving from ground vehicles, the motivation for utilising autonomy becomes stronger. When vehicles are to move through the air, underwater, or in space, direct pilot/driver control can become difficult. Some of these environments are also potentially dangerous to any human in the vehicle. It is not surprising then that autonomy is increasingly being built into the controlling software for such vehicles. For example, choices made without human intervention are an important element in many aerospace applications, such as *unmanned air vehicles* (Figure 1.2a) or cooperative *formation flying satellites* (Figure 1.2b).

Robotic Assistants. The use of industrial robotics, for example in manufacturing scenarios, is well established. But we are now moving towards the use of

https://doi.org/10.1017/9781108755023.001 Published online by Cambridge University Press

(a) Unmanned Aircraft
Source: Stocktrek Images/Stocktrek Images via Getty. Used with permission.



(b) 'Formation Flying' Satellites
Source: Stocktrek/Stockbyte via Getty. Used wtih permission.

Figure 1.2 Autonomous vehicles

Figure 1.3 Care-o-Bot 4 robotic home assistant
Source: Fraunhofer IPA: `www.care-o-bot.de/en/`. Used with permission

more flexible, autonomous robotic assistants not only in the workplace but in our homes. Robotic cleaning devices, such as the Roomba,[1] already exist but it is much more autonomous *Robotic Assistants* that are now being designed and developed. Initially intended for the elderly or incapacitated, we can expect to see such robots as domestic assistants in our homes quite soon (see Figure 1.3).

[1] `www.irobot.com/uk/Roomba`.

https://doi.org/10.1017/9781108755023.001 Published online by Cambridge University Press

As we get towards this stage, and as these robotic assistants are required to exhibit increasing levels of autonomy, we might see these robots less like 'servants' and more as 'friends' or 'team-mates'! It is not surprising, therefore, that there is considerable research into *human–robot teamwork*; not only how to facilitate such teamwork, but how to ensure that the team activity is effective and reliable. It is recognised that sophisticated human–robot interaction scenarios will be with us very soon.

1.3 Why use *Formal* Verification?

While there are many potential applications for autonomous systems, few current systems involve full autonomy. Why? Partly this is because the regulatory frameworks often do not *permit* such systems to be deployed; partly this is because we (developers or users) do not *trust* such systems. In both these cases, the ability to *formally verify* properties of an autonomous system will surely help. Mathematical proof that a system has certain 'safe' behaviour might be used in certification or regulation arguments; the certainty of such proofs can also help alleviate public fears and provide designers with increased confidence.

1.3.1 What is *Verification*?

In this book we will distinguish between validation and verification. We use the term *validation* for a process that aims to ensure that whatever artefact we have produced meets the 'real' world needs we have. We will use *Verification*, on the other hand, to refer to any process used to check that the artefact matches our specification or requirements. Such verification might often be carried out using methods such as *testing* or *simulation*.

1.3.2 Formal Approach

On the other hand, *formal* verification utilises strong mathematical techniques, particularly logical proof, to assess the system being produced against its specification. One or both of the system and specification may be described using formal logic and then formal verification will attempt to show either that the system satisfies the given specification or, if it does not, provide an example system execution that violates the specification. There are a wide range of formal verification techniques, from formal proof carried out by hand through to automated, exhaustive exploration of the execution possibilities of the system. It is the latter type that we consider in this book, specifically in Chapters 4 and 6.

https://doi.org/10.1017/9781108755023.001 Published online by Cambridge University Press

1.3.3 Why?

Formal verification is difficult and time-consuming, and formal verification techniques are not at all common in the development of autonomous systems. So, why bother? What do such techniques give us that is important enough to expend all this effort on?

As we have seen, autonomous systems are on the increase, and are set to be widespread in the future. One place where formal verification is widely used is in the development of safety-critical systems, particularly in aircraft where its use is often required by regulators. Unsurprisingly, we can anticipate that in order to be *legal* some of these systems may need to have been verified.

The use of formal verification techniques, with their strong mathematical basis, can not only provide formal evidence for certification and regulation processes, but can potentially increase the public's confidence and *trust* in these systems (Chatila et al., 2021). More generally, formal verification gives us much greater certainty about the decision-making that is at the heart of autonomous systems. Before these systems came along, engineers were principally concerned with the questions 'what does the system do?' and 'when does the system do it?'. But, now, we must address the key question relating to autonomy: 'why does the system choose to do this?'. As we will see later, having viable formal verification techniques impacts upon a wide range of areas: safety, ethics, certification, and so on.

In the following chapters, we will describe *hybrid agent architectures* where high-level decision-making in an autonomous system is carried out by a type of software component referred to as a *cognitive agent*. This architecture and the use of cognitive agents enable us to perform formal verification of the system's decision-making.

https://doi.org/10.1017/9781108755023.001 Published online by Cambridge University Press