2015

Merging Two Worlds: Agent-Based Simulation Methods for Autonomous Systems

Andreas Tolk
Old Dominion University, atolk@odu.edu

Original Publication Citation

Tolk, A. (2015). Merging two worlds: Agent-based simulation methods for autonomous systems. In A. P. Williams & P. D. Scharre (Eds.), *Autonomous Systems: Issues for Defence Policymakers* (pp. 291-317). Norfolk, VA: North Atlantic Treaty Organisation Headquarters Supreme Allied Commander Transformation.

13

Merging Two Worlds: Agent-based Simulation Methods for Autonomous Systems

Andreas Tolk

Abstract

This chapter recommends the increased use of agent-based simulation methods to support the design, development, testing, and operational use of autonomous systems. This recommendation is motivated by deriving taxonomies for intelligent software agents and autonomous robotic systems from the public literature, which shows their similarity: intelligent software agents can be interpreted as the virtual counterparts of autonomous robotic systems. This leads to examples of how simulation can be used to significantly improve autonomous system research and development in selected use cases. The chapter closes with observations on the operational effects of possible emergent behaviour and the need to align the research agenda with other relevant organisations facing similar challenges.

Introduction

Modelling and simulation (M&S) is well known and often applied in NATO. Although mostly used in the training domain in the form of computer-assisted exercises, the NATO M&S Master Plan identifies five application areas that can capitalise on M&S: support to operations, capability development, mission rehearsal, training and education, and procurement.(1) This chapter therefore explores how M&S can best be used to support the various capacities of autonomous systems.

In their application-focused overview of M&S paradigms, Hester and Tolk describe the broad spectrum of M&S approaches:(2)

1. NATO 2012.
2. Hester and Tolk 2010.

- *Monte Carlo simulation (*the method of repetitive trials): This paradigm uses probabilistic models of usually static systems, and is used to evaluate systems that are analytically untraceable (such as those that cannot easily be described by mathematical functions).
- *Systems dynamics*: This paradigm is used to understand the behaviour of nonlinear, highly interconnected systems over time. It uses internal feedback loops, flows with time delays, and stocks and piles to model the system; systems are usually described using a top-down approach.
- *Discrete event simulation*: This paradigm is used for the dynamic simulation of systems in which the states are changing instantaneously when defined events occur. Event, time, and state change have to be defined precisely.
- *Continuous simulation*: This paradigm is used for the dynamic simulation of systems in which the states are changing continuously over time. They are normally described by differential equations that have to be approximated numerically.
- *Agent-based simulation*: The 'agent' metaphor uses agents as 'intelligent objects' that build a system from the bottom up, using agents to define the components of the systems. Agents perceive and act within their situated environment to reach their goals. They communicate with other agents following a set of rules. By adapting their rules to new constraints in the virtual environment, software agents can 'learn'.

It is worth mentioning that system dynamics implements a typical top-down design approach, while agent-based models are more useful for building systems from the bottom up based on component descriptions. Discrete event simulation supports both approaches, but traditionally is used more often to implement top-down solutions.

While all M&S paradigms can provide some support to autonomous systems, the agent-based simulation paradigm is of particular interest, as autonomous systems reflect characteristics similar to those of software agents. Autonomous robotic systems(3) are defined by the ability (e.g., by using integrated sensing, perceiving, analysing, communicating, planning, decision making, and acting/

3. The insights derived in this chapter are primarily based on the field of robotics, which is why the term 'autonomous robotic systems' is used. It is reasonable to assume that the results are generalisable, but a formal evaluation has not yet been conducted.

executing) to achieve assigned goals.[4] Intelligent software agents are defined by their ability to perceive their situated environment; socially interact with other agents for planning and executing; act in their environment based on their programmed beliefs, desires, and intentions; observe the results; and adjust their actions based on these results.[5]

The research presented in this chapter not only shows that the characteristics of autonomous robotic systems and intelligent agents are similar, but also that the taxonomies are alike as well. This is important because it means that agent-based simulation methods can be used to support autonomous robotic systems in various application domains. Furthermore, observations of emergent behaviour typical of agent-based systems are likely to be observed in autonomous system populations as well.

Characteristics and Taxonomy

This section describes the characteristics and taxonomy of intelligent software agents, as they are used within agent-based modelling, and of autonomous robotic systems, as they are dealt with in this book. Its goal is to provide the researchers of both domains with a basic understanding of why it is pivotal for them to work together to maximise the benefit for NATO.

Intelligent Software Agents

This section is mainly derived from the contribution of Tolk and Uhrmacher to the seminal work of Yilmaz and Ören.[6] The agent metaphor is a well-researched topic, but the results are distributed among a huge variety of research domains. The metaphor is based in various computer science areas – such as distributed systems, software engineering, and artificial intelligence – and has been strongly influenced by research results from disciplines such as sociology, biology, cognitive sciences, systems sciences, and many others. Although there are many definitions of agents, the following working definition provided by

4. Huang, Messina, and Albus 2003.
5. Yilmaz and Ören 2009.
6. Tolk and Uhrmacher 2009; Yilmaz and Ören 2009.

Tolk and Uhrmacher describes the main characteristics of intelligent software agents:(7)

- *The agent is situated, it perceives its environment, and it acts in its environment.* The environment typically includes other agents, other partly dynamic objects, and passive ones, which are, for example, the subject of manipulation by the agent. The communication with other agents is of particular interest in systems comprising multiple agents, as agents can collaborate and compete for tasks. This latter characteristic has also been referred to as 'social ability'.
- *The agent is autonomous*, in the sense that it can operate without the direct intervention of humans or others; autonomy requires control of its own state and behaviour. Agents must be guided by some kind of value system, which leads to the often-used statement: 'objects do it for free, agents do it for money'.(8) In other words, an object always executes functions that are invoked, while agents can decide if (and how) they react to a request.
- *The agent is flexible*, which means it can mediate between reactive behaviour (being able to react to changes in its environment) and deliberativeness to pursue its goals. A suitable mediation is one of the critical aspects for an agent to achieve its tasks in a dynamic environment. An agent can act upon its knowledge, rules, beliefs, operators, goals, and experiences, etc. and to adapt to new constraints and requirements – or even new environments – as required. For example, new situations might require new goals, and new experiences might lead to new behaviour rules. In other words, an agent can learn. Furthermore, being mobile adds to its flexibility.

The following figure exemplifies these characteristics. It shows an intelligent agent in the centre of its environment. The agent perceives its environment, which includes other agents he can interact *with* and objects he can act *on*. He maps this perception to an internal representation, which may be incomplete,

7. Tolk and Uhrmacher (2009, 77). As a rule, intelligent software agents are virtual entities that exist in software programs. However, this technology is already applied commercially to support intelligent internet-based software agents that support the automatic updating of travel arrangements, recommend new products to customers, etc.

8. Jennings, Sycara, and Wooldridge 1998.

e.g., he only knows about one of the square posts and nothing about the triangular-shaped object. Objects can be static or expose dynamic behaviour, like the ball that will roll once it is kicked. Advanced agents may use simulation to act on their perception to support their decision-making process; for example, they may simulate each alternative and apply measures of merit that reflect their goals, desires, and beliefs onto the projected result and select the alternative with the highest expected value. An agent communicates with other agents and acts on the objects, such as kicking the ball in the direction that the other agent is running. If the plan does not work as expected, the agent will learn from his observations that this action does not lead to the desired outcome, and he will choose other options in the future. If something works well, he will use this strategy more often. Holland describes several learning algorithms that can be applied, and many more have been developed and successfully applied since then.(9) Some agents use game theory approaches to select a mixed strategy based on the expected pay-offs of possible alternatives.(10)

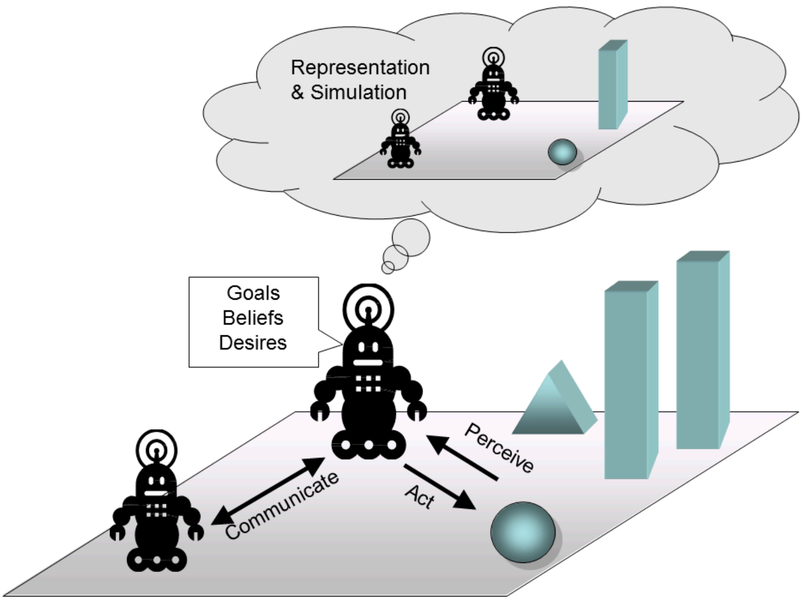

Figure 13.1. Intelligent Software Agents in the Situated Environment

9. Holland 1986.

10. Parsons and Wooldridge 2002.

In order to implement these characteristics, several architectural frameworks were recommended in the agent-based literature, many of which focus on the particular area of interest. Moya and Tolk 2007 composed the various ideas into a common architectural framework that captures the taxonomy of a single intelligent software agent.[11] This taxonomical structure was kept simple and adaptable in order to support as many viewpoints as possible. It identifies three external domains and four internal domains.

The three external domains comprise the functions needed within an agent to interact with the situated environment, which includes objects and other agents:

1. The *perception domain* observes the environment. Using its sensors, the agent receives signals from his environment and sends this information to the internal sense-making domain.
2. The *action domain* comprises the effectors. If the agent acts in his environment, the necessary functions are placed here. It receives the task to perform tasks from the internal decision-making domain.
3. The *communication domain* exchanges information with other agents or humans. If it receives information, it is sent to the internal sense-making domain. It receives tasks to send information from the internal decision-making domain.

The four internal domains categorize the functions needed for the agent to decide how to act and adapt as an autonomous object (see Figure 13.2):

1. The *sense-making domain* receives input (via sensors and communication) and maps this information to the internal representation that is part of the memory domain. These domains comprise potential data correlation and data fusion methods; data mediation capabilities; methods to cope with uncertain, incomplete, and contradictive data, etc.
2. The *decision-making domain* supports reactive as well as deliberative methods, as they have been discussed in this chapter. It uses the information stored in the memory domain and triggers communications and actions.
3. The *adaptation domain* may be connected with perception and action as well, but that is not a necessary requirement. The comprised function

11. Moya and Tolk 2007.

group updates the information in the memory domain to reflect current goals, tasks, and desires.

4. The *memory domain* stores all information needed for the agent to perform his tasks. It is possible to distinguish between long-term and short-term memory, and different methods to represent knowledge can be used alternatively or in hybrid modes.

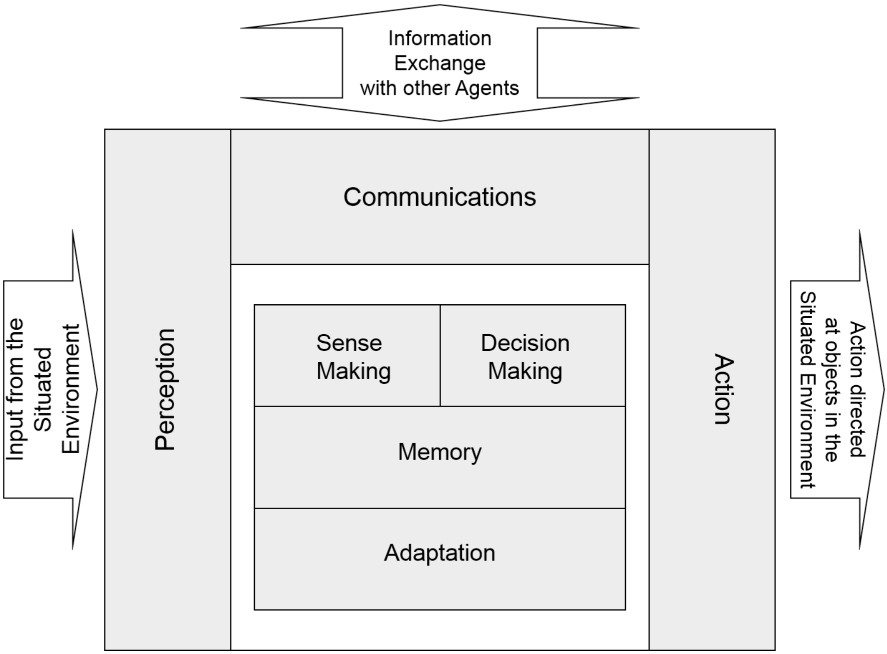

Figure 13.2. General Taxonomy of an Intelligent Software Agent

A complete agent taxonomy for agent-based simulation needs to reflect not only on the individual agents, but also on the characteristics of the situated environment as well as of the agent society. Wooldridge identifies five categories to characterize the environment for an intelligent software agent:[12]

1. *Accessibility*. The environment can be *accessible or non-accessible*. This category addresses how much of its attributes the environment exposes.

12. Wooldridge 2000.

This is different from the question of how the agent perceives what is exposed, and whether he can make sense of what he perceives.

2. *Determinacy.* The environment can be *deterministic or stochastic*. In deterministic environments, an action always has the same deterministic effect. In stochastic environments, this is not the case. For all practical purposes, the 'real world' can be assumed to be stochastic in nature.
3. *Reactivity.* The environment can be *episodic or sequential*. In episodic environments, the action is only relevant for the current episode. In sequential environments, an action may have effects in future states as well. This includes the idea of 'effects of effects' that often take some time to be exposed. The 'real world' is sequential in nature.
4. *Degree of change.* The environment can be *static or dynamic*. A static environment does not change during the evaluation period; a dynamic environment does. The real world is dynamic in principle, but in many practical applications is static for this particular application.
5. *Type of change.* The environment can be *discrete or continuous*. Furthermore, discrete environments can differ in the level of resolution, accuracy, and granularity. This category directly connects back to the modelling paradigm used to provide the environment for the software agents.

In their analysis of different collections of agents and how they act in society, Moya and Tolk identified the size (number of agents) and diversity (type of agents) as the driving categories in the literature dealing with agent societies.[13] To cope with all observations, two additional categories – social interactions and openness – were proposed to characterize the agent societies built by intelligent software agents more generally.

- *Size*. The number of agents within the population can vary between large-scale numbers of several thousand agents down to a few agents. In some cases, only a single agent is used, although this is the exception.
- *Diversity.* The society can comprise agents that are all of the same type, building a homogeneous society, or agents of different types, building a heterogeneous society. It is worth mentioning that even agents of the

13. Moya and Tolk 2007.

same type can exhibit different behaviour depending on their state and initialization.

- *Social interactions*. The agents can either cooperate with each other or be in competition. In addition, all mixed forms are possible, such as coalitions that cooperate with each other but compete with others. It is also possible that agents are agnostic and that every agent follows its own objectives without any interaction.
- *Openness*. The agent society can be open or closed. In open societies, anyone can contribute agents and add them to the society. In closed societies, the number of contributors is limited by constraints. The contributors can be human, other agents, or systems. As before, various mixed forms are possible.

The resulting agent taxonomy, which reflects all characteristics of agenthood regarding the agent, the situated environment, and the agent society, is exemplified using the top-level categories in Figure 13.3.

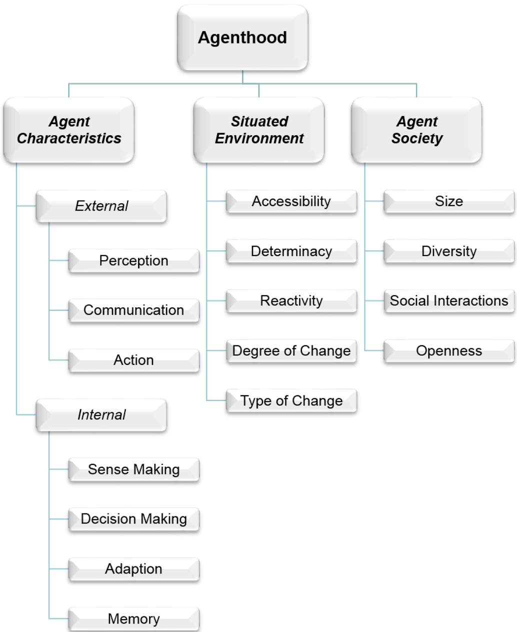

Figure 13.3. Taxonomical Components Describing Agenthood

The viewpoint of this summary of characteristics and taxonomy of intelligent software agents is biased toward the applicability of agent-based modelling methods in support of autonomous robotic systems, in particular to show the similarities.[14]

Autonomous Robotic Systems

The characteristics and taxonomical structures presented in this section are simplified to improve easier understanding of the mapping potentials between components of the agent-based simulation domain and the domain of autonomous robotic systems.

As with intelligent agents, a multitude of heterogeneous application domains and supporting disciplines contributed to the definition of autonomous robotic systems. A common understanding of autonomy is that the system has the capability to make decisions about its actions without the involvement of an operator, which also entails entrusting the system to make these decisions. This is far more than automation, which involves using control systems and information technology to reduce the need for human intervention within well-defined constraints.

The Autonomy Level for Unmanned Systems project by the National Institute of Standards and Technology defines autonomy as:

> The condition or quality of being self-governing. When applied to unmanned autonomous systems (UAS), autonomy can be defined as UAS's own ability of integrated sensing, perceiving, analysing, communicating, planning, decision-making, and acting/executing, to achieve its goals as assigned by its human operator(s) through designed human-robot interface (HRI) or by another system that the UAS communicate with.[15]

Autonomous robotic systems are an example of UAS. Their characteristics are close to the working definition presented for intelligent software agents above. The main difference is that software agents are virtual actors in a virtual

14. For more detailed information, see Yilmaz and Ören 2009.
15. Kendoul 2013.

world, while robots are physical actors in the physical world. The following list has been compiled to clarify the similarities of intelligent software agents and autonomous robotic systems.

- *The autonomous robotic system is situated, it perceives its environment, and it acts in its environment.* The environment typically includes other autonomous robotic systems and objects that are subject to manipulation by the system. In scenarios with multiple robots, they can collaborate and compete for tasks. Therefore, autonomous robotic systems often have the ability to communicate with each other.
- *The robotic system is autonomous* in the sense that it can operate without the direct intervention of humans or others; autonomy requires control of its own state and behaviour. As a rule, it needs to have a plan that is often assigned by one or several human operators.
- *The autonomous robotic system is flexible.* If the observation shows that current actions do not lead to the desired effects, the autonomous robotic system can identify and execute alternatives. Advanced systems may even create a new plan together. Mixed strategies between immediate reactive behaviour and more time-consuming deliberate behaviour ensure that the system is safe in critical environments.
- *The autonomous robotic system is mobile,* in the sense that it can move in the environment within the physical constraints imposed on it. Eventually, additional rules of engagements may set more constraints than just the physical ones.

The resulting taxonomy presented here is a simplified aggregate of the ideas presented in Matarić's and Siegwart, Nourbakhsh, and Scaramuzza's seminal works on robotics and autonomous mobile robots.(16) The following components comprise the taxonomy of an autonomous mobile robotic system:

- The *locomotion component* moves the system in its environment, and is constrained by the different degrees of freedom. These can be tracks or wheels, but also the rotors of a helicopter or quadrocopter, etc.

16. Matarić 2007; Siegwart, Nourbakhsh, and Scaramuzza 2011.

- *Actuator components* are moving parts of the autonomous robotic system, such as robot arms, sensors, antennas, etc., which are used in order to act on things, perceive better, etc.
- *Manipulation components* interact with objects and the environment. They grab, push, turn, and do whatever is needed to act on the environment to conduct actions according to the plan.
- *Sensor components* observe the environment. They are the eyes and ears of the robot. They can be passive or active. They can be as easy as switches operated by bumpers, or they can be lasers and sonars or complex cameras.
- *Signal processing components* are used to convert sensor signals into computable information. They are sometimes seen as components of the sensor, but Matarić points out that these components are also used to convert computed information into actuator signals. As such, they are sitting between the eyes, ears, and arms of the robot, and its brain.
- The *control component* is the 'brain' of the autonomous system. It makes the decisions based on the perception created from the input of the sensors and the plan the robot is following. As a rule, the control component is a computer.
- *Communication components* exchange information with other robots, as well as with humans, via HRI. The variety of communication components is as big as that of sensors, but they all serve the same purpose: allowing the control component of the robot to exchange information with other entities.
- Of critical importance are the *power supply components*, as they are the energy source for all actions. Usually, these are batteries or solar panels, but alternatives are possible as well, depending on the size and the tasks of the robot.

Figure 13.4 displays the components using the structure of the intelligent agent taxonomy as a guide.

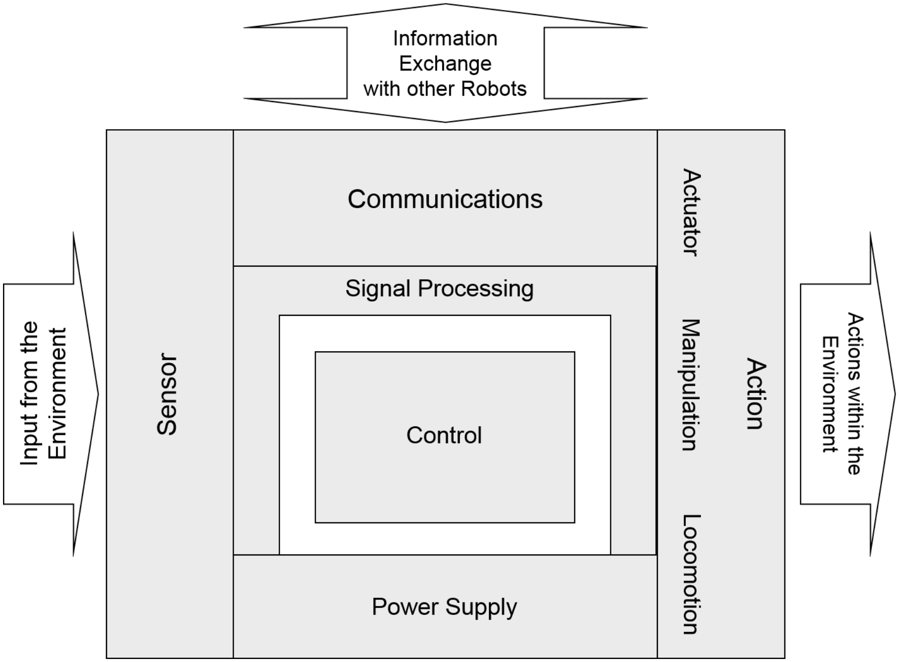

Figure 13.4. General Taxonomy of an Autonomous Robotic System

There are, without a doubt, many differences between agents and robots. Agents live in the virtual world, while robots live in the physical world. As such, robots have many tasks regarding locomotion, perception, localization, and navigation that the agents do not have to address in the virtual world. A lot of research has therefore been directed at better sensors, better locomotion and manipulator components, and other components that are necessary to make a robot work in the physical environment. As a result, the external domains are dealt with in greater detail in the robotics domain than in the agent domain. Many robotics practitioners even regard simulation as inferior, stressing that there is a huge reality gap between the needs of robotics and the contribution capability of simulation.

In contrast to such perceptions, the focus in this chapter shifts toward domains in which intelligent agent research already provides results that are directly applicable to improving autonomous robotic systems. The social ability of agents; the ability to learn; and the algorithms developed, implemented, and tested regarding sense making, decision making, and other components of the interior domain can support more intelligent robotics behaviour. Furthermore, the simulated environment of agents is a safe and inexpensive test bed for the intelligent behaviour of autonomous robotic systems as well.

The taxonomical similarity between intelligent software agents and autonomous robotic systems allows the identification of these areas of mutual support. In the next section, several application domains are identified that can be immediately utilized to improve the behaviour of robots by applying methods from the agent paradigm.

Applying the Agent Paradigm in Support of Autonomous Robotic Systems

This section is neither complete nor exclusive. The objective is to give three examples of very different application domains showing the synergy of intelligent software agents, M&S, and autonomous robotic system research. The first example shows the synergy in the domain of test and evaluation, starting earlier in the procurement phase, where first testing is possible using only virtual prototypes of an envisioned autonomous robotic system. The second example shows how sense making and machine understanding, as used in intelligent software agents, can be used successfully to improve the sensing and perceiving activities needed in autonomous robotic systems as well. The third topic shows the relevance of research results in the domain of intelligent software agents for operational experts who are interested in using autonomous robotic systems.

Developing and Testing Autonomous Robotic Systems

Many simulation publications give examples of using simulation as a test bed to stimulate systems under test, some of them more than a decade ago, such as McKee.(17) The US Army launched the Simulation and Modelling for Acquisition, Requirements and Training initiative to support this domain,(18) which spawned several follow-on activities in the other services as well as in

17. McKee 1998.
18. Page and Lunceford 2001.

NATO. Neugebauer et al. present related work on a distributed test bed based on simulation services.[19]

Today, the idea of using simulation to provide stimulation for a system under test is well established and often applied. However, autonomous robotic systems are posing new challenges for the test-and-evaluation community. In addition to exposing a great variety of requirements from many stakeholders, the main challenge is that autonomous robotic systems operate in a dynamic and unpredictable environment, and onsite testing of the total feature set of a new system under realistic operational conditions is impractical.

Agent-based test environments can help address these challenges in two ways. First, they allow the behaviour of an autonomous robotic system to be tested using its virtual counterparts before it is implemented within the physical robot. An intelligent software agent can follow the same rule sets in the virtual environment that the autonomous robotic system would follow in the real environment. Its sensors can be simulated, as can its interactions with objects. This idea is not new, as documented by Akin et al. using the example of the RoboCup Rescue Robot and Simulation Leagues.[20] Many of these simulated rescue robots can be programmed in the same programming language and with the same tools that the real robots will be using later on. The Defense Advanced Research Project Agency Robotic Challenge uses the same approach. Aleotti et al. envisioned such an approach early on.[21]

The author supported the US Navy with research on the 'Riverscout', an unmanned surface water vehicle that demonstrates some autonomous behaviour as well. Figure 13.5 shows various simulation screen shots as well as the real system in a test. Some additional information has been published by Barboza.[22]

The second way in which agent-based models support the testing is by providing a smart and adaptive test environment. The test environment is not just a script-driven stimulation provider that allows the researcher to systemically provide all sorts of inputs; it actually reacts meaningfully to the actions of the system. In other words, agent-based models realistically replicate the dynamic and unpredictable operational environment for the test. Furthermore,

19. Neugebauer, Nitsch, and Henne 2009.
20. Akin et al. 2013.
21. Aleotti, Caselli, and Reggiani 2004.
22. Barboza 2014.

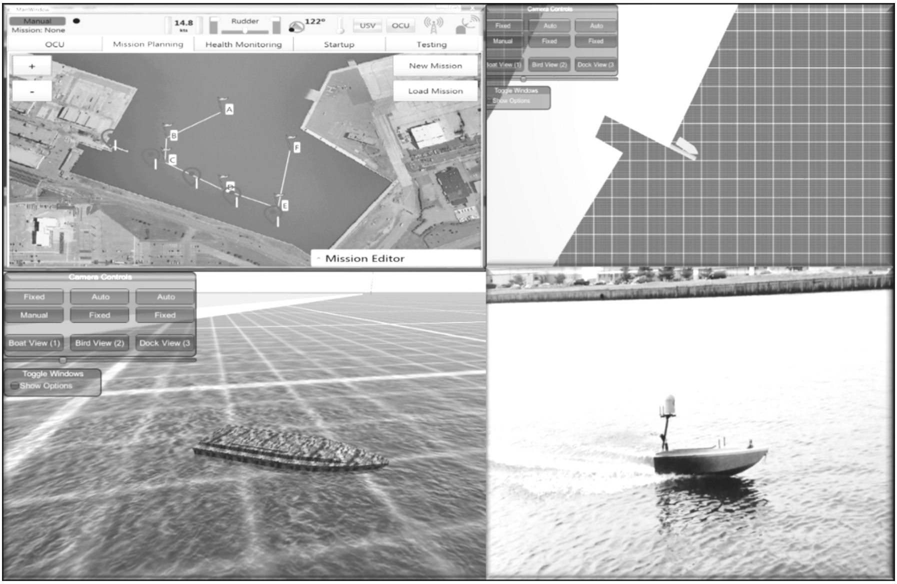

Figure 13.5. Simulation Support for the Unmanned Surface Vehicle 'Riverscout'

they offer the other autonomous robotic systems that are needed to provide the full operational functionality within a scenario realistically. Every system is represented by intelligent software agents. If the ideas of executable architectures based on operational specifications (as provided by common system architectures) are applied, as described in more detail by Garcia and Tolk, each intelligent software agent can take over the role of a system with the envisioned portfolio to provide the most realistic operational conditions for testing, even if not all systems of the portfolio are physically available.[23]

A major challenge for agents representing human actors has been to elicit expert knowledge in a form that agents can use. Hoffman et al. showed that Applied Cognitive Task Analysis and critical decision methods allow for the creation of a cognitive representation for agents within the virtual environment.[24] Garrett conducted a series of experiments to show their applicability for

23. Garcia and Tolk 2013.

24. Hoffman, Crandall, and Shadbolt 1998.

intelligent software agents.[25] This research is highly relevant when autonomous robotic systems are supposed to be surrogates for human experts.

These last two paragraphs also emphasize that the behaviour of an intelligent software agent shall never be rooted in interpretations of a software developer, but driven by experts' insights and operationally validated artefacts. This insight becomes even more important when the methods are applied to implement intelligent behaviour for robots, as they act (and interact) in the physical world, and failures and wrong decisions may have dangerous – and even deadly – consequences. The autonomous robot system community should therefore carefully analyse the lessons learned from the intelligent software agent community.

The first systems using related technologies are already in operational-like use. The Control Architecture for Robotic Agent Command and Sensing project conducted by the Office of Naval Research evaluates swarm technology, which is a sub-set of agent-based technology, to control a set of autonomous surface vessels to protect selected ships. The sensor and software kit can be transferred between small vessels that are under human control, but follow simple rules that all contribute to a new capability to better protect ships. Similar approaches have been successfully tested for search and rescue operations.

Using Agent Methods to Implement Intelligent Behaviour

Even in their seminal book on autonomous mobile robots, Siegwart et al. explicitly state that the focus lies on mobility.[26] Comparing the two taxonomical structures presented in Figures 13.2 and 13.4 also shows that the focus of agents is the sense making and decision making to act meaningfully in the virtual world, while the focus of robots is more geared toward interacting with the physical world. Again, this represents a possibility to create synergy by bringing both worlds together and using intelligent software agent methods to enable autonomous robotic systems to expose the same degree of sense making, decision making, memory, and adaptation as agents do.

25. Garrett 2009.

26. Siegwart, Nourbakhsh, and Scaramuzza 2011.

An applicable lesson learned goes back to Zeigler's work on how machines gain understanding.[27] This model will be explained here in a slightly modified form. Nearly 60 more types of machine understanding, many of them also applicable to this chapter, have been evaluated by Ören et al., and many have been successfully applied within intelligent software agents.[28] They provide a rich body of knowledge that the autonomous robotic system community can draw from. In order for a machine to understand, four premises have to be fulfilled, as they are visualized in Figure 13.6:

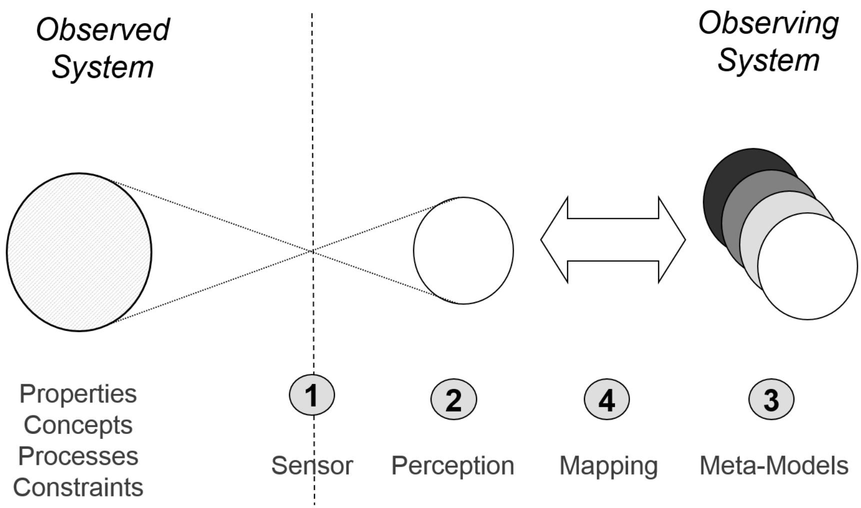

Figure 13.6. Using Meta-Models and Mappings in Support of Machine-based Sense-making

1. The machine must have sensors to observe its environment. The type of sensor defines which attributes can be observed and which properties and processes can be recognized if the target-noise ratio between the observed system and the situated environment is high enough.[29]

27. Zeigler 1986.
28. Ören et al. 2007.
29. Some general sensor modelling constraints have been documented in Tolk (2012).

2. The machine has a perception of the system to be understood. The properties used for the perception should not significantly differ in scope and resolution from those exposed by the system under observation. They are often closely coupled with the sensor's abilities as well as with the machine's computational abilities.
3. The machine has meta-models of observable systems, which are descriptions of properties, processes, and constraints of the expected behaviour of observed systems. Understanding is not possible without such models. They can be understood as types of systems that may be observed in the situated environment.
4. The machine can map the observations, resulting in the perception of a suitable meta-model explaining the observed properties, processes, and constraints.

Understanding involves pairing the perception with the correct meta-model. If such a model does not exist, a sufficiently similar model can be used that explains at least part of the behaviour. The learning algorithms mentioned above can then be applied to adapt existing models or to create new models that can be applied to future observations.

This principle is not too far from how humans gain knowledge. When primitive peoples first make contact with higher civilizations, they often use familiar terms to address new concepts that are 'close enough'. Examples may be 'giant metal birds' when addressing airplanes, or 'giant locusts with human faces prepared for battle' when describing attack helicopters with pilots in their cockpits. The concept of a bird addresses the ability to fly, but one of the properties is very different from birds, as they normally are not made out of metal. Once enough information is collected, new concepts can be created to deal with the observed system, so that it can be recognized.

Another aspect of interest is the possibility of capturing desires and beliefs in machine-understandable form. Harmon et al. documented canonical structure to allow mimicking the behaviour of both single human beings and collectives.[30] Again, this knowledge can be applied to make autonomous robotic systems act 'more human' if this is in the objectives of the development.

30. Harmon et al. 2001.

Emergence and Operational Implications

Intelligent software agents are often connected with emergence. Maier distinguishes in the context of system-of-systems engineering between weak, strong, and 'spooky' emergence.[31] Looking at this mainly from the systems perspective (not as an artificiality of the representing model or software), he defines the terms as follows:

> *Weak Emergence*: An emergent property that is readily and consistently reproduced in simulations of the system, but not in reduced complexity non-simulation models. It can be understood through reduced complexity models of the system after observation, but not consistently predicted in advance.
>
> *Strong Emergence*: An emergent property that is consistent with the known properties of the system's components but which is not reproduced in any simplified model of the system. Direct simulations of the system may reproduce the emergent property but do so only inconsistently and with little pattern to where they do so and where they fail. Reduced complexity models or even simulations do nor reliably predict where the property will occur.
>
> *Spooky Emergence*: An emergent property that is inconsistent with the known properties of the system's components. The property is not reproduced in any model of the system, even one with complexity equal to that of the system itself, even one that appears to be precisely simulating the system itself in all details.[32]

A well-known example of the general emergence of system behaviour that results from simple rules between the implementing components is Schelling's segregation model.[33] It has been implemented multiple times to show how people use very simple rules to select their neighbourhood. This decision is based on a preferred mix between two population groups. On the system level, segregation patterns emerge without any implicit or explicit formulation of such a property on the component level.

31. Maier 2014.
32. Ibid., 22
33. Schelling 1969.

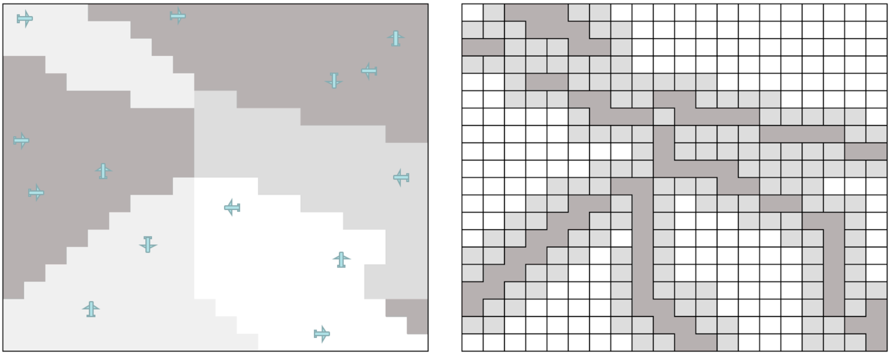

Figure 13.7. Emergent Observation Patterns and Gaps

Bonabeau's seminal paper on using agent-based modelling to represent self-deciding entities made a strong connection between emergent behaviour in the system and such entities.[34] While accepted in social systems, such emergent behaviour is often not wanted in technical systems, such as societies of autonomous robotic systems. Nonetheless, as the system composition is analogous to that of an agent-based population, emergent properties are more likely to be observed.

The following figure shows an example implemented in Netlogo, which is an open system developed by the Center for Connected Learning at Northwestern University.[35] The left figure shows six zones in which agents can move. The rule for the agents is to move around the zone using an equal distribution to cover all ground, but to avoid colliding with other agents in the same zone and to keep a safe distance from agents in a neighbouring zone. An operational interpretation could be drones in the air zones of coalition partners that want to observe their areas of responsibility. When counting the number of times an agent visits another agent, the right picture emerges: there are some islands close to the border between the areas of responsibility that are rarely visited, although no rule excluded these areas from observation.

In an operational context, such areas can easily become safe havens for activities that were supposed to be observed in the first place. While this example is trivial and easy to fix, it demonstrates the underlying challenge: how can a

34. Bonabeau 2002.
35. Tisue and Wilenski 2004.

commander support positive emergence that helps reach his objective while avoiding negative emergence that is counterproductive?

A first step is to raise awareness that emergence in systems comprising interacting autonomous robotic systems will occur. It is not an artificiality. It will happen, and it needs to be controlled in the context of the operational objectives. A second step is to conduct more research on whether – and to what degree – it is possible to engineer positive emergence into such systems and avoid negative emergence. Although initial results have been published, and some answer may even be found in the fundamentals of cybernetics as described by Ashby, the topic itself is still in its infancy.(36) Tolk and Rainey are making it a priority in their recommended research agenda.(37) The engagement of the autonomous robotic system community is highly encouraged, as the results of this research are directly applicable to societies of autonomous systems as well.

Summary

In the only recently released report on this topic,(38) internationally recognized experts recognised the perpetually increasing importance of modelling and simulation for the design, development, testing, and operation of increasingly autonomous systems and vehicles for a wide variety of applications on the ground, in space, at sea, and in the air. Section 4 of the report identifies the four most urgent and most difficult research projects:

1. behaviour of adaptive/non-deterministic systems;
2. operation without continuous human oversight;
3. modelling and simulation; and
4. verification, validation, and certification.

All four research domains are at least touched on this chapter: (1) agents are virtual prototypes that can be used to evaluate the behaviour, (2) establishing robust rules and procedures allows the operation to be conducted without human oversight, (3) the use of modelling and simulation is an important area of research, and (4) verification, validation, and certification are an

36. Ashby 1963.
37. Tolk and Rainey 2014.
38. NRC 2014.

important aspect that justifies the use of agent-based test beds. This shows not only the necessity of related research, as recommended in this chapter, but it also demonstrates the potential of fruitful collaboration in this domain with other research agencies. Many civil agencies and government organisations are interested in these questions. The results of their efforts should be observed, as it is likely that many of their key findings can help answer the urgent NATO research questions.

The domain of intelligent software agents has an enormous potential to enrich the research on autonomous robotic systems. It can (and should) be applied in the domain of procurement:

1. test and evaluation – testing robotic behaviour in a virtual environment before the robot is built;
2. providing a flexible and intelligent test bed for autonomous systems, as traditional test and evaluation methods are insufficient to test systems' autonomous characteristics;
3. using established methods and algorithms to enable the learning and adaptability of robots, as intelligent software agents are well known for their ability to learn and adapt to new situations (and the topological similarity shown in this chapter implies that many ideas can be mapped and reused); and
4. operational support – from building awareness of new challenges and demonstrating new capabilities to evaluating emergent behaviour in the operational context.

Both communities should actively engage to develop synergisms and potentially multi-disciplinary collaboration with NATO, which has the necessary structures for common research and presentation in place. Hopefully, this chapter will help establish a common research agenda and encourage mutual benefit from results that are already available.

References

Aleotti, J., Caselli, S., and Reggiani, M. 'Leveraging on a Virtual Environment for Robot Programming by Demonstration'. *Robotics and Autonomous Systems* 47/2 (2004): 153–61.

Akin H.L., Ito, N., Jacoff, A., Kleiner, A., Pellenz, J., and Visser, A. 'RoboCup Rescue Robot and Simulation Leagues'. *AI Magazine* 34/1 (2013): 78–86.

Ashby, W.R. *Introduction to Cybernetics*. Hoboken, NJ: Wiley, 1963.

Barboza, J. *Conversion Challenges of an Autonomous Maritime Platform: Using Military Technology to Improve Civilian Security*. Paper presented at the MODSIM World Conference & Expo 2014, Hampton, VA, 15–17 April 2014.

Bonabeau, E. 2002. 'Agent-based Modeling: Methods and Techniques for Simulating Human Systems'. *Proceedings of the National Academy of Sciences of the United States of America* 99/3 (2002): 7280–7.

Garcia, J.J., and Tolk, A. 'Executable Architectures in Executable Context enabling Fit-for-Purpose and Portfolio Assessment'. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology* (2013), 10.1177/1548512913491340.

Garret, R.B. 2009. 'A Method for Introducing Artificial Perception (AP) to Improve Human Behaviour Representation (HBR) Using Agents in Synthetic Environments'. Ph.D. Dissertation. Norfolk, VA: Old Dominion University.

Harmon, S.Y., Hoffman, C.W.D., Gonzalez, A.J., Knauf, R., and Barr, V.B. 'Validation of Human Behavior Representations'. In *Foundations for V&V in the 21st Century Workshop*, 22–4, Laurel, MD, 2002.

Hester, P.T., and Tolk, A. 2010. 'Applying Methods of the M&S Spectrum for Complex Systems Engineering'. Paper presented at the Emerging Applications of M&S in Industry and Academia Spring Simulation Multi-Conference, 17–24, 11–15 April, Orlando, FL.

Hoffman, R.R., Crandall, B., and Shadbolt, N. 'Use of the Critical Decision Method to Elicit Expert Knowledge: A Case Study in the Methodology of Cognitive Task Analysis'. *Human Factors* 40/2 (1998): 254–76.

Holland, J. H. 'Escaping Brittleness: The Possibilities of General Purpose Learning Algorithms Applied to Parallel Rule-based Systems'. In *Machine Learning: An Artificial Intelligence Approach,* Vol. II, edited by Michalski, Carbonell, and Mitchell, 593-623. Los Altos, CA: Morgan Kaufmann, 1986.

Huang, H. M., Messina, E., and Albus, J. *Autonomy Level Specification for Intelligent Autonomous vehicles.* Gaithersburg, MD: National Institute of Standards and Technology, 2003.

Jennings, N. R., Sycara, K., and Wooldridge, M. 'A Roadmap of Agent Research and Development'. *Autonomous Agents and Multi-Agent Systems* 1/1 (1998): 7–38.

Kendoul, F. 2013. 'Towards a Unified framework for UAS Autonomy and Technology Readiness Assessment (ATRA).' In *Autonomous Control Systems and Vehicles: Intelligent Unmanned Systems*, edited by Nonami, K., Kartidjo, M., Yoon, K.-J., and Budiyono, A. 55–72. Tokyo: Springer, 2013.

Maier, M. W. 'The Role of Modeling and Simulation in System-of-Systems Development'. In *Modeling and Simulation Support of System-of Systems Engineering Applications*, edited by Rainey, L. and Tolk, A., 11–41. Hoboken, NJ: Wiley, 2014.

Matarić, M. J. *The Robotics Primer.* Cambridge, MA: MIT Press, 2007.

McKee, L. 1998. 'Latency and Its Effects on the Fidelity of Air-to-Air Missile T&E Using Advanced Distributed Simulations'. *Joint Advanced Distribution Simulation/Joint Test and Evaluation*, Albuquerque, NM.

Moya, L. J., and Tolk, A. 'Towards a Taxonomy of Agents and Multi-agent Systems'. In *Proceedings of the 2007 Spring Simulation Multi-Conference*-Volume 2, 11–18. Society for Computer Simulation International, San Diego CA, 2007.

National Research Council (NRC). *Autonomy Research for Civil Aviation: Toward a New Era of Flight.* Aeronautics and Space Engineering Board. Washington, DC: National Academic Press, 2014.

Neugebauer, E., Nitsch, D., and Henne, O. *Architecture for a Distributed Integrated Test Bed*. NATO RTO Modelling and Simulation Group Symposium (MSG-069), RTO-MP-MSG-069, paper 19. Brussels, 2009.

North Atlantic Treaty Organization (NATO). *NATO Modelling and Simulation Master Plan – NATO Modelling and Simulation Implementation Plan Version 2.0.* Document AC/323/NMSG(2012)-016, Brussels, 2012.

Ören, T. I., Ghassem-Aghaee, N., and Yilmaz, L. 'An ontology-based dictionary of understanding as a basis for software agents with understanding abilities.' In *Proceedings of the 2007 Spring Simulation Multi-Conference*-Volume 2, 19–27. Society for Computer Simulation International, San Diego CA, 2007.

Page, E. H., and Lunceford, W. H. 'Architectural Principles for the US Army's Simulation and Modeling for Acquisition, Requirements and Training (SMART) Initiative'. *Proceedings of the Winter Simulation Conference*, IEEE, Piscataway, NJ, 2 (2001): 767–70.

Parsons, S., and Wooldridge, M. 'Game Theory and Decision Theory in Multi-agent Systems'. *Autonomous Agents and Multi-Agent Systems* 5/3 (2002): 243–54.

Schelling, T. C. 'Models of Segregation'. *The American Economic Review* 59/2 (1969): 488–93.

Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D. *Introduction to Autonomous Mobile Robots*. 2nd edition. Cambridge, MA: MIT Press, 2011.

Tisue, S., and Wilensky, U. 'Netlogo: A Simple Environment for Modeling Complexity'. In *International Conference on Complex Systems*, 16–21, 2004.

Tolk, A., and Uhrmacher, A.M. 'Agents: Agenthood, Agent Architectures, and Agent Taxonomies'. In *Agent-Directed Simulation and Systems Engineering,* edited by Yilmaz and Ören, 75–109. Weinheim, Germany: Wiley-VCH, 2009.

Tolk, A. 'Modeling Sensing'. In *Engineering Principles of Combat Modeling and Distributed Simulation*, edited by Tolk, A., 127–44. Hoboken, NJ: Wiley, 2012.

Tolk, A., and Rainey, L.B. 'Toward a Research Agenda for M&S Support of System-of-Systems Engineering'. In *Modeling and Simulation Support of System-of Systems Engineering Applications*, edited by Rainey and Tolk, 583–92. Hoboken, NJ: Wiley, 2014 .

Wooldridge, M. J. *Reasoning about Rational Agents*. Cambridge, MA: MIT Press, 2000.

Yilmaz, L. and T. Ören. *Agent-Directed Simulation and Systems Engineering.* Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA, 2009.

Zeigler, B.P. 'Toward a Simulation Methodology for Variable Structure Modeling'. In *Modeling and Simulation Methodology in the Artificial Intelligence Era*, edited by Elzas, M.S., Oren, T.I., and Zeigler, B.P., 195–210. Amsterdam: North Holland, 1986.