Verifying Autonomous Robots: Challenges and Reflections

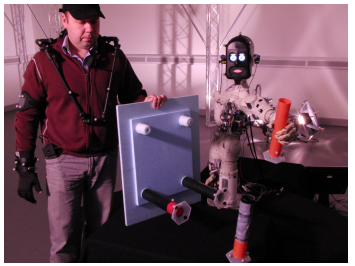
Clare Dixon
Department of Computer Science
University of Manchester

Thanks to collaborators from

- University of Liverpool/Manchester Autonomy and Verification Lab (autonomy-and-verification-uol.github.io)
- Trustworthy Robotic Assistants Project (www.robosafe.org)
- FAIR-SPACE Project www.fairspacehub.org
- Science of Sensor Systems Software Project (www.dcs.gla.ac.uk/research/S4/)
- RAIN Project (rainhub.org.uk)

Autonomous Robots

- Autonomous robots are being developed to operate in a variety of situations such as industrial, transportation, domestic and health care settings which may benefit society and improve lives.
- The robots will need to be able to act autonomously and make decisions to choose between a range of actions.
- In addition they may need to operate in changing, unknown or hazardous environments working close to or in collaboration with humans.
- How do we make sure they are trustworthy, safe, reliable and do what they are supposed to?



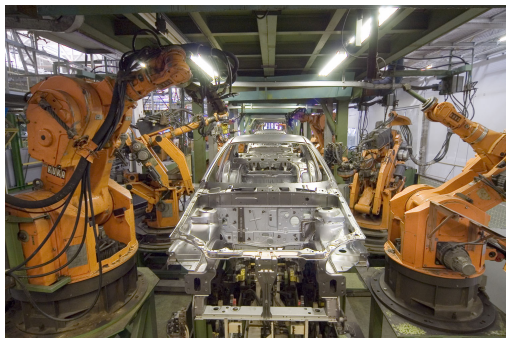
What is Trustworthiness and Safety?

- Safety involves showing that the robot does nothing that (unnecessarily) endangers the person.
- There are ISO safety requirements and guidelines for industrial robots (ISO 10218, 2011), personal care robots (ISO 13482, 2014), and for collaborative robots (ISO 15066, 2016).
- Trustworthiness involves social issues beyond pure safety.
- It is not just a question of whether the robots are safe but whether they are *perceived* to be safe, useful and reliable.
- BSI 754 considers software trustworthiness including safety, reliability, availability, resilience and security.
- There are also legal, ethical, privacy etc issues such as
  - the robot spills a hot drink on someone;
  - the robot doesn't remind the person to take their medicine;
  - the robot doesn't go to the kitchen when told?

Robots in the Workplace and at Home

Currently many robots used in industry or domestic use operate in limited physical space or have limited functionality. This helps assure their safety.

- Robots' industrial environments are limited so they can only move in a fixed area and have limited interactions with humans e.g. welding or paint spraying robots.
- Small or limited capability domestic robots, e.g., vacuum cleaning robots, robot lawn mowers, pool cleaning robots etc



Verification and Validation

We advocate an approach using verification and validation of systems.

Verification: Are we building the system right?

Validation: Are we building the right system?

Verification, for example

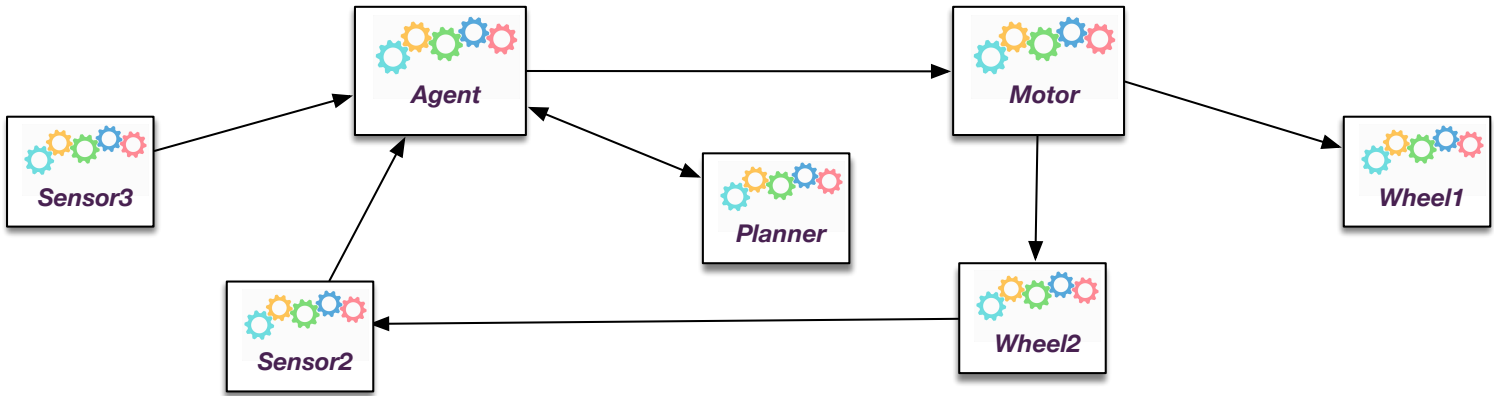
- formal verification
- simulation-based testing
- physical testing

Validation, for example

- physical testing
- user validation
- test scenarios

Robot Architectures: Modularity

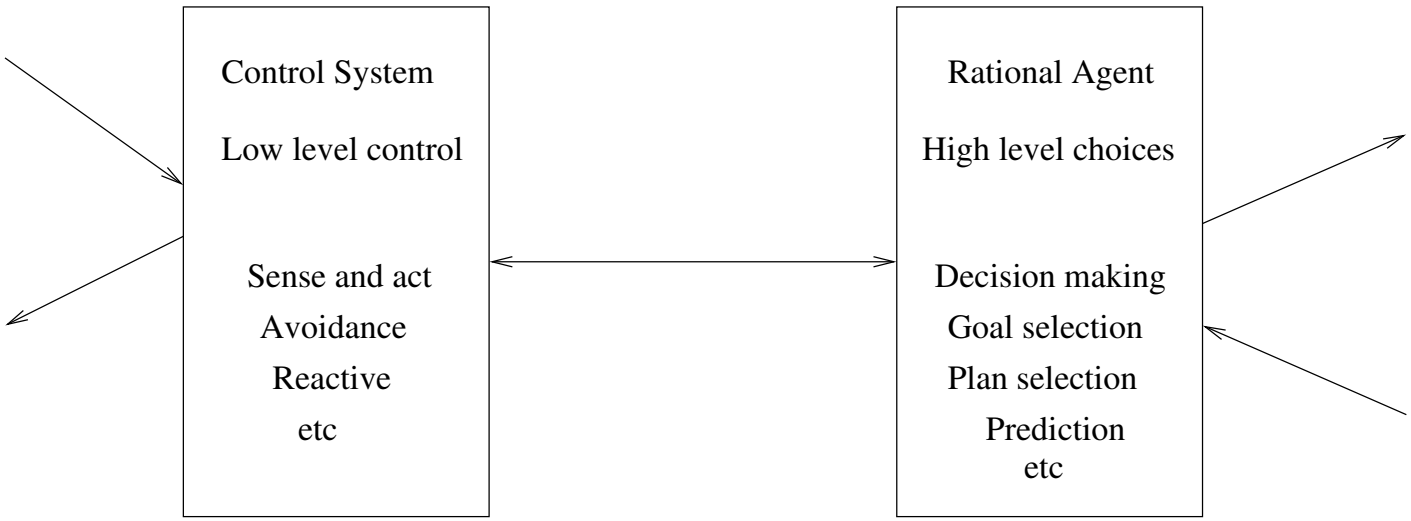
Architectures that are modular, separating key components are important to not only for verification but also for design, analysis, compositionality, maintenance, re-use etc.



Different types of verification may be more appropriate to different components.

Robot Architectures: Decision Making

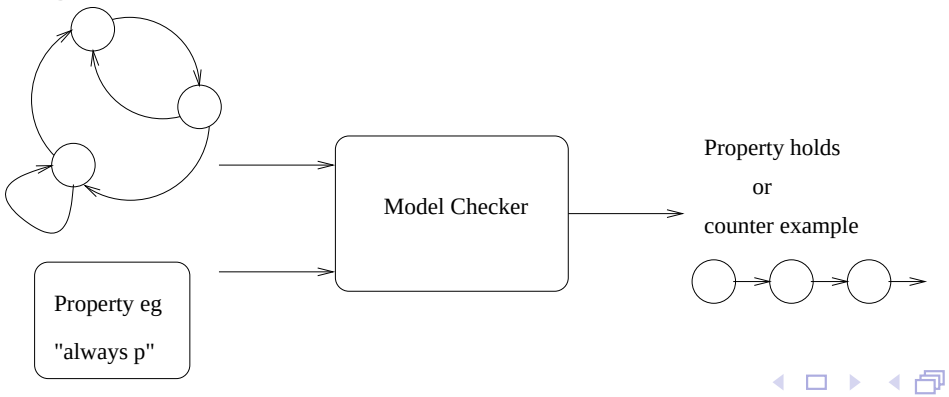
We assume an architecture where there is a separation between the high level decision making layer and the low level control layer.



We aim to represent and (formally) verify the decision making layer and we don't deal with low level control such as movement etc.

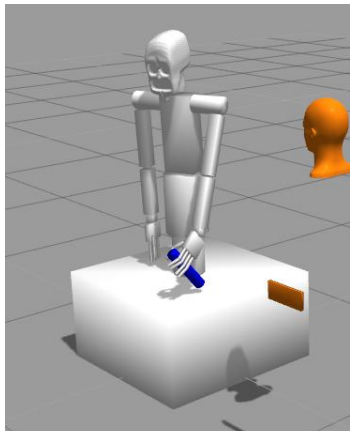
Formal Verification

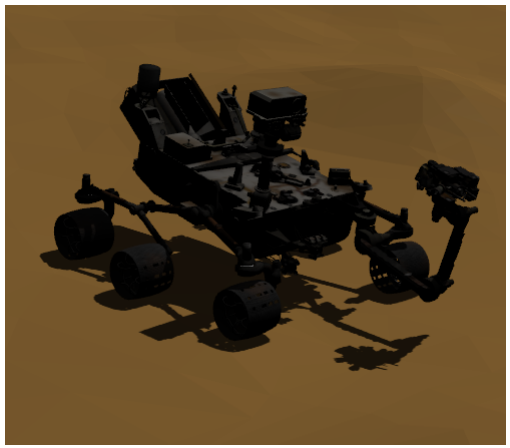
- A mathematical analysis of all behaviours using logics, and tools such as theorem provers or model checkers.
- We focus on temporal verification using automatic tools and techniques that do not require user interaction.
- Model checking is a fully automatic, algorithmic technique for verifying the temporal properties of systems.
- Input to the model checker is a model of the system and a property to be checked on that model.
- Output is that the property is satisfied or a counter example is given.



Simulation Based Testing

- This is an testing methodology widely used in the design of micro-electronic and avionics systems.
- Testing in simulation environments can cover a wide range of practical situations and may allow many more tests to be carried out than with testing in the real world.
- Producing tests can be carried out in different ways (model based, pseudorandom, etc) and tools are used to automate the testing and analyse the coverage of the tests.

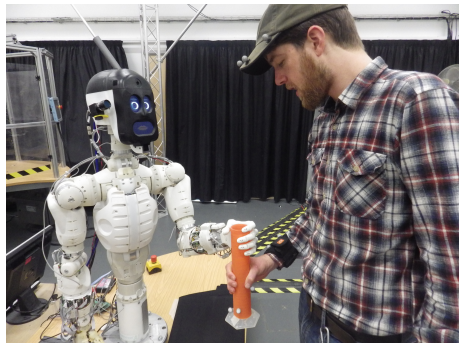




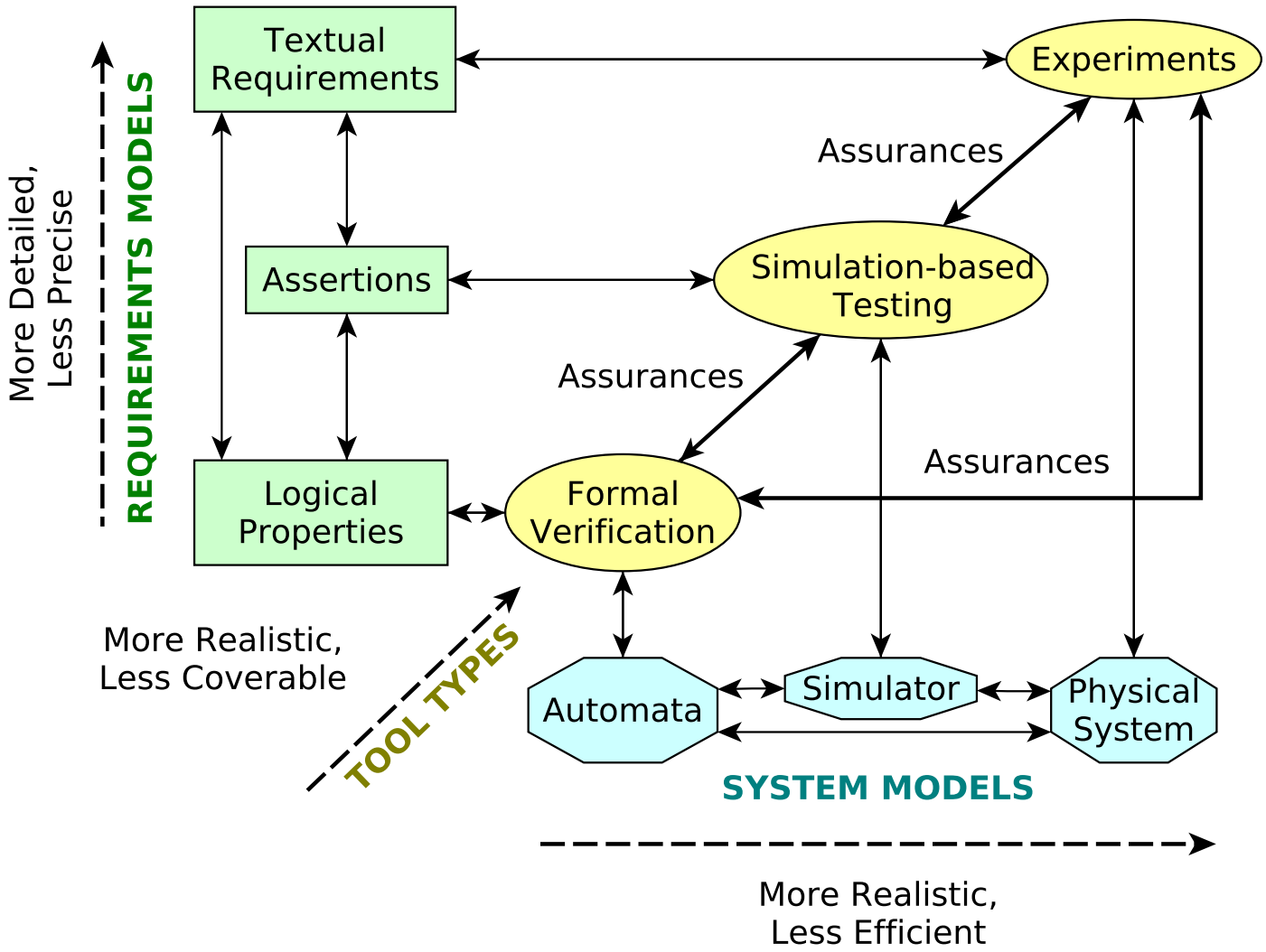
End User Validation

- This approach involves experiments and user evaluations in practical robot scenarios.
- Scenarios relating to robot human interaction are developed to test some hypothesis and experiments with users carried out.
- This helps establish whether the human participants indeed view the robots as safe and trustworthy etc.





Overall Approach

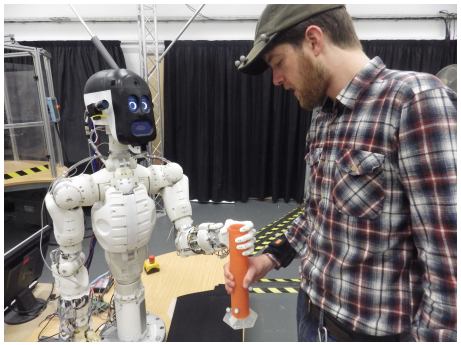


Verifying Robot Assistants

The Trustworthy Robot Assistants Project developed and applied three different approaches to verification and validation of robot assistants.

Each approach is aimed at increasing trust in robot assistants.

- Formal Verification (UoL)
- Simulation-based Testing (BRL)
- End-user Validation (UoH)



We focus on two use cases domestic (Care-O-bot® at UoH) and manufacturing (BERT at BRL).

A Domestic Robot Assistant

- Here we apply model checking to the high level behaviours controlling the (commercially available) Care-O-bot®, manufactured by Fraunhofer IPA.
- It is based on the concept of a “robot butler” which has been developed as a mobile robot assistant to support people in domestic environments.
- It has a manipulator arm, an articulated torso, stereo sensors serving as “eyes”, LED lights, a graphical user interface, and a movable tray.

- The robot’s sensors monitor its current location, the state of the arm, torso, eyes and tray.
- Its software is based on the Robot Operating System.

Care-O-bot and Robot House

- This is deployed in a domestic-type house (the robot house) at the University of Hertfordshire.
- The robot house is equipped with sensors which provide information on the state of the house and its occupants, such as whether the fridge door is open and whether someone is seated on the sofa.
- Low-level robot actions such as movement, speech, light display, etc., are controlled by groups of high-level rules that together define particular behaviours.

Care-O-bot Decision Making: Behaviours

- The Care-O-bot's high-level decision making is determined by a set of behaviours of the form **precondition** $\rightarrow$ **action** (each a sequence of rules).
- UoH have developed a number of behaviour sets. Here we focus on a set with 31 default behaviours.
- Examples of high-level rules can take the form "lower tray", "move to sofa area of the living room", "say 'The fridge door is open' ", set a flag, check a sensor etc.
- Only one behaviour executes at once.
- Each behaviour has a priority (integer between 0 and 90). Higher priority behaviours are executed in preference to lower priority behaviours.
- Each behaviour is flagged as interruptible or not.
- Once it has started executing, a behaviour will execute to completion, if it is not interruptible.

The `S1-alertFridgeDoor` Behaviour

Behaviours (a set of high level rules) take the form:

Precondition-Rules -> Action-Rules

```
27  Fridge Freezer Is *ON* AND has been ON for more than 30 secs
31  ::514:: GOAL-fridgeUserAlerted is false
    32  Turn light on ::0::Care-o-Bot 3.2 to yellow
    34  move ::0::Care-o-Bot 3.2 to ::2:: Living Room and wait for
            completion
    35  Turn light on ::0::Care-o-Bot 3.2 to white and wait for
            completion
    36  ::0::Care-o-Bot 3.2 says 'The fridge door is open!' and
            wait for completion
    37  SET ::506::GOAL-gotoCharger TO false
    38  SET ::507::GOAL-gotoTable TO false
    39  SET ::508::GOAL-gotoSofa TO false
    40  ::0::Care-o-Bot 3.2 GUI, S1-Set-GoToKitchen, S1-Set-WaitHere
    41  SET ::514::GOAL-fridgeUserAlerted TO true
```

Its priority is 60 and it is not interruptible.

Models and Properties

- We need to abstract away from some of the timing details included in the database to obtain a model that is discrete, finite and not too large.
- We developed a (by hand) model in the input language for the model checker NuSMV and later developed a tool (`CRutoN`) to automatically translate from behaviours to NuSMV input.
- We also need a set of properties of the system to check over the model.
- Ideally these would come from a specification or standards documents about what is expected of the robot with respect to functionality, safety etc.
- Here we focus on issues relating to the scheduling of behaviours, priorities and interruptions (which at least provide a sanity check).

Sample Properties and Model Checking Results

1 $\Box((fridge\_freezer\_on \wedge \neg goal\_fridge\_user\_alerted) \Rightarrow \Diamond(location = livingroom \wedge \Diamond say = fridge\_door\_open))$

2 $\Box((fridge\_freezer\_on \wedge \neg goal\_fridge\_user\_alerted \wedge schedule = schedule\_alert\_fridge\_door) \Rightarrow \Diamond(location = livingroom \wedge \Diamond say = fridge\_door\_open))$

Property	Output	Time (sec)
1	FALSE	11.1
2	TRUE	12.3

The model had 130,593 reachable states.

- We did find a small bug in the behaviours (a flag was wrongly set) but this was by inspection of the behaviours.
- It would be better to try properties relating to the requirements of the robot.

Domestic Robot Assistant: Discussion

- Understanding the semantics of the robot execution cycle took a lot of close work and interaction with UoH.
- The state explosion problem means we have to find a balance between the level of detail/abstraction and verification times (timing details were not well represented).
- This approach isn't very general for different ways of robot decision making and the person has not been modelled.
- `CRutoN` allowed us to translate from different databases of behaviours into input for a model checker, setting parameters to control particular aspects of the translation.
- `CRutoN` uses an intermediate representation so that input to different model checkers can potentially be generated.
- We could deal better with uncertainty or timing constraints by using a different model checker.

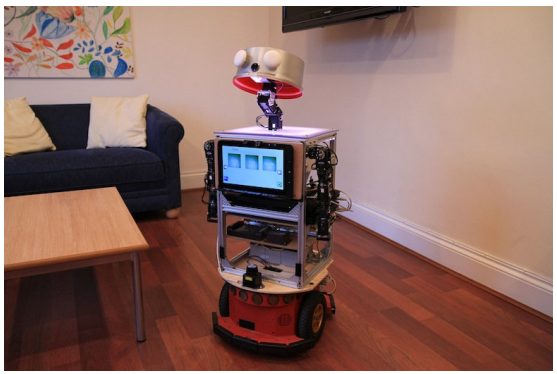
Experiments with Trust and Reliability

UoH experimented (40 participants) using two scenarios in the robot house where the robot appeared faulty or not.

In both scenarios the person was asked to carry out a task with the robot.

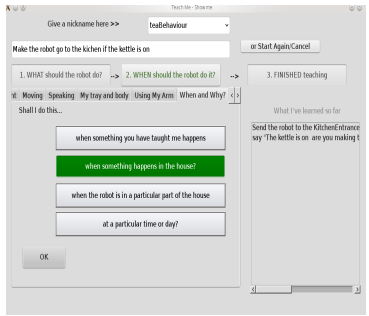
Results suggested that although errors in a robot's behaviour are likely to affect participant's perception of its reliability and trustworthiness, this doesn't seem to influence their decisions to comply with instructions (or not).

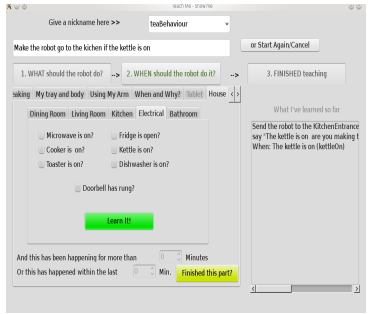
Their willingness to comply with the robot's instructions seem to depend on the nature if the task, in particular, whether its effects are irrevocable.



Learning New Behaviours

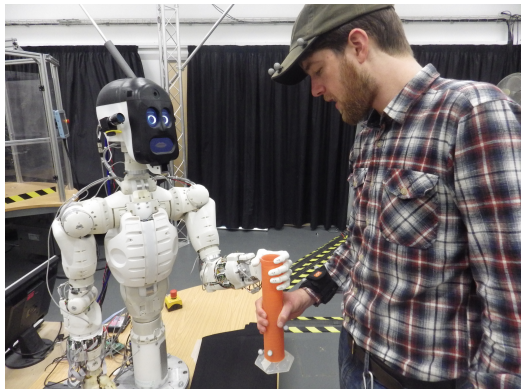
- The robot has an interface to personalise behaviours so that users can use existing primitives to create new behaviours.
- However these may affect or be affected by existing behaviours and may never run.
- We developed a verification algorithm that carries out a static check on newly added behaviours relating to their priorities and preconditions presenting issues to the user.
- A user study showed that the verification approach was significantly more useful for understanding and resolving interference between behaviours than without it and no technical background was needed to understand this.

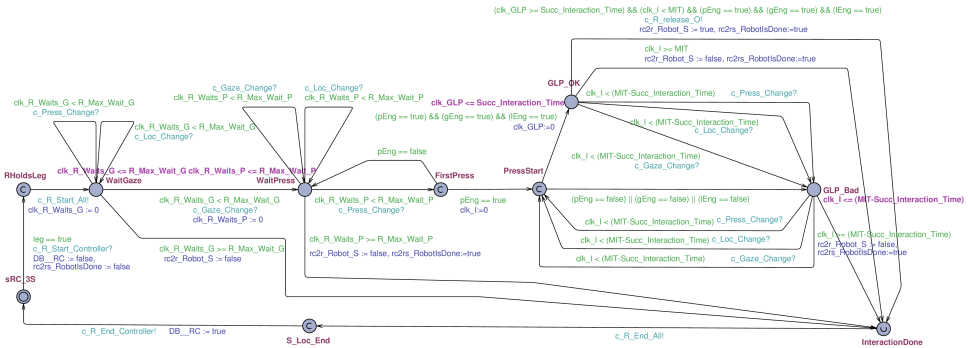




The Manufacturing Scenario: Verification

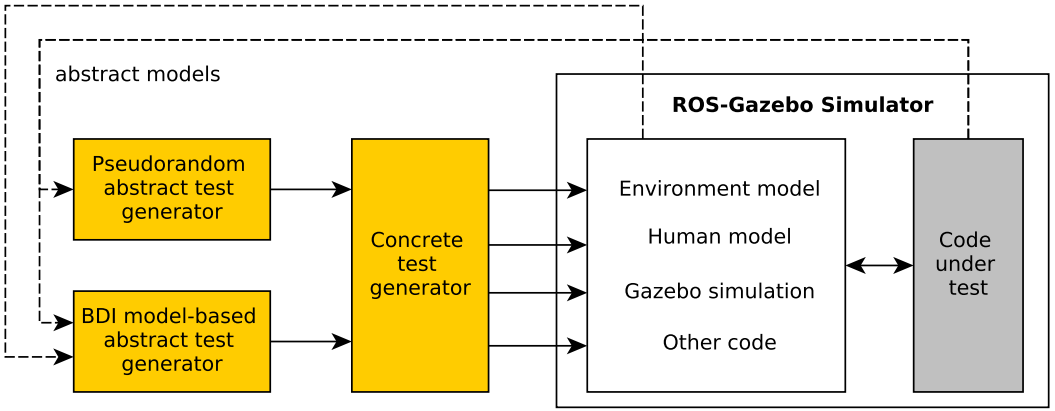
The focus was on a table leg handover task. The gaze, hand location and hand pressure of the human should be correct before the handover takes place.

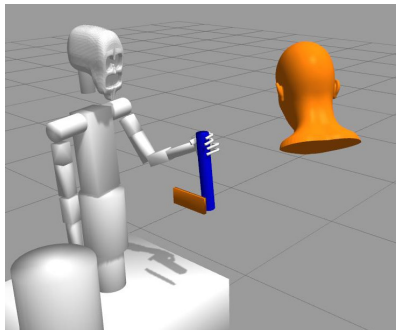




Modelling was carried out using Probabilistic Timed Automata (PTA) and verification via the PRISM probabilistic model checker.

Manufacturing Scenario: Simulation Based Testing





A simulator was implemented in the ROS framework for robot code development and the Gazebo simulator (BRL).

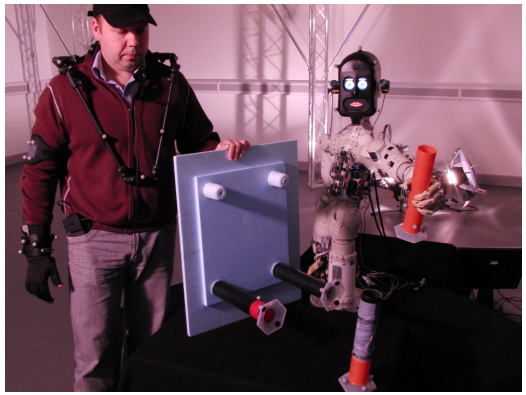
A combination of model-based and pseudorandom test generation was used.

We used the formal PTA model to develop abstract tests of high-level actions for the human.

Simulation based testing revealed that the robot sometimes dropped the table leg accidentally (gripper failure).

Manufacturing Scenario: Real Robot Experiments

We carried out a small user validation study with 10 participants each carrying out 10 handover tasks.



Subjects were given clear instructions on how to successfully complete the task, followed by practice sessions.

They were instructed to try to complete the task successfully in each test.

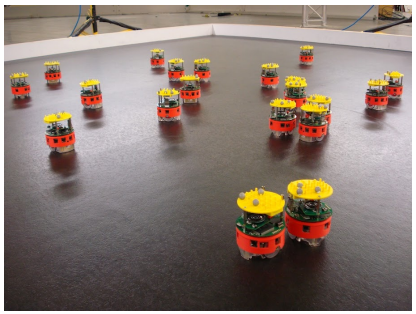
The experiments revealed false negative results for the pressure and location sensors, i.e. they were wrongly reported as too low/incorrect hand position when they were in fact correct.

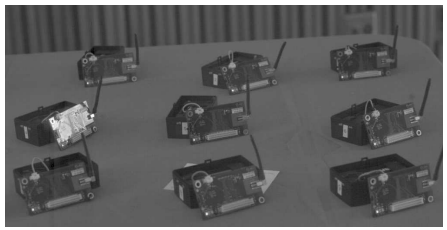
The Manufacturing Scenario: Discussion

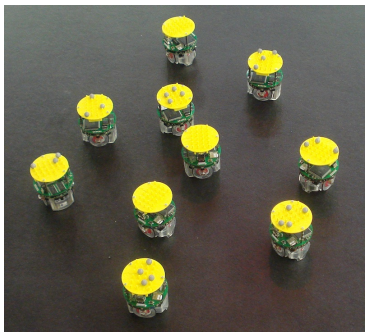
- A number of properties checked inspired by the ISO requirements, e.g. “At least 95% (60%) of handover attempts should be completed successfully”.
- Disagreement between outcomes from some of the techniques meant further investigation and refinement of the models was needed:
  - simulation based testing revealed that the robot sometimes dropped the table leg accidentally which was not modelled in the formal verification;
  - real experiments revealed false negatives for the pressure and location sensors not represented elsewhere.
- Some of the techniques were not suitable for verifying some of the requirements, for example for aspects such as speed or closeness formal verification may not be the best technique to use.

Verification of Swarm Robots and Sensor Systems

- A robot swarm is a collection of simple (often identical) robots working together to carry out some task.
- Each robot has a small set of behaviours and is typically able to interact with nearby robots and its environment.
- Usually there is no overall controller and are interested in emergent behaviour.
- Some similarities to (networks of) sensor systems.
- Using robot swarms is appealing in hostile environments e.g. underwater, contaminated areas, or space as they are claimed to be robust to failure of individuals.

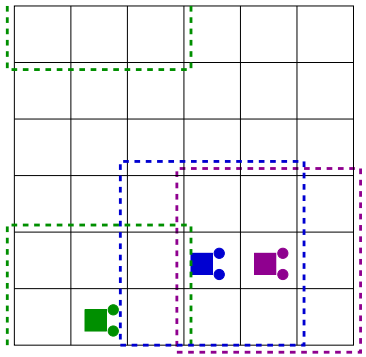


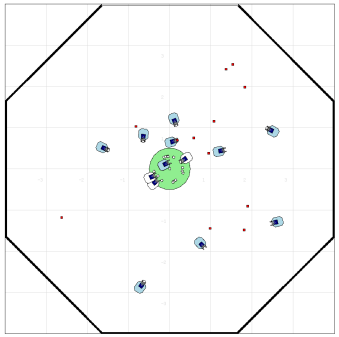


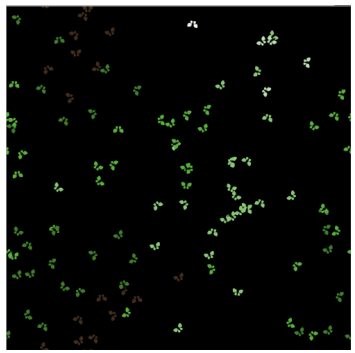


Case Studies

- Verification of the connectedness property of a particular robot swarm algorithm, the alpha algorithm, which makes use of local wireless connectivity information alone to achieve swarm aggregation.
- Probabilistic model checking to a swarm of foraging robots.
- Verification of UAVs as a communication network.
- Verification of synchronisation and gossip protocols used for swarm robots and sensor networks.

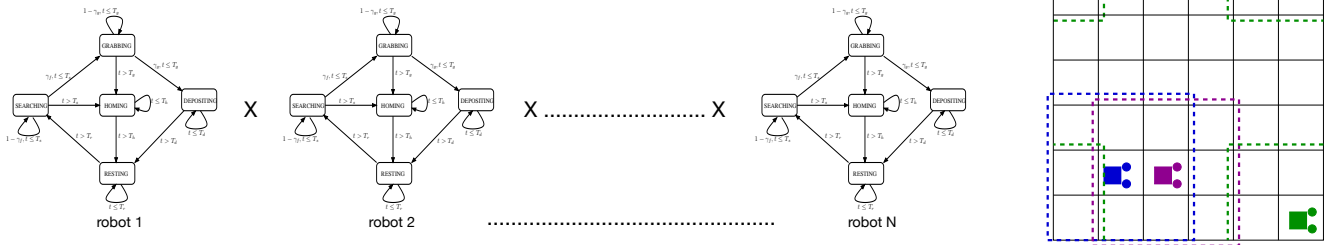






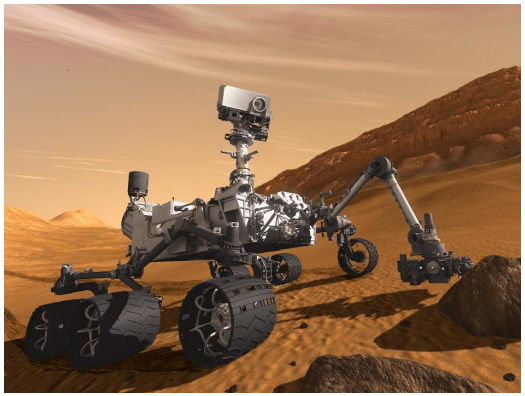
Robot Swarms: Discussion

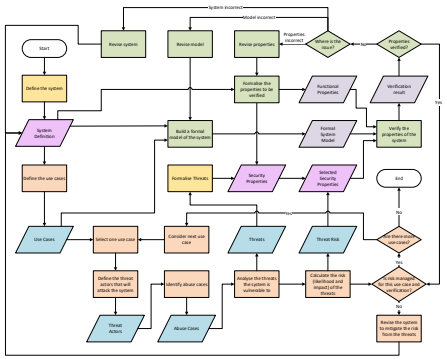
- Whilst the algorithms for these systems tend to be small modelling more than a small number of robots leads to a large number of states (the state explosion problem).
- Population based models can help as long as we don't need to identify each individual.
- How can we be sure that the correct verification of a property for n robots will still hold for $n + 1$?
- Abstractions can help reduce the state space but then counter models must be checked to see whether they represent real issues rather than side effects of this.

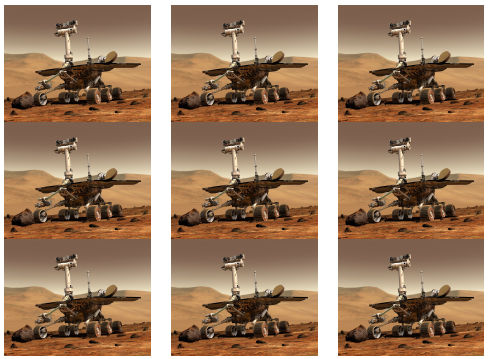


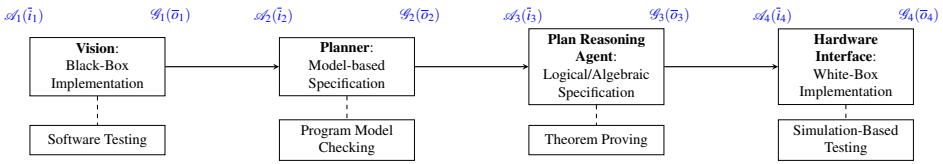
Robots in Hazardous Environments

We are currently developing and applying verification techniques to robotics and autonomous systems in extreme and hazardous environments.









Concluding Remarks: Summary

We gave an overview to the research carried out on several projects approaches to trust, safety, reliability and robustness for robots.

We advocate the use of a suite of verification and validation techniques to help gain assurance of the robot's safety, reliability and functional correctness.

We discussed the combination of formal verification (model checking), simulation-based testing, and user validation in experiments with real robots.

We advocate the use of modular robot architectures and a separation of decision making components.

Requirements are essential so we know what the robot is expected to do. We can use these to derive properties and assertions.

Concluding Remarks: Challenges

Standards and certification: We need to work with regulators to develop standards and routes to certification better suited for autonomous systems.

Design How can we design autonomous systems to facilitate verification?

Environment How do we model uncertain, unstructured environments?

State space explosion: Formal verification suffers from the state space explosion how can we develop and utilise it for such systems?

Learning: How do we verify and certify systems that learn?

Trustworthiness: There are issues of both over trusting such systems and lack of trust.

Thanks to Funders and Collaborators

- Trustworthy Robot Assistants: UKRI/EPSRC funded research grant, `robosafe.csc.liv.ac.uk`
- Science of Sensor Systems Software: UKRI/EPSRC funded programme grant, `www.dcs.gla.ac.uk/research/S4/`
- Future AI and Robotics for Space (FAIR-SPACE) UK Industrial Strategy Challenge Fund (ISCF) and delivered by UKRI and managed by EPSRC. `www.fairspacehub.org`
- Robotics and AI in Nuclear (RAIN): UK Industrial Strategy Challenge Fund (ISCF) and delivered by UKRI and managed by EPSRC. `rainhub.org.uk`

Sample Papers

Webster, M, Western, D, Araiza-Illan, D, Dixon, C, Eder, K, Fisher, M & Pipe, A G
A corroborative approach to verification and validation of human-robot teams, The International Journal of Robotics Research, 39(1), 2020

Gainer, P, Dixon, C, Dautenhahn, K, Fisher, M, Hustadt, U, Saunders, J & Webster, M,
CRutoN: Automatic Verification of a Robotic Assistant's Behaviours, Critical Systems: Formal Methods and Automated Verification, 2017

Webster, M, Dixon, C, Fisher, M, Salem, M, Saunders, J, Koay, K L, Dautenhahn, K & Saez-Pons, J
Toward Reliable Autonomous Robotic Assistants Through Formal Verification: A Case Study, IEEE Transactions on Human-Machine Systems, 46(2), 2016

Gainer, P, Linker, S, Dixon, C, Hustadt, U & Fisher, M
Multi-Scale Verification of Distributed Synchronisation, Formal Methods in System Design, 2020

Dixon, C, Winfield, A F T, Fisher, M & Zeng, C
Towards temporal verification of swarm robotic systems, Robotics and Autonomous Systems, 60(11), 2012

Konur, S, Dixon, C & Fisher, M
Analysing robot swarm behaviour via probabilistic model checking, Robotics and Autonomous Systems. 60(2), 2012

Image Credits

- Robot assistant images from University of Hertfordshire: slides 10, 12, 13, 14, 20, 21
- Robot assistant and swarm images from University of Bristol Robotics Lab: slides 2, 9, 10, 12, 22, 23, 24, 26, 27
- University of Liverpool/Manchester Autonomy and Verification Lab: slide 6, 9, 11, 27, 28
- Nine Micaz sensor motes image from Imperial College, London: slide 26
- GATEway Podcar by Spsmiler available under CC0 1.0 Universal Public Domain Dedication at commons.wikimedia.org/wiki/File:GATEway-Podcar-NthGreenwich-London-P1400407.jpg: slide 2
- Drone Fumigation Agriculture by Herney under CC0 for Public Domain Dedication at www.needpix.com/photo/1120439/drone-fumigation-agriculture-spray-fine-drop: slide 2
- Ocado Warehouse bots by Techwords under CC BY-SA 4.0 https://upload.wikimedia.org/wikipedia/commons/b/bc/Ocado_warehouse_bots.jpg: slide 2
- Industrial Robots by Mixabest under CC BY-SA 3.0 at https://commons.wikimedia.org/wiki/File:KUKA_Industrial_Robots_IR.jpg: slide 4
- Robot vacuum, under CC Copyright-Only Dedication at https://pixabay.com/photos/robot-vacuum-cleaner-carpet-cleaning-5073580/: slide 4
- Robot Mower by Slaunger under CC BY-SA 3.0 at https://commons.wikimedia.org/wiki/File:Robomow_110_City_2012-06-05.jpg: slide 4
- Mars Curiosity Rover, by NASA/JPL-Caltech under CC BY 2.0 at https://www.flickr.com/photos/nasablueshift/7753901656/: slide 29
- Astronaut and vehicle, under CC Copyright-Only Dedication at https://snappygoat.com/: slide 29
- Mars Rover, under CC Copyright-Only Dedication at https://pixabay.com/photos/mars-mars-rover-space-travel-robot-67522/: slide 29