Reflective Architectures for Survival: Bridging the Gap between Philosophy and Engineering

Catriona M. Kennedy
School of Computer Science
University of Birmingham
Edgbaston, Birmingham B15 2TT
Great Britain
C.M.Kennedy@cs.bham.ac.uk

Abstract

An essential feature of a mind is its ability to initiate original action in unforeseen circumstances. An artificial agent with this feature should recognise that its software is behaving differently in the new situation and take action if necessary to continue functioning as desired. This requires self-protection and creativity. We address the problem by exploring architectures for whole agents instead of looking for particular algorithms or specialist methods. This means that we are experimenting with patterns of interrelationships between specialist algorithms, without implementing the algorithms themselves in detail. To avoid excessive "shallowness" in the architecture, we use a bottom-up, low-level approach where an agent must "survive" in an environment in which its software can actually be damaged. Reflection enables the agent to monitor its internal execution patterns and detect any deviation from expectations. We are currently investigating distributed architectures where the functionality of an agent is produced collectively by two or more lower level agents which mutually observe and repair each other. First results indicate that the distributed architecture is more robust than a single-agent architecture.

1 Introduction

When we consider the mind's ability to cope with unusual events, there are many situations where it first recognises that there is something unusual about its own internal processing (e.g. a sudden apprehension or increased alertness) due to a still unidentified but novel feature of the environment. We find it is useful to model this as follows:

1. the unusual event is detected by the monitoring of *internal processes* (introspection), and not just by sensing the external environment, i.e. it is at least partly *reflective*.
2. this self-monitoring process is low-level and unconscious, but may be part of the micro-structure of the higher-level process by which we deliberately and consciously observe ourselves, which has been called meta-management (Sloman, 1997).

We are not claiming that this is exactly how a human mind works; instead we use it as a basis for the design of autonomous agents with self-protective capabilities in unpredictable and hostile environments. We believe that this problem must be solved if we are to understand the evolution of higher-level mental processes, which cannot exist as abstract entities, but must have a biological foundation. See e.g. Maturana and Varela(1980). Therefore, we first identify some essential features of a sufficiently challenging environment.

We define a *partially known environment* as one in which events can occur which are not taken into account by the agent's model of the world (based on current knowledge about it) and may include situations which the agent was not explicitly designed to handle. We call these events "anomalies".

We define a *hostile environment* as one in which the agent's executive and control systems do not cope well (e.g. they may be overloaded) or where they may be directly attacked; in other words, its software is subject to interference such as deception, modification or distraction (e.g. denial of service attacks). In these situations, the agent should recognise that its current software is not coping and either ask for help or find some innovative method of working around the problem.

2 Reflective Architectures

To investigate reflective architectures for autonomous agents, we focus on the comparison of whole architectures, instead of particular algorithms or techniques. We may characterise the architecture of an agent as the pattern of interrelationships (causal connections etc.) between entities representing specialist functions (e.g. they may be layers, individual components or sets of functionally similar components). Each entity may be regarded as a "slot" into which a specialist AI technique can be plugged in (e.g. a human interface layer or a planning specialist).

Since our question involves interrelationships, the "slots" of the architecture must be "shallow" (at least initially), otherwise the problem becomes unmanageable. In practise, this means that a slot contains only a minimal implementation of a technique.

However, an architecture that is too shallow may tell us nothing new. An extreme example is a synthetic "personality" that only displays a surprised facial expression in response to an external stimulus.

2.1 Situated reflection

We attempt to find a middle ground between shallow architectures and specialist techniques by ensuring that mentalistic concepts are given a concrete interpretation in an engineering sense. Therefore, we do not *simulate* a challenging environment, but ensure that it happens in reality, so that conditions are more similar to "survival" in the real world.

We implemented the requirements for a hostile, partially known environment as follows:

- *Hostile environment*: The agent is attacked at random intervals by a fault-insertion agent; faults may be produced in any part of the agent, including its self-monitoring and repair components.
- *Partially known environment*: The agent has no knowledge of the kinds of damage that can occur; instead it must detect anomalies in its software execution patterns as deviations from expected patterns. Then it must attempt to repair the damaged software or take evasive action as necessary.

To detect anomalies in its execution, it must have a model of its own normal operation, which is acquired gradually by self-observation during a protected "development phase". A simple type of model is that of a *signature* which is a collection of "normal" patterns of activity (Forrest et. al. 1994). We initially used a simple signature which required certain patterns to be present in execution traces.

2.2 Distributed reflection

The simplest form of reflection is a two-layered architecture containing a meta-level which monitors the agent's software execution patterns. The configuration where the meta-level is also applied to itself is shown schematically in figure 1(a). The whole agent is labelled A (the meta-level is not shown).

There are situations where the meta-level will not detect anomalies in itself (e.g. it cannot detect that it has just been deleted); in other situations it is unreliable (e.g. if the anomaly-detection process has been modified so that it gives false alarms and does not detect its own anomalous execution patterns). An alternative is a distributed architecture where the functionality of the first agent is produced collectively by two or more lower-level agents. Figure 1(b) shows this configuration for two agents (labelled A1 and A2). Figure 1(c) shows a possible three agent configuration. Each agent's meta-level not only monitors the agent's own software but also the meta-level of at least one other agent. In this way all meta-levels are protected. For related work, see Kornman(1996).

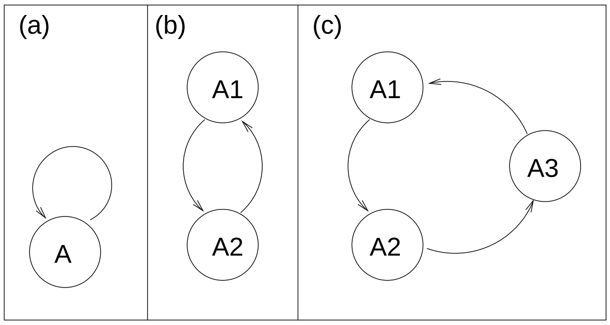

Figure 1: Reflective architectures

We are investigating the practical feasibility of a distributed architecture. Our first results indicate that it can provide survival advantages over a non-distributed version, but it requires a complex model acquisition process, involving higher-level category-formation. Details of the latest results can be found in Kennedy(2000).

References

S. Forrest et al. Self-Nonself Discrimination in a Computer. *Proceedings of the IEEE Symposium on Research in Security and Privacy*, 1994.

C. Kennedy. Reflective Architectures for Autonomous Agents. School of Computer Science Technical Report, April 2000.

S. Kornman. Infinite Regress with Self-Monitoring. Proceedings *Reflection'96*, San Fransico, April 21-23, 1996.

H. Maturana and F. J. Varela. *Autopoiesis and Cognition: The Realization of the Living*. D. Reidel Publishing Company, Dordrecht, 1980.

A. Sloman. What sort of architecture is required for a human-like agent? *Foundations of Rational Agency*, edited by Wooldridge, M. and Rao, A., Kluwer Academic Publishers, 1997.