Increasing Confidence in Autonomous Systems

Michael Fisher[1] Angelo Ferrando[2] Rafael C. Cardoso[1]

1: *University of Manchester, UK* 2: *University of Genova, Italy*

VORTEX Workshop *12th July 2021*

Engineering and Physical Sciences Research Council

Royal Academy of Engineering

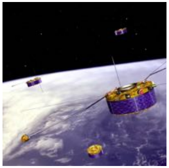
Overview

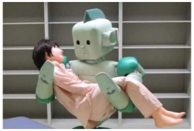
- *Why is Autonomy Important?*
  (software is taking more control)
- *Why do we need Verification?*
  (might remove barriers to wider use of autonomy)
- *Runtime Verification*
  (can be useful?)
- *Examples*
  (of RV, often in combination with other techniques)

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

Do Decisions Matter?

In a predictable and known environment
.... *we can enumerate all decisions that might be needed and pre-code answers*

In non-critical scenarios
.... *we probably don't care too much how decisions are made*

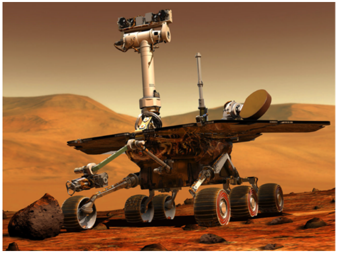
But: for critical systems in uncertain environments
.... *we need to be much clearer about how decisions are made*
.... *crucially, we need to know* **why** *decisions are made*

Context: Nuclear

Robotics and AI for Nuclear (RAIN)
`https://rainhub.org.uk`

RAIN
ROBOTICS AND AI IN NUCLEAR

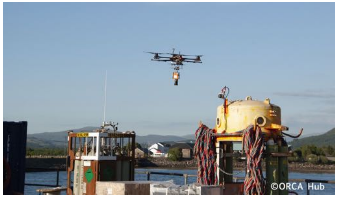
Context: Space



Future AI and Robotics for Space
(FAIR-SPACE)
`https://www.fairspacehub.org`

FAIR-SPACE

Context: Offshore



UK Robotics and Artificial Intelligence Hub for Offshore Energy Asset Integrity Management (ORCA)
`https://orcahub.org`

ORCA HUB
Offshore Robotics for Certification of Assets
Remote Safety and Integrity

Context: Domestic/Social/Vehicles

Care-o-Bot 3

`http://www.sartre-project.eu`

TAS Verifiability Node
`https://verifiability.org`

UKRI Verifiability Node

Why Verification?

Once systems are truly autonomous, able to make key decisions without human intervention, all the following need stronger evidence:

- developers/designers/engineers
- users/passengers
- regulators

All aim for comprehensive V&V for autonomous systems.

Who Cares? — Public

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

Who Cares? — Standards and Regulators

Standards:

- IEEE P7009 standard on *Fail-safe Design of Autonomous Systems* `standards.ieee.org/project/7009.html`
- IEEE P7001 standard on *Transparency of Autonomous Systems* `standards.ieee.org/project/7001.html`
- British Standards Institution (BSI) BS8611 standard on *Ethical Design and Application of Robots and Robotic Devices*

  etc....

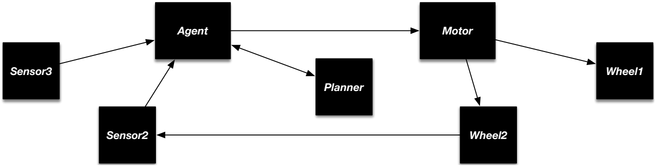
Regulators:

- across nuclear, offshore, air, road, etc....
- issue of "autonomy" remains a barrier
- stronger evidence required

  etc....

Full System Verification is Difficult, but....

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



Important for many aspects, especially compositional verification.

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

Architectures help focus Verification Effort

We can identify which components need the *strongest* verification - usually the ones that take the key (safety-crtitical) decisions

$\longrightarrow$ *identified through straighforward causality analysis.*

Risk/Causality analysis, together with the current trajectory of autonomous systems design, leads us to

heterogeneous, modular architectures

Importantly, since the type of component we have crucially affects the varieties of verification that can give us strong results, then we are naturally led to

heterogeneous verification

Heterogeneous/Corroborative Verification

Heterogeneous Verification $\longrightarrow$ different verification techniques for distinct components

Corroborative Verification $\longrightarrow$ different verification applied to one component

Role of RV?

We use (or plan to use, in some cases) runtime verification to

recognise situations we do not expect

To give a flavour of where we have used, and might use, runtime verification to help increase confidence, we provide a range of examples...

ROSMonitoring (1)

ROSMonitoring has three main components:

Monitor $\longrightarrow$ implementation, responsible for intercepting messages between ROS nodes and communicating with the oracle.

Instrumentation $\longrightarrow$ inserts the monitor(s) into the ROS communication structure.

Oracle $\longrightarrow$ checks whether the events that are observed by the monitor conform to the required formal specification or not.

Ferrando, Cardoso, Fisher, Ancona, Franceschini, Mascardi. ROSMonitoring: A Runtime Verification Framework for ROS. In Proc. TAROS, 2020.

`github.com/autonomy-and-verification-uol/ROSMonitoring`

ROSMonitoring (2)

Default Oracle is **RML**

Runtime Monitoring Language (RML): A rewriting-based and system agnostic Domain Specific Language for RV used for defining formal properties and synthesising monitors from them.

RML is system agnostic, imposing no restrictions on how events are generated.

RML

`https://rmlatdibris.github.io`

ROSMonitoring (3)

From

$$\begin{tikzcd}\boxed{\text{NodeA}} \arrow[r, "\text{S}"] & \boxed{\text{NodeB}} \arrow[r, "\text{T}"] & {}\end{tikzcd}$$

To

$$\begin{tikzcd}\boxed{\text{NodeA}} \arrow[r, "\text{S}"] & \boxed{\text{Mon}} \arrow[r, "\text{X}"] & \boxed{\text{NodeB}} \arrow[r, "\text{T}"] & {}\end{tikzcd}$$

Robotic Safety

Standards often describe minimum safe requirements in practice.
$\longrightarrow$ typically: proximity; speed; force.

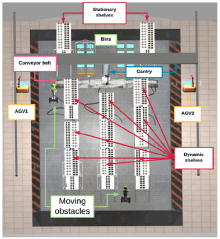
International Organization for Standardization (ISO). ISO 10218 — Safety Requirements for Industrial Robots, 2016.
`https://www.iso.org/standard/51330.html`.

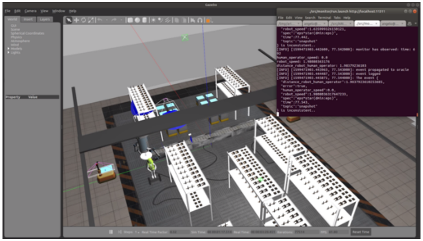
While we will likely carry out some analysis (e.g. simulation-based testing) on systems, these are always incomplete.

So: natural to add runtime monitors to ensure proximity/speed requirements are maintained in practice.

Example: ARIAC

Agile Robotics for Industrial Automation Competition (ARIAC) is an automated manufacturing competition hosted by NIST (USA).





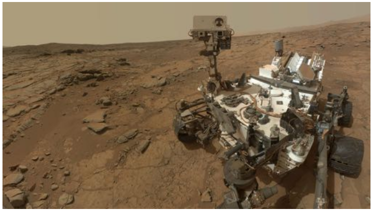
```
  "robot_speed":1.635999326150121,
  "spec":"eps*star(dmin:eps)",
  "time":77.442,
  "topic":"snapshot"
} is inconsistent..
[INFO] [1599471903.442609, 77.542000]: monitor has observed: time: 6
92
human_operator_speed: 0.0
robot_speed: 1.98888363176
distance_robot_human_operator: 1.98379236183
[INFO] [1599471903.443963, 77.543000]: event propagated to oracle
[INFO] [1599471903.444487, 77.543000]: event logged
[INFO] [1599471903.445871, 77.544000]: The event {
  "distance_robot_human_operator":1.9837923610213685,
  "error":true,
  "human_operator_speed":0.0,
  "robot_speed":1.98888363176 47233,
  "spec":"eps*star(dmin:eps)",
  "time":77.543,
  "topic":"snapshot"
} is inconsistent..
```

Competition assessed on achieving tasks, but we added RV components to recognise when robot is too close/fast near human.

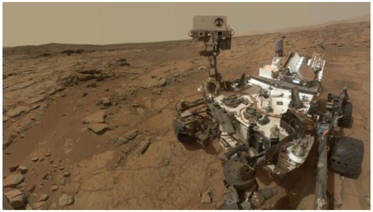
Ferrando, Kootbally, Piliptchak, Cardoso, Schlenoff, Fisher — Runtime Verification of the ARIAC Competition: Can a Robot be Agile and Safe at the same time? AIRO@AI*IA 2020.

Intervention Example: Mars Curiosity

No intervention:



Intervention/Filtering:



Example: Security

There are so many potential threats that we cannot realistically add monitors for all of these.

So, our approach here is to use general security analysis to highlight the most likely, or most significant, threats and aim to recognise these through formal verification.

Maple, Bradbury, Yuan, Farrell, Dixon, Fisher, Atmaca — Security-minded Verification of Space Systems. In Proc. IEEE Aerospace Conference, 2020.

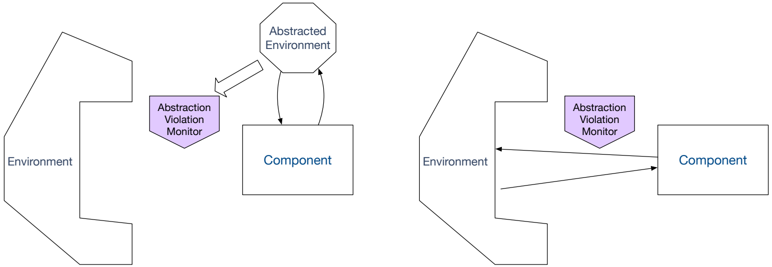
Violating Assumptions

RV naturally fits well with static analysis techniques.

With these analyses we can guarantee correctness - but under assumptions.

So: produce runtime monitors to recognise assumption violation.

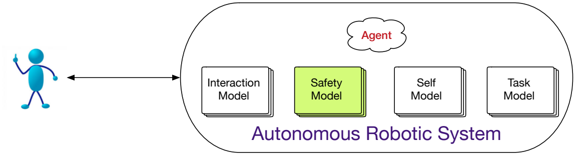
Example: Verification Violations



If a violation is recognised, then behaviour is no longer guaranteed.

Ferrando, et al. Toward a Holistic Approach to Verification and Validation of Autonomous Cognitive Systems. ACM Transactions on Software Engineering and Methodology (2021).

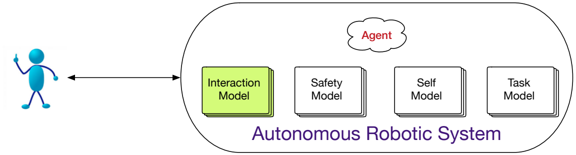
Example: Certification Violations



Assumptions in Safety Case — need to know when violated!

Dennis, Fisher. Verifiable Self-Aware Agent-Based Autonomous Systems. Proc. IEEE, 2020.

Example: User Model Violations

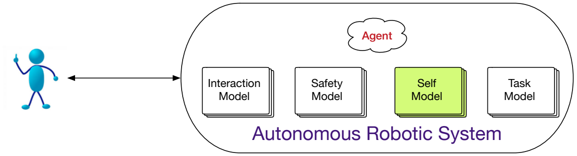


Typically, complex autonomous systems incorporate models of the users they deal with — often of the "Theory of Mind" kind.

However, it will be important to recognise when human behaviour regularly slips beyond the expected envelope

Dennis, Fisher. Verifiable Self-Aware Agent-Based Autonomous Systems. Proc. IEEE, 2020.

Example: Failure



RV can be used to monitor whether the behaviour of various components match expectations.

Example: we might add a monitor to recognise whether the movement module actually achieves its target destination or not.

In case of failure, the agent might choose to use other movement options or simply adapt its description of the module's capabilities.

Cardoso, Dennis, Fisher. Plan library reconfigurability in BDI agents. In Proc. EMAS, 2019.

Example: Predictions

RV not only used to report violations, but to help predict unexpected behaviour according to some previous knowledge of the system $\longrightarrow$ predictive runtime verification.

Becomes relevant when detecting a failure occurs too late to avoid.

Example: UAV battery consumption

This can be done by performing a prognostic analysis on the battery, and using the resulting model to predict future events at runtime for the monitor.

Zhao, et al. Towards Integrating Formal Verification of Autonomous Robots with Battery Prognostics and Health Management. In Proc. SEFM, 2019.

Concluding Remarks

RV is useful.

Particularly when used in combination with other verification techniques.

But: not for everything

- use (informal) analysis to decide where to use RV
- use RV as a by-product of (critical) formal verification
- resource implications (less of a problem in ROS)

Sample Relevant Publications

- Cardoso, Dennis, Fisher. Plan library reconfigurability in BDI agents. In *Proc. EMAS*, 2019.
- Dennis, Fisher. Verifiable Self-Aware Agent-Based Autonomous Systems. *Proc. IEEE*, 2020.
- Dennis, Fisher, Lincoln, Lisitsa, Veres. Practical Verification of Decision-Making in Agent-Based Autonomous Systems. *Journal of Automated Software Engineering*, 2016.
- Ferrando, Cardoso, Fisher, Ancona, Franceschini, Mascardi. ROSMonitoring: A Runtime Verification Framework for ROS. In *Proc. TAROS*, 2020.
- Ferrando, Dennis, Cardoso, Fisher, Ancona, Mascardi. Toward a Holistic Approach to Verification and Validation of Autonomous Cognitive Systems. *ACM Transactions on Software Engineering and Methodology* (2021).
- Ferrando, Kootbally, Piliptchak, Cardoso, Schlenoff, Fisher. Runtime Verification of the ARIAC Competition: Can a Robot be Agile and Safe at the same time? In *Proc. AIRO@AI*IA*, 2020.
- Fisher, Dennis, Webster. Verifying Autonomous Systems. *CACM*, 2013
- Fisher, Mascardi, Rozier, Schlingloff, Winikoff, Yorke-Smith.Towards a Framework for Certification of Reliable Autonomous Systems. *Autonomous Agents and Multi-Agent Systems*, 2021.
- Kamali, Dennis, McAree, Fisher, Veres. Formal Verification of Autonomous Vehicle Platooning. *Sci. Computer Programming*, 2017.
- Luckcuck, Farrell, Dennis, Dixon, Fisher. Formal Specification and Verification of Autonomous Robotic Systems: A Survey. *ACM Computer Surveys*, 2019.
- Maple, Bradbury, Yuan, Farrell, Dixon, Fisher, Atmaca — Security-minded Verification of Space Systems. In *Proc. IEEE Aerospace Conference*, 2020.
- Webster, Cameron, Fisher, Jump. Generating Certification Evidence for Autonomous Unmanned Aircraft Using Model Checking and Simulation. *Journal of Aerospace Information Systems*, 2014.
- Zhao, Osborne, Lantair, Robu, Flynn, Huang, Fisher, Papacchini, Ferrando. Towards Integrating Formal Verification of Autonomous Robots with Battery Prognostics and Health Management. In *Proc. SEFM*, 2019.