

This is a repository copy of *Assurance Case Process of RoboChart Supported by Formal Verification*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/193121/>

Conference or Workshop Item:

Yan, Fang, Foster, Simon David orcid.org/0000-0002-9889-9514 and Habli, Ibrahim orcid.org/0000-0003-2736-8238 Assurance Case Process of RoboChart Supported by Formal Verification. In: YorRobots and RoboStar Industry Exhibition, 11-12 Oct 2022, University of York. (Unpublished)

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

Assurance Case Process of RoboChart Supported by Formal Verification

What is an Assurance Case ?

A reasoned and compelling argument, supported by a body of evidence, that a system, service or organisation will operate as intended for a defined application in a defined environment.

Can we produce ACs for RoboChart designs automatically?

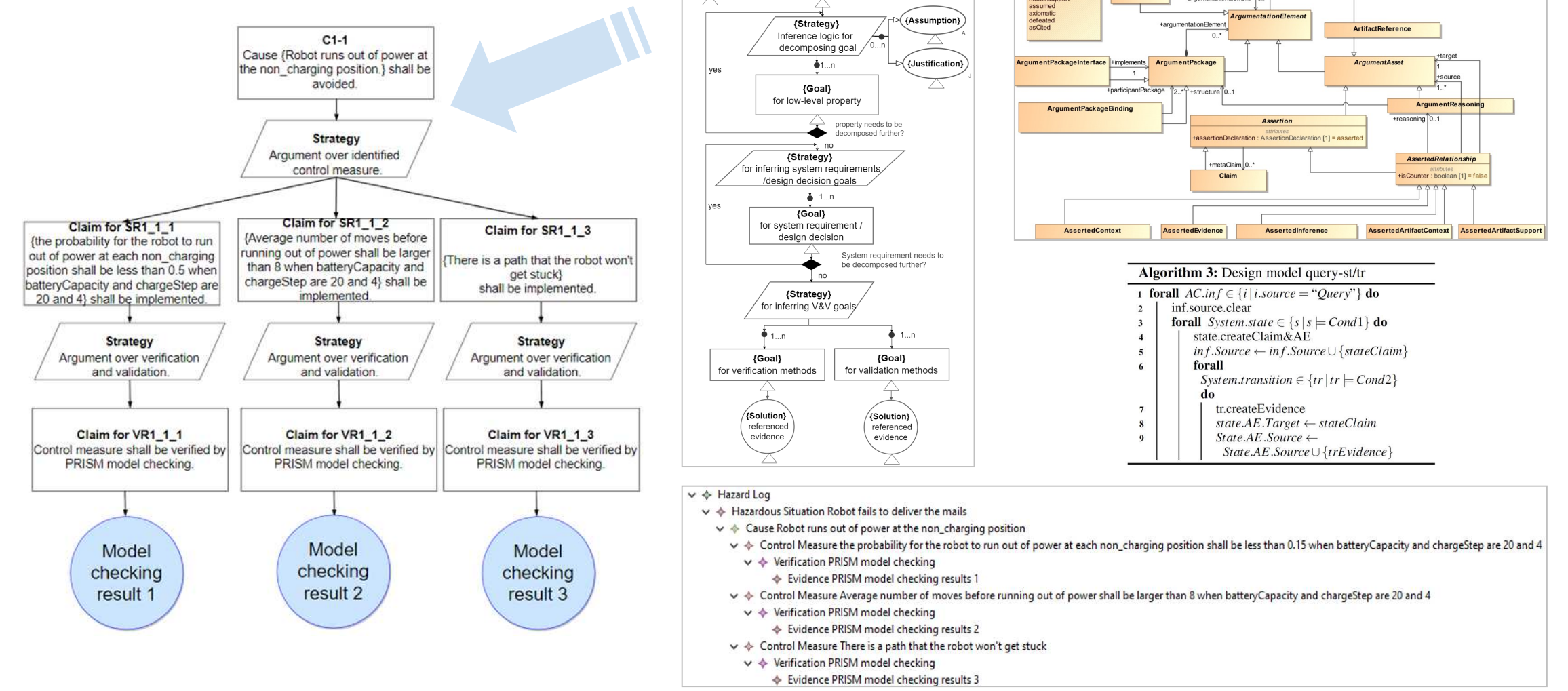
YES. Partially. We provide a model-based solution to (i) generate ACs from RoboChart models and (ii) to generate AC evidence with the formal verification capabilities supported by RoboTool.

What's the advantages of the method?

Formal verification has been applied for producing AC evidence, but formal expertise and manual processes are usually involved. We fill the gap between AC claims and the corresponding AC evidence with an AUTOMATIC process for RoboChart designs.

AC structure and claim generation

- A SACM compliant framework for model-based AC construction from RoboChart models combining the pattern instantiation and model query methods.



AC Evidence generation by model checking

- FDR and PRISM model checking
- Automated RoboChart DSL assertion generation
- Manual input of requirement in structure natural language/template

Results of untimed analysis of assertions in spec3.assertions using FDR

Assertion	Status	Transition	Result
untimed mod_sys_setpoint refines Spec3 (A3) [traces model]	210022	4674325	true
untimed mod_sys_setpoint refines Spec3 (A3_1) [traces model]	19	63	false

Results of timed analysis of assertions in spec3.assertions using FDR

```

untimed csp Spec3 csp-begin
Spec3 = CHAOS[Events] [| (
  mod_sys::ext_pow24VStatus.in.Power_Off |) |
  (RUN(|
    mod_sys::ext_pow24VStatus.in.Power_Off |) |
    mod_sys::ext_setPoint.out.0 -> Spec3)
  csp-end

untimed csp mod_sys_setpoint associated to
mod_sys csp-begin
mod_sys_setpoint = (mod_sys:0_0) | \ {
  mod_sys::ext_pow24VStatus.in.Power_Off, mod_sys:
  s::ext_setPoint.out |}
  csp-end

untimed assertion A3 : mod_sys_setpoint
refines Spec3 in the traces model
  
```

Safety requirement in natural language:

SetPoint shall be set to 0 when the 24V power signal is switched off.

REQUIREMENT TEMPLATE CT-2

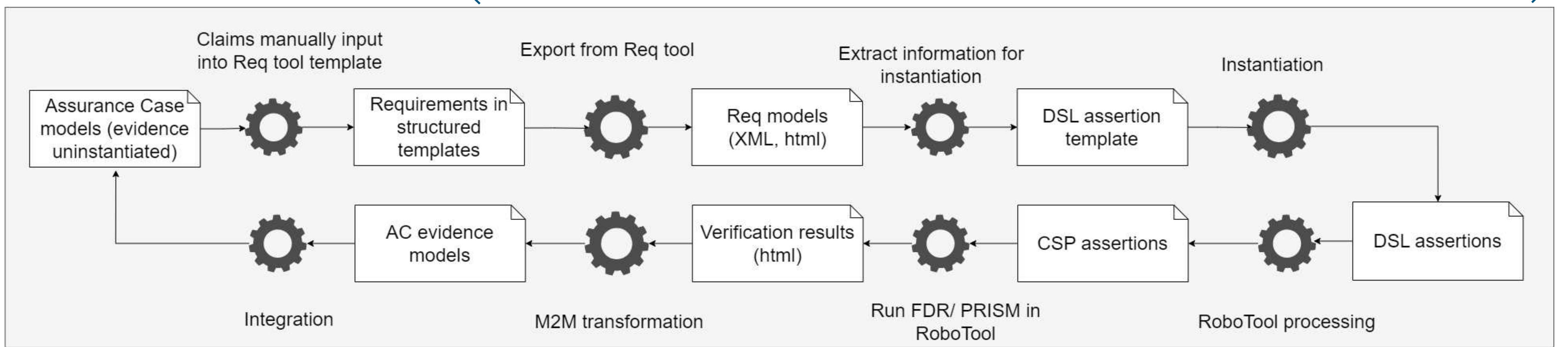
While (guard_event)+ is true, required_event shall be true.
 where
 guard_event = module::event.in/out.value
 required_event = module::event.in/out.value

Data flow of a High Voltage Controller example

ASSERTION TEMPLATE AT-5

```

untimed csp Spec csp-begin
Spec = CHAOS[Events] [| (guard_event |) | (RUN(|
  guard_event |) | required_event > Spec)
  csp-end
untimed csp Spec_impl associated to module csp-begin
Spec_impl = module:0_0 | (guard_event,
  required_event_channel |)
  csp-end
untimed assertion A_Spec : Spec_impl refines Spec in
the traces model
  
```



AC Evidence generation by theorem proving

- Automated transformation from RoboChart to formal notation in Isabelle/HOL
- Automated theorem proving (ATP) in Isabelle/HOL
- Manual formalization of AC claims

