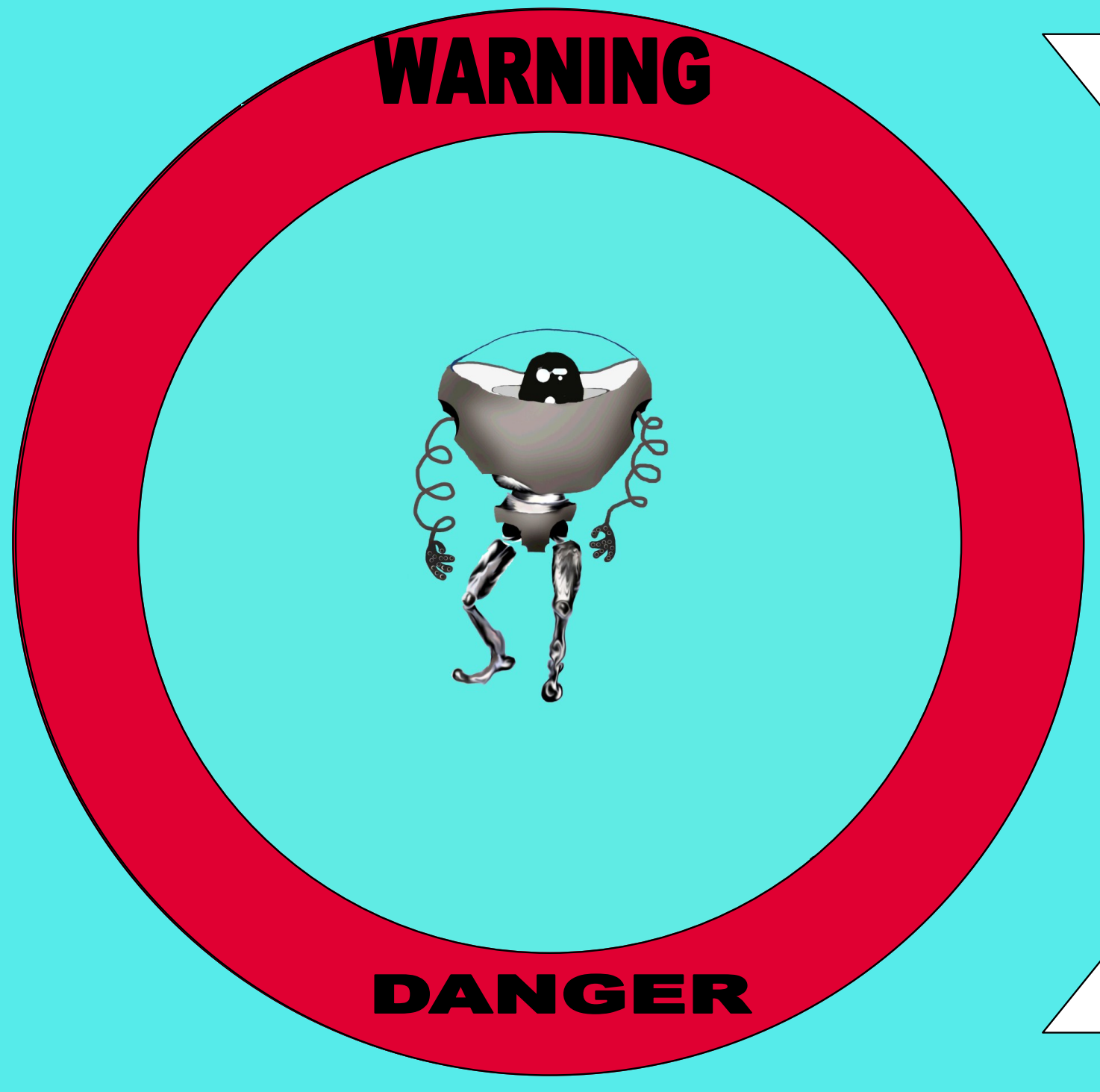
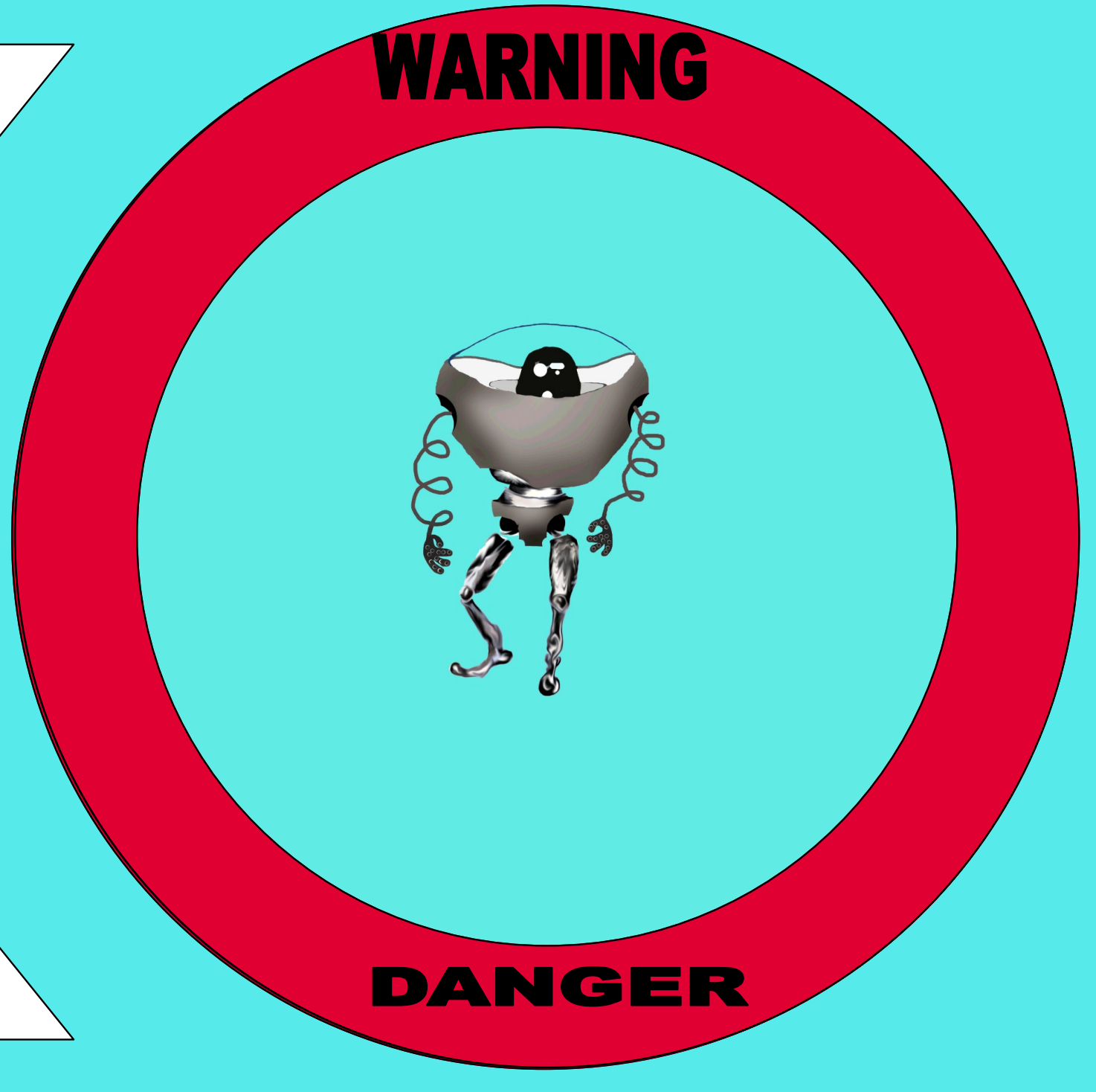
Towards The Formal Verification of Human-Agent Teamwork

By Richard Stocker



ABSTRACT

The formal analysis of computational processes is now a well-established field. Yet, the problem of dealing with the interactions with humans still remains. In this project I am concerned with addressing this problem. The overall goal is to provide formal verification techniques for human-agent teamwork.



Stop!
KILL!!
What the....
HELP!!!!
What happened?
I just asked it to be careful with my mum's vase, else she'll kill me

Dan : You know that robot I was building?
Rich: Yes?
Dan: It misinterpreted my command: "take care of that vase else my mum will kill me" as "Take vase and kill Dan".
Rich: Are you ok?
Dan: Yeah, Lolly was there and pulled its plug. How did this happen?
Rich: Did you not check this would not happen before you switched it on?
Dan: How?!?
Rich: By formal verification. This will check the robot cannot possibly confuse such additional information as a command. It can also check it will never reach a state where it can harm someone. I'll create you a tool which will do this for you, so you know you can trust your robot to work with people. It could even be my Ph.D thesis!

Agent

An agent [3] essentially captures the idea of an autonomous entity, being able to make its own choices and carry out its own actions. Beyond simple autonomy, *rational agents are increasingly used as a high-level abstraction/*metaphor for building complex/ autonomous systems [2]. Rational agents can be seen as agents that make their decisions in a *rational and explainable way (rather than, for example, purely randomly).*

Formal Verification

Formal verification involves analyzing the correctness of a system using mathematical proofs. One technique for doing so is model-checking [1]. Model-checking is the process of checking a specification of the system against a model representing that system. This model takes the form of a finite state machine i.e. a directed graph consisting of vertices and edges. The specification is checked on every path within that model. If the specification fails on any path (representing a potential execution), then this is identified to the designer. A specification could be "If robot believes human is close then it will always reduce its movement speed".

References
1. C. Baier, and J. P Katoen. Principles of Model Checking. The MIT Press, May 2008.
2. L. A. Dennis, M. Fisher, A. Lisitsa, N. Lincoln, and S. M. Veres. Satellite Control Using Rational Agent Programming. *IEEE Intelligent Systems, 25(3):92–97, 2010.*
3. M. Wooldridge. *An Introduction to Multiagent Systems. John Wiley & Sons, 2002.*

UNIVERSITY OF LIVERPOOL