Verifiable Autonomy and Responsible Robotics

Michael Fisher

Department of Computer Science, University of Liverpool

RoboSoft, London *13th November 2019*

[with help from **very many** people: Louise Dennis; Matt Webster; Marija Slavkovik; Clare Dixon; Alan Winfield; ...]

My Talk in One Slide

Questions I will *try* to address are

- What is our real worry about autonomous robots?
- How much confidence should we have in robots?

  ....and what evidence/justification can/should be provided?

My Talk in One Slide

Questions I will *try* to address are

- What is our real worry about autonomous robots?
- How much confidence should we have in robots?

  ....and what evidence/justification can/should be provided?

Trust in autonomous systems is complex and subjective.

My Talk in One Slide

Questions I will *try* to address are

- What is our real worry about autonomous robots?
- How much confidence should we have in robots?

  ....and what evidence/justification can/should be provided?

Trust in autonomous systems is complex and subjective.

Trustworthiness comprises at least two aspects:

My Talk in One Slide

Questions I will *try* to address are

- What is our real worry about autonomous robots?
- How much confidence should we have in robots?

  ....and what evidence/justification can/should be provided?

Trust in autonomous systems is complex and subjective.

Trustworthiness comprises at least two aspects:

1. **reliability** — does robot work reliably and predictably?

My Talk in One Slide

Questions I will *try* to address are

- What is our real worry about autonomous robots?
- How much confidence should we have in robots?

  ....and what evidence/justification can/should be provided?

Trust in autonomous systems is complex and subjective.

Trustworthiness comprises at least two aspects:

1. **reliability** — does robot work reliably and predictably?
2. **beneficiality** — robot is trying to help us, not harm us.

My Talk in One Slide

Questions I will *try* to address are

- What is our real worry about autonomous robots?
- How much confidence should we have in robots?

  ....and what evidence/justification can/should be provided?

Trust in autonomous systems is complex and subjective.

Trustworthiness comprises at least two aspects:

1. **reliability** — does robot work reliably and predictably?
2. **beneficiality** — robot is trying to help us, not harm us.

Evidence for both is difficult for most practical robotic systems.

My Talk in One Slide

Questions I will *try* to address are

- What is our real worry about autonomous robots?
- How much confidence should we have in robots?

  ....and what evidence/justification can/should be provided?

Trust in autonomous systems is complex and subjective.

Trustworthiness comprises at least two aspects:

1. **reliability** — does robot work reliably and predictably?
2. **beneficiality** — robot is trying to help us, not harm us.

Evidence for both is difficult for most practical robotic systems.

However: if we build the system appropriately and apply formal verification techniques, we can have much greater confidence in both, especially (2). We expose the **intention** of the system.

Overview

- *Autonomy*
  software taking more control

- *Software Architectures*
  designing for verifiability

- *Verification (and some issues)*
  techniques for ensuring software behaviour

- *Our Approach*
  an overview of some applications, and some benefits

Autonomous Systems

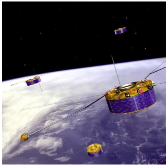
Autonomy:

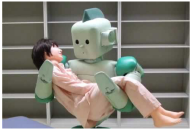
> *the ability of a system to make its own decisions and to act on its own, and to do both without direct human intervention.*

Autonomous Systems

Autonomy:

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





rtc.nagoya.riken.jp/RI-MAN

www.volvo.com

Who makes the Decisions?

Even within 'autonomy', there are important variations concerning decision-making:

Automatic: involves a number of fixed, and prescribed, activities; there may be options, but these are generally fixed in advance.

Who makes the Decisions?

Even within ‘autonomy’, there are important variations concerning decision-making:

Automatic: involves a number of fixed, and prescribed, activities; there may be options, but these are generally fixed in advance.

Adaptive: improves its performance/activity based on feedback from environment — typically developed using tight continuous control and optimisation, e.g. feedback control system.

Who makes the Decisions?

Even within 'autonomy', there are important variations concerning decision-making:

Automatic: involves a number of fixed, and prescribed, activities; there may be options, but these are generally fixed in advance.

Adaptive: improves its performance/activity based on feedback from environment — typically developed using tight continuous control and optimisation, e.g. feedback control system.

Autonomous: decisions made based on system's (belief about its) current situation at the time of the decision — environment still taken into account, but internal motivations/beliefs are important.

Who makes the Decisions?

Even within ‘autonomy’, there are important variations concerning decision-making:

Automatic: involves a number of fixed, and prescribed, activities; there may be options, but these are generally fixed in advance.

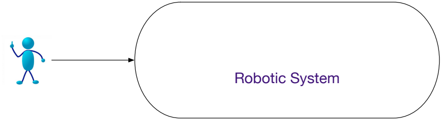
Adaptive: improves its performance/activity based on feedback from environment — typically developed using tight continuous control and optimisation, e.g. feedback control system.

Autonomous: decisions made based on system’s (belief about its) current situation at the time of the decision — environment still taken into account, but internal motivations/beliefs are important.

Distinguishing *between* these variations is often crucial.

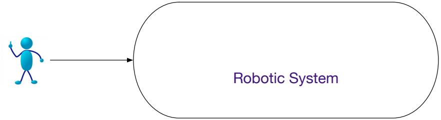
Who is in Control?

[Past] human operator/pilot/driver makes all the *key* decisions:

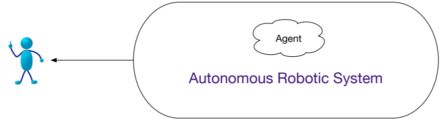


Who is in Control?

[Past] human operator/pilot/driver makes all the *key* decisions:

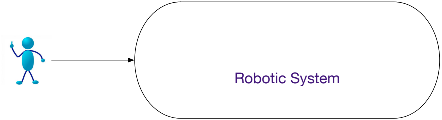


[Future] software agent makes many/most/all of these *decisions*:

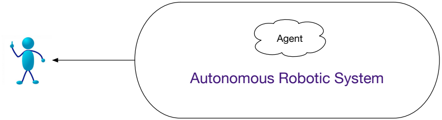


Who is in Control?

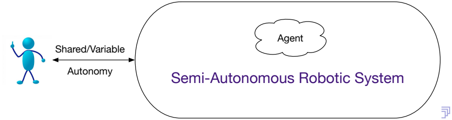
[Past] human operator/pilot/driver makes all the *key* decisions:



[Future] software agent makes many/most/all of these *decisions*:



[Present] *shared/variable autonomy* with changing responsibilities:



Concerns about Autonomy

Once the decision-making process is taken away from humans, even partially, can we be sure what autonomous systems will do?

Will they be safe? Can we ever trust them? What if they fail?

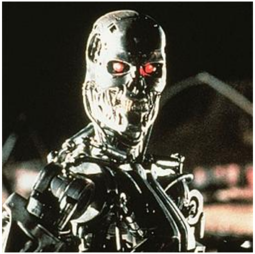
Especially important as robotic devices, autonomous vehicles, etc, are increasingly being deployed in safety-critical situations.

Concerns about Autonomy

Once the decision-making process is taken away from humans, even partially, can we be sure what autonomous systems will do?

Will they be safe? Can we ever trust them? What if they fail?

Especially important as robotic devices, autonomous vehicles, etc, are increasingly being deployed in safety-critical situations.



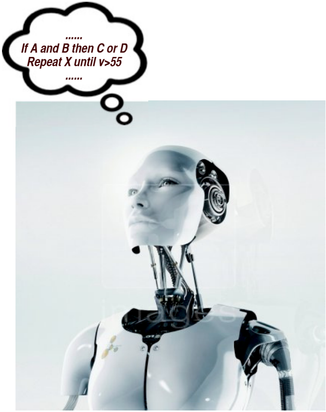
Terminator — 1984

No Psychiatrists for Robots?

As we move towards increased autonomy, we need to assess not just **what** the robot will do, but **why** it chooses to do it.

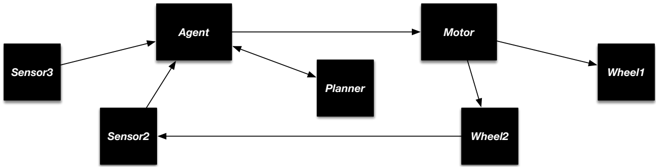
With an *autonomous system* we can (at least in principle) examine its internal programming and find out exactly

1. *what* it is "thinking",
2. what *choices* it has, and
3. why it *decides* to take particular ones.



Robot Architectures: Modularity

Middle-ware such as ROS (the "Robot Operating System") provides the separation of key architectural components and the basic mechanism for inter-operability, e.g:

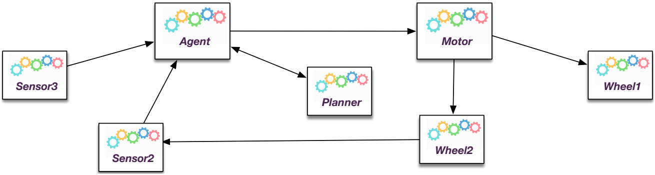


Important for: design; analysis; compositionality; maintenance; etc.

See: *ISO/BSI standard on Modularity (22166)*
`https://committee.iso.org/home/tc299`

Robot Architectures: Transparency

We do not want all/many/any of our modules to be *black boxes*.



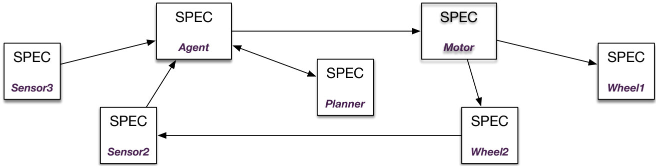
Increasingly, we require modular components to be *transparent* e.g. we must be able to inspect the internal behaviour of the module.

See: *IEEE P7001 — Transparency of Autonomous Systems*.
`https://standards.ieee.org/project/7001.html`

Robot Architectures: Verifiability

Being able to see the code still might not help us — understanding adaptive behaviour just from a component's implementation can be hard/impossible.

Instead, we require a concise, and *formal*, description of the anticipated/expected behaviour of each component:

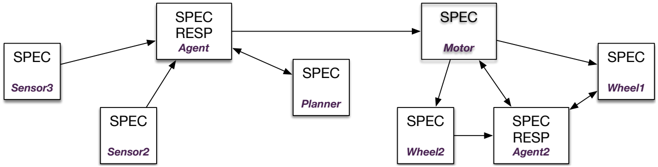


Robot Architectures: Responsibility

As systems become increasingly autonomous it is important to be clear where key decisions are made.

Who (human or agent) is *responsible* for these decisions.

And how is responsibility distributed, shared, and reinforced?



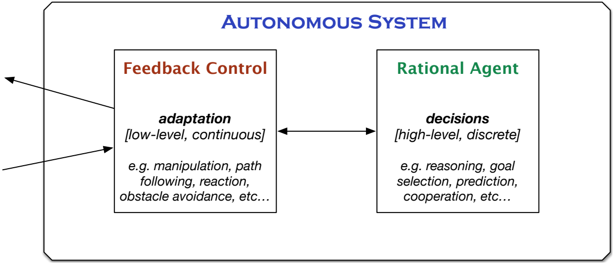
Areas of responsibility, and mechanisms for changing them, are explicitly highlighted in decision-making components (agents).

Hybrid Agent Architectures

The requirements for self-awareness and *reasoned* decisions and explanations has led on to *hybrid agent architectures* combining:

1. *rational agents* for high-level autonomous decisions, and
2. traditional *feedback control systems* for low-level activities,

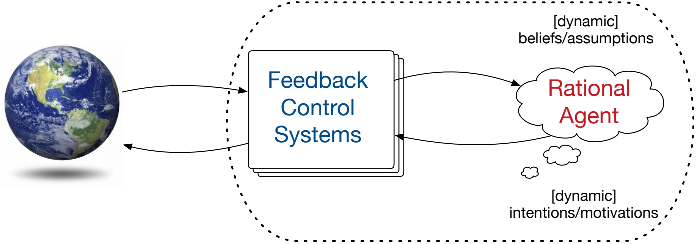
These have been shown to be easier to *understand*, *program*, *maintain* and, often, much more *flexible*.



Our Approach

Our approach is that

we should be certain what the autonomous system **intends** *to do and how it* **chooses** *to go about this*



A *rational agent* (typically, a BDI Agent):

must have explicit **reasons** *for making the choices it does, and should expose/explain these when needed*

Example: from Pilot to Rational Agent

Autopilot can essentially fly an aircraft

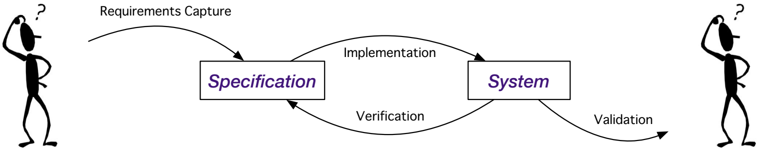
- keeping on a particular path,
- keeping flight level/steady under environmental conditions,
- planning routes around obstacles, etc.

Human pilot makes high-level decisions, such as

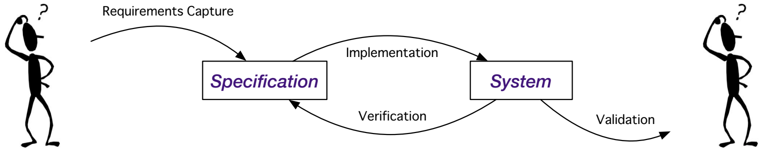
- where to go to,
- when to change route,
- what to do in an emergency, etc.

Rational Agent now makes the decisions the pilot used to make.

Verification and Validation



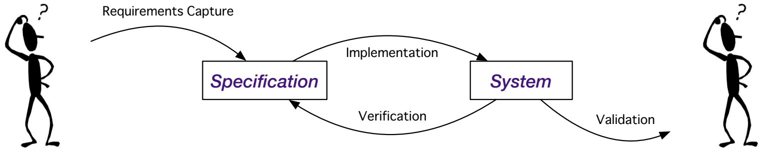
Verification and Validation



Verification, typically

- formal verification
- simulation-based testing
- physical testing

Verification and Validation



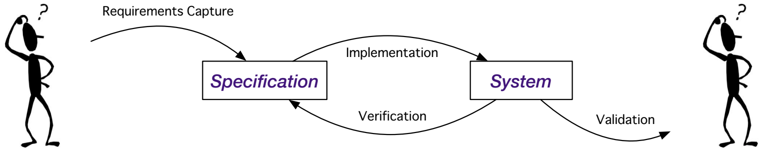
Verification, typically

- formal verification
- simulation-based testing
- physical testing

Validation, typically

- physical testing
- user validation
- test scenarios

Verification and Validation



Verification, typically

- formal verification
- simulation-based testing
- physical testing

Validation, typically

- physical testing
- user validation
- test scenarios

Verification = "are we building the system right?"
Validation = "are we building the right system?"

Varieties of Verification

Verifying requirement R of an agent can be done in many ways.

Varieties of Verification

Verifying requirement R of an agent can be done in many ways.

- Proof: behaviour of the agent is itself described by the logical formula, A, and verification involves *proving* $\vdash A \Rightarrow R$.

Varieties of Verification

Verifying requirement R of an agent can be done in many ways.

- Proof: behaviour of the agent is itself described by the logical formula, A, and verification involves *proving* $\vdash A \Rightarrow R$.
- Traditional Model-Checking: where R is checked against a representation of all possible execution paths of the agent.

Varieties of Verification

Verifying requirement R of an agent can be done in many ways.

- Proof: behaviour of the agent is itself described by the logical formula, A, and verification involves *proving* $\vdash A \Rightarrow R$.
- Traditional Model-Checking: where R is checked against a representation of all possible execution paths of the agent.
- Testing: where R is checked on a subset of the possible executions of the agent.

Varieties of Verification

Verifying requirement R of an agent can be done in many ways.

- Proof: behaviour of the agent is itself described by the logical formula, A, and verification involves *proving* $\vdash A \Rightarrow R$.
- Traditional Model-Checking: where R is checked against a representation of all possible execution paths of the agent.
- Testing: where R is checked on a subset of the possible executions of the agent.
- Synthesis: from R generate an automaton that only produces runs satisfying R.

Varieties of Verification

Verifying requirement R of an agent can be done in many ways.

- Proof: behaviour of the agent is itself described by the logical formula, A, and verification involves *proving* $\vdash A \Rightarrow R$.
- Traditional Model-Checking: where R is checked against a representation of all possible execution paths of the agent.
- Testing: where R is checked on a subset of the possible executions of the agent.
- Synthesis: from R generate an automaton that only produces runs satisfying R.
- Dynamic Fault Monitoring (aka Runtime Verification): where executions *actually* generated by the agent are checked against R.

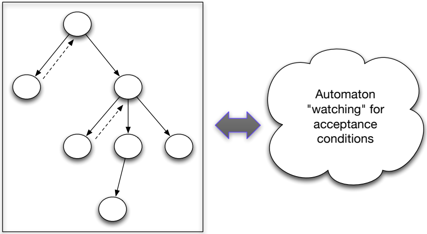
Varieties of Verification

Verifying requirement R of an agent can be done in many ways.

- Proof: behaviour of the agent is itself described by the logical formula, A, and verification involves *proving* $\vdash A \Rightarrow R$.
- Traditional Model-Checking: where R is checked against a representation of all possible execution paths of the agent.
- Testing: where R is checked on a subset of the possible executions of the agent.
- Synthesis: from R generate an automaton that only produces runs satisfying R.
- Dynamic Fault Monitoring (aka Runtime Verification): where executions *actually* generated by the agent are checked against R.
- Program Model-Checking: all execution paths the agent can take are checked against R.

Verifying (Rational) Agents

We usually use Program Model-Checking whereby a logical specification is checked against the *actual* agent code that is used in the robot.

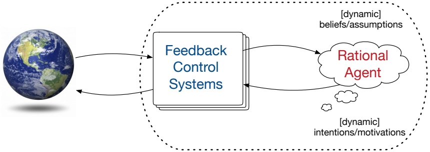


Modified Java virtual machine exploring all possible execution branches, not only by forward execution but by backtracking

Combines (backtracking) symbolic execution and a monitoring automaton in parallel (*"on the fly" model-checking*).

Verification Summary

With a hybrid agent-based architecture we can employ different verification techniques to different parts.



- We can *formally verify* the agent's decision-making $\longrightarrow$ we can be certain about this.
- We can *simulate/test* or *verify/monitor* the feedback control
- We can *practically test* whole system $\longrightarrow$ more confidence?

Agent essentially replaces the high-level, human, decision-making so formal verification here has an important role in certification.

Problems: Where do Requirements come from? (1)

Problems: Where do Requirements come from? (1)

They don't!

Problems: Where do Requirements come from? (1)

They don't!

We might be asked to verify.... Is it safe?

Problems: Where do Requirements come from? (1)

They don't!

We might be asked to verify.... Is it safe?

Is it usable?

Problems: Where do Requirements come from? (1)

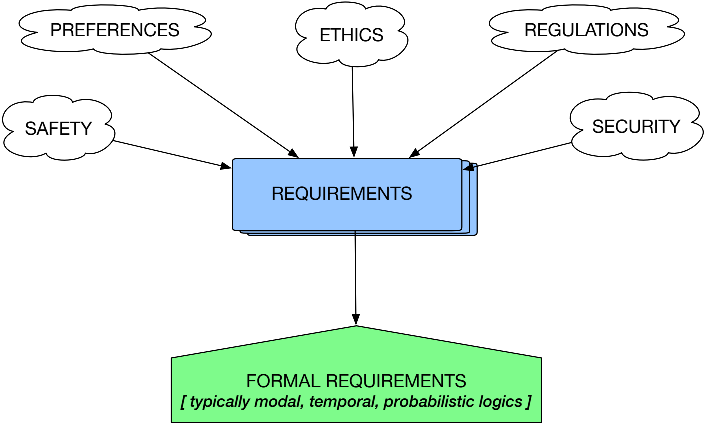
They don't!

We might be asked to verify.... Is it safe?

Is it usable?

Is it nice?

Problems: Where do Requirements come from? (2)



Problems: Complex Agent Requirements

Choosing the appropriate logic provides a level of abstraction close to the key concepts of the system. For example:

- dynamic communicating systems $\longrightarrow$ *temporal logics*
- systems managing information $\longrightarrow$ *logics of knowledge*
- autonomous systems $\longrightarrow$ *logics of motivation*
- situated systems $\longrightarrow$ *logics of belief, contextual logics*
- timed systems $\longrightarrow$ *real-time temporal logics*
- uncertain systems $\longrightarrow$ *probabilistic logics*
- cooperative systems $\longrightarrow$ *cooperation/coalition logics*

Problems: Complex Agent Requirements

Choosing the appropriate logic provides a level of abstraction close to the key concepts of the system. For example:

- dynamic communicating systems $\longrightarrow$ *temporal logics*
- systems managing information $\longrightarrow$ *logics of knowledge*
- autonomous systems $\longrightarrow$ *logics of motivation*
- situated systems $\longrightarrow$ *logics of belief, contextual logics*
- timed systems $\longrightarrow$ *real-time temporal logics*
- uncertain systems $\longrightarrow$ *probabilistic logics*
- cooperative systems $\longrightarrow$ *cooperation/coalition logics*

$\Rightarrow$ In realistic scenarios, we will need to *combine* several logics.

Explainability for Free

We have a rational agent that

1. has symbolic representations of its motivations (goals, intentions) and beliefs
2. reasons about these in order to decide what to do, and
3. records all the other options/reasons explored.

So, we have a trace of reasoned decisions.

Explainability for Free

We have a rational agent that

1. has symbolic representations of its motivations (goals, intentions) and beliefs
2. reasons about these in order to decide what to do, and
3. records all the other options/reasons explored.

So, we have a trace of reasoned decisions.

We can provide:

- recording facilities — *ethical black box*;
- interactive capabilities — "why did you do that?"
- .... "what will you do next, and why?"

Koeman, Dennis, Webster, Fisher, Hindriks — The "Why did you do that" Button: Answering Why-questions for end users of Robotic Systems. In Proc. EMAS, 2019.

Example: Robotic Safety?

`Care-o-Bot 3`

Robotic Assistants are now being designed to help the elderly or incapacitated.

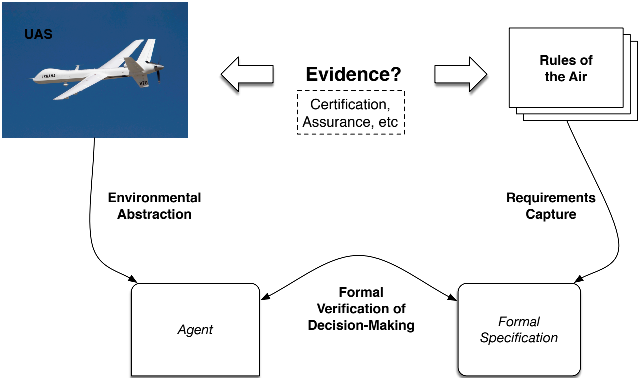
We can formally verify high-level rules that robot *actually* uses in deciding what to do.

And so can potentially prove critical properties such as

> *"robot will always try to wake the human when it believes there is a fire"*

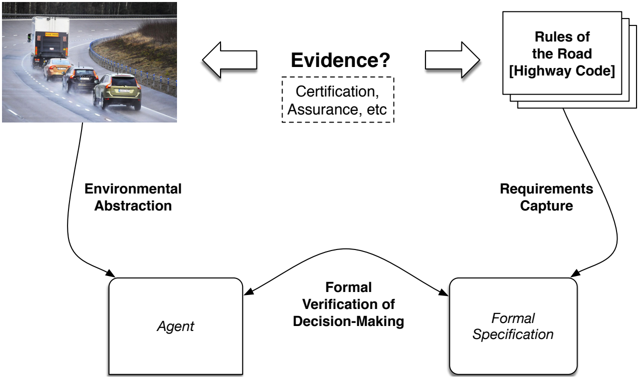
Webster, Dixon, Fisher, Salem, Saunders, Koay, Dautenhahn, Saez-Pons — Toward Reliable Autonomous Robotic Assistants Through Formal Verification: A Case Study. IEEE Trans. Human-Machine Systems, 2016.

Example: UAS Certification?



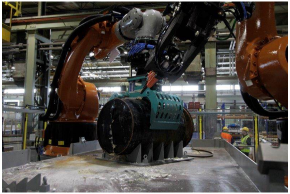
Webster, Cameron, Fisher, Jump — Generating Certification Evidence for Autonomous Unmanned Aircraft Using Model Checking and Simulation. Journal of Aerospace Information Systems, 2014.

'Driverless' Car Analysis?



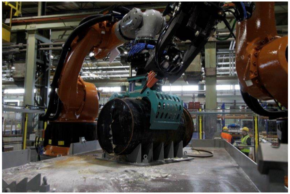
Alves, Dennis, Fisher — Formalisation of the Rules of the Road for embedding into an Autonomous Vehicle Agent. In Proc. FMAS'19 and Renault Book (2020).

Self-Awareness

National Nuclear Laboratory (Workington, Cumbria)

Self-Awareness

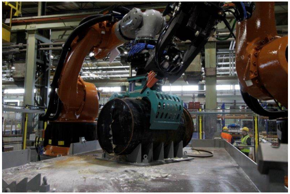



National Nuclear Laboratory (Workington, Cumbria)

To minimize human interventions, the robot will be

- aware of system components and their expected behaviours,
- able to monitor its own performance, and
- able to autonomously *reconfigure* its software

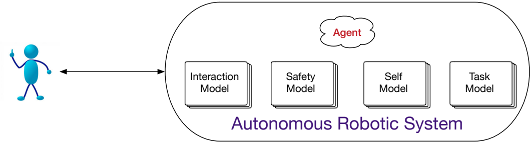
Self-Awareness



National Nuclear Laboratory (Workington, Cumbria)

To minimize human interventions, the robot will be

- aware of system components and their expected behaviours,
- able to monitor its own performance, and
- able to autonomously *reconfigure* its software



Ethical Example (1)

Ethical priority? save life >> save animals >> save property

System can order its options to take most *ethical* one.

Ethical Example (1)

Ethical priority? save life >> save animals >> save property

System can order its options to take most *ethical* one.

Once the agent decisions take ethical concerns into account then we can extend formal verification to also assess these.

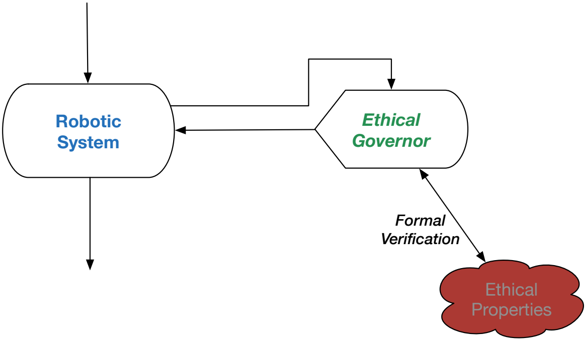
For example, we can formally verify that

if *a chosen course of action violates some substantive ethical concern, A*

then *the other available choices all violated some concern that was equal to, or more severe than, A.*

Dennis, Fisher, Slavkovik, Webster. Formal Verification of Ethical Choices in Autonomous Systems. Robotics & Autonomous Systems, 2016.

Ethical Example (2) [See Alan's talk]



Ethical governor is essentially a rational agent, so verify this agent against ethical requirements/properties.

Bremner, Dennis, Fisher, Winfield. On Proactive, Transparent, and Verifiable Ethical Reasoning for Robots. In Proc. IEEE, 2019.

Summary

- Use architectures that make design, analysis, and understanding easier
- Expose robot intentions (to users, to designers, to regulators)
- Use formal verification to provide stronger evidence

Summary

- Use architectures that make design, analysis, and understanding easier
- Expose robot intentions (to users, to designers, to regulators)
- Use formal verification to provide stronger evidence

As we move towards autonomy, we need to assess not just what the robot will do, but why it chooses to do it.

Reliable Autonomy

Especially with autonomous robotics, in safety-critical scenarios,

it is vital that we use strong verification techniques for any component/software we rely on

Reliable Autonomy

Especially with autonomous robotics, in safety-critical scenarios,

> it is vital that we use strong verification techniques for any component/software we rely on

Perhaps, as a corollary,

> never fully **rely** on any component that has not been formally verified

Reliable Autonomy

Especially with autonomous robotics, in safety-critical scenarios,

it is vital that we use strong verification techniques for any component/software we rely on

Perhaps, as a corollary,

never fully **rely** on any component that has not been formally verified

With increasing autonomy, we rely on the decision-making agent, and so must be **sure** it will do what we want.