# Adaptable and Verifiable BDI Reasoning

Doctoral Consortium

Peter Stringer
Department of Computer Science, The University of Manchester
peter.stringer@postgrad.manchester.ac.uk

## ABSTRACT

Long-term autonomy requires autonomous systems to adapt as their capabilities no longer perform as expected. To achieve this, a system must first be capable of detecting such changes. In this extended abstract, a specification for Belief-Desire-Intention (BDI) autonomous agents capable of adapting to changes in a dynamic environment is discussed, and the required research is outlined. Specifically, an agent-maintained self-model is described alongside the accompanying theories of durative actions and learning new action descriptions in BDI systems.

## KEYWORDS

Engineering MAS; Learning & Adaptation; Knowledge Representation; Reasoning and Planning

**ACM Reference Format:**
Peter Stringer. 2021. Adaptable and Verifiable BDI Reasoning: Doctoral Consortium. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 2 pages.

## 1 INTRODUCTION

Long-term autonomy requires autonomous systems to adapt as their capabilities no longer perform as expected. To achieve this, a system must first be capable of detecting such changes. Creating and maintaining a *system ontology* is a comprehensive solution for this; an agent-maintained formal *self-model* will take the role of this system ontology. It would act as a repository of information about all the processes and functionality of the autonomous system, forming a systematic approach for detecting action failures.

The work in this thesis will focus on Belief-Desire-Intention (BDI)[10] programming languages as they are well known for their use in developing intelligent agents [1, 4, 6, 8]. Agents that are capable of controlling an array of cyber-physical autonomous systems such as autonomous vehicles, spacecraft and robot arms have been programmed using BDI agents (e.g., Mars Rover[6], Earth-orbiting satellites[4] and robotic arms for nuclear waste-processing[1]). Coupled with their use of plans and actions, BDI languages offer an appropriate platform to build upon for the development of an adaptable autonomous system.

The agent-maintained self-model includes *action descriptions*, consisting of pre- and post-conditions of all known actions/capabilities. An action's pre-conditions are the environment conditions that must exist for an action to be executed whilst post-conditions are defined as the expected changes in the environment made directly by a completed action. These action descriptions are based on the Planning Domain Definition Language (PDDL) [9], commonly used in classical automated planning. The complete availability of current system information will provide the ability to monitor the status of actions, presenting the opportunity to detect failure. *Action life-cycles* based on a theory of durative actions for BDI systems [5] are used to detect persistent abnormal behaviour from action executions that could denote hardware degradation or other long-term causes of failure such as exposure to radiation or extreme temperature. Once a failure has been detected, machine learning methods can be used to update the action description in the self model. Then, actions are repaired or replaced in any existing plans by using an automated planner to patch these plans. The resulting plans can then be verified to ensure the system's safety properties are intact.

*Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.*


## 2 PROPOSED WORK

The overarching aim of this research is to create a framework for the verification of autonomous systems that are capable of learning new behaviour(s) descriptions and integrating them into existing BDI plans: using the framework as a route to certification. In this abstract, the current ability of BDI systems in adaptable reasoning is discussed, largely focusing on actions. Research into Artificial Intelligence (AI) planning on modelling actions is also considered in addition to the methods and implications of introducing machine learning for replacing action descriptions.

### 2.1 Durative Actions

BDI languages are increasingly being used for developing agents for physical systems where actions could take considerable time to complete [5]. Currently, most BDI languages suspend an agent entirely until an action completes or implement actions in such a way that an agent may start a process but then must be programmed to explicitly track the progress of the action in some way.

Introducing an explicit notion of duration to actions will allow for the creation of principled mechanisms to let an agent continue operating once an action is started, meaning the agent is available to monitor the status of actions in progress. [6] introduced an abstract theory of *goal life-cycles*, whereby every goal pursued by the agent moves through a series of states: *Pending* to *Active*; *Active* to either *Suspended* or *Aborted* or a *Successful* end state; and so on. Dennis and Fisher [5] extended the formal semantics provided by Harland et al. to show how the behaviour of durative actions could integrate into these life-cycles. They advocate associating actions not only with pre- and post-conditions containing durations but also with explicit success, failure and abort conditions (an abort is used if the action is ongoing but needs to be stopped) and suggest goals be suspended while an action is executing and then the action's

behaviour be monitored for the occurrence of its success, failure or abort conditions. When one of these occurs the goal then moves to the *Active* or *Pending* (where re-planning may be required) part of its life-cycle as appropriate. Adding these additional states to actions should not add to the cost of model checking as this should not add branches. Adding states should only add more information which would make no significant difference.

### 2.2 Action Failure

The idea of monitoring an action's life-cycle exists in current literature [5–7]. A range of states can be attributed to an action that can subsequently be traced for irregularities or consistent errors, providing a basis for determining failure. If it is assumed that the performance of actions may degrade then the concept of an action life-cycle in which an action is introduced into the system as *Functional*, may move into a *Suspect* state if it is failing, and finally becomes *Deprecated* following repeated failures, is required.

Cardoso et al. [3] assumes a framework along these lines and builds upon it to outline a mechanism that allows reconfiguration of the agent's plans in order to continue functioning as intended if some action has become *Deprecated*. However, this assumed ability to detect persistent failures does not yet exist. This thesis proposes a framework that will allow for the detection of persistent abnormal behaviour from action executions for use with Cardoso et al.'s reconfiguration mechanism.

### 2.3 Current Contributions

So far the main contribution of this research is an initial design of a system architecture [12] for BDI autonomous agents capable of adapting to changes in a dynamic environment, published in the proceedings of AREA 2020, the First Workshop on Agents and Robots for reliable Engineered Autonomy. The system architecture consolidates the agent-maintained self-model with the theory of durative actions and learning new action descriptions into a cohesive and adaptable BDI system.

## 3 RELATED WORK

The work in [3] describes a reconfigurability framework that is capable of replacing faulty action descriptions based on formal definitions of action descriptions, plans, and plan replacement. The implementation uses an AI planner to search for viable action replacements. The proposed research will extend their approach by adding the concept of a self-model, durative actions, and failure detection. Furthermore, future work has scope for adding a learning component to the framework in order to be able to cope with dynamic environment events that require new action descriptions to be formulated at runtime.

Troquard et al.'s work on logic for agency in [13] considers the modelling of actions with durations although a different approach was taken: actions are given duration using continuations from STIT (Seeing To It That) logic. In BDI systems, the focus of handling plan failure is the effect that failure has on goals [2, 11]. This is a reasonable focus considering the central role that goals have in agent-oriented programming. Consequently, action failure recovery has not been explored as an option for managing plan failure.

## 4 FUTURE WORK

A number of questions and challenges have been identified whilst outlining this program of research. Firstly, it has been noted that the term 'persistent failure' is subjective and should be accompanied by a formal and precise specification to avoid ambiguity. Secondly, considerations for the steps taken after reconfiguration and the learning process require further work (e.g. What happens to failing actions in the model after reconfiguring?). Finally, the proposed learning strategy has produced many challenges which will be considered once implementation has reached this stage. Notably, how learning methods can ensure valid solutions; how planning time could be minimised and how an action's state could influence the learning strategy. These challenges will serve as guidance for future work.

Currently, future work includes defining the learning component to be able to handle dynamic environment events that require the creation of new action descriptions (specifically pre- and post-conditions and durations of actions) at runtime, a formal definition of the self-model with an outline of the concepts included in this, the implementation of the system architecture, and the evaluation of the approach.

## ACKNOWLEDGMENTS

The author thanks Louise A. Dennis, Clare Dixon and Rafael C. Cardoso for their supervision and support.

## REFERENCES

[1] Jonathan M. Aitken, Affan Shaukat, Elisa Cucco, Louise A. Dennis, Sandor M. Veres, Yang Gao, Michael Fisher, Jeffrey A. Kuo, Thomas Robinson, and Paul E. Mort. 2017. Autonomous Nuclear Waste Management. *IEEE Intelligent Systems* (2017). In Press.

[2] Rafael H Bordini and Jomi Fred Hübner. 2010. Semantics for the Jason Variant of AgentSpeak (Plan Failure and some Internal Actions).. In *ECAI*. 635–640. https://doi.org/10.3233/978-1-60750-606-5-635

[3] Rafael C. Cardoso, Louise A. Dennis, and Michael Fisher. 2019. Plan Library Reconfigurability in BDI Agents. In *Proc. of the 7th International Workshop on Engineering Multi-Agent Systems (EMAS)*.

[4] Louise Dennis, Michael Fisher, Alexei Lisitsa, Nicholas Lincoln, and Sandor Veres. 2010. Satellite control using rational agent programming. *IEEE Intelligent Systems* 25, 3 (2010), 92–97. https://doi.org/10.1109/mis.2010.88

[5] Louise A Dennis and Michael Fisher. 2014. Actions with Durations and Failures in BDI Languages.. In *ECAI*. 995–996. https://doi.org/10.3233/978-1-61499-419-0-995

[6] James Harland, David N Morley, John Thangarajah, and Neil Yorke-Smith. 2014. An operational semantics for the goal life-cycle in BDI agents. *Autonomous agents and multi-agent systems* 28, 4 (2014), 682–719. https://doi.org/10.1007/s10458-013-9238-9

[7] James Harland, David N Morley, John Thangarajah, and Neil Yorke-Smith. 2017. Aborting, suspending, and resuming goals and plans in BDI agents. *Autonomous Agents and Multi-Agent Systems* 31, 2 (2017), 288–331. https://doi.org/10.1007/s10458-015-9322-4

[8] Viviana Mascardi, Daniela Demergasso, and Davide Ancona. 2005. Languages for Programming BDI-style Agents: an Overview.. In *WOA*, Vol. 2005. 9–15.

[9] D. Mcdermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins. 1998. *PDDL - The Planning Domain Definition Language.* Technical Report TR-98-003. Yale Center for Computational Vision and Control.

[10] Anand S Rao and Michael P Georgeff. 1992. An abstract architecture for rational agents. *KR* 92 (1992), 439–449.

[11] Sebastian Sardina and Lin Padgham. 2011. A BDI agent programming language with failure handling, declarative goals, and planning. *AAMAS* 23, 1 (2011), 18–70. https://doi.org/10.1007/s10458-010-9130-9

[12] Peter Stringer, Rafael C. Cardoso, Xiaowei Huang, and Louise A. Dennis. 2020. Adaptable and Verifiable BDI Reasoning. In Proceedings of the First Workshop on *Agents and Robots for reliable Engineered Autonomy,* Virtual event, 4th September 2020 *(Electronic Proceedings in Theoretical Computer Science, Vol. 319)*. Open Publishing Association, 117–125. https://doi.org/10.4204/EPTCS.319.9

[13] Nicolas Troquard and Laure Vieu. 2006. Towards a Logic of Agency and Actions with Duration. *Frontiers in Artifical Intelligence and Applications* 141 (2006), 775.