TOWARDS AUTONOMOUS DEPENDABLE SYSTEMS

Tope Omitola
too20@cam.ac.uk

DJ Greaves
djg@cl.cam.ac.uk

University of Cambridge, Computer Laboratory

Abstract

Today's modern society is dependent on the internet and other information systems. But our current information systems need to be more reliable and better secure before society can put their trusts in such systems. A trustworthy dependable system must have autonomous behaviour integral to it. Our current project, at Cambridge's Computer Laboratory AutoHan group, is building autonomous behaviour into everyday objects. We are applying AI Planning technique with Partial Evaluation to solve the perennial problem of how to build dependable trustworthy systems.

Introduction

It has been said that "complexity is the business we are in, and complexity is what limits us." [1]. The IT industry has spent many years building systems of ever-increasing complexity. We live in a world where complex systems are being composed and networked together. And we live in a highly mobile society where environments and conditions are continually changing. As a result, our complex systems break down whenever the environments around them change, because of the brittleness that we have un-intentionally built into them.

What are needed are systems that can continuously configure and re-configure themselves under varying and predictable conditions, and that can dynamically monitor and tune the resources available to them. We believe that building autonomous behaviour into our systems will make them better able to adapt and respond to changes around them.

Dynamic Adaptability

An autonom*ous* system is different from an automated system. An autonomous system takes sophisticated system-level decisions, deals with (unexpected) situations that an automated system cannot deal with, while an automat*ed* system takes low-level, mechanical decisions and is designed for a limited class of situations.

Here at Cambridge Computer Laboratory's AutoHan group, we use AI Planning and Partial Evaluation techniques to provide autonomous behaviour in the everyday objects around us in the home.

A.I. Planning [2] is a key enabling technology for intelligent systems. A.I. Planning is used widely to build control algorithms enabling a system to generate actions to achieve its goals. A planner automatically generates plan of actions that achieve goals while obeying resource and operations constraints, and it continuously revises plans in response to events, thereby leading to better adaptation, robust fault protection, and more efficient resource utilisation. Partial Evaluation [3] is widely used in the programming language community for program specialisation. A partial evaluator takes a program *P* and a goal *G*, and derive a program *P`* specialised with respect to *G*.

The AutoHan group at Cambridge are building an architecture that combines model checking with partial evaluation. We have built a model checker, which we use as a planner, that generates actions in response to goals and/or contrapositions given it. And we have partial evaluators that take the outputs from our model checker, spawn these and activate them to control the devices in our home network.

Model checking [4] is a formal verification technique which is highly popular in industry. In model checking, a system is represented as a finite state machine (FSM), and checked if it satisfies some properties. The checking is done with the aid of model checking algorithms which traverse the system model. Model checking has also gained considerable success as a planning tool, for example, Cimmatti et al. [5] used a model checker, as a planner, to generate paths for robot traversals.

Our test bed for these ideas is a UPnP-home area network [6] which connects together a variety of UPnP-enabled devices. Each device in our network has its properties, and the services they are hosting, described in XML. Our model checker has a model of these devices and services. The variability of the environment is encoded as input actions and the services we want to activate are encoded as goals to our model checker. The variability in the environment includes factors such as differences in device capabilities and network bandwidth. The model checker responds by traversing the states of the model generating actions that satisfy our goals. The generated actions are services we want to activate. The partial evaluator turns each generated action into imperative codes using the variability of the environment as the dynamic input. These codes are then activated on the appropriate home devices.

Conclusion

Our vision of the home of the future is one where the devices in the home will be more responsive to changes in our environment giving better quality of service to simplify our lives. To achieve that, these devices will need autonomic behaviour built into them. We believe that the combination of planning and partial evaluation will make this possible. We are currently testing our architecture using a variety of UPnP-enabled devices.

References:

1. F. P. Brooks. *The Mythical Man-Month*, pub. Addison-Wesley, 1995.

2. S. J. Russell and P. Norvig. *Artificial Intelligence*, pub. Prentice Hall, 1995.

3. N. D. Jones, C. K. Gomard, and P. Sestoft. *Partial Evaluation and Automatic Program Generation*, pub. Prentice Hall International, 1993.

4. E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*, pub. The MIT Press, 1999.

5. A. Cimatti, E. Giunchiglia, F. Giunchiglia, and P. Traverso. Planning via Model Checking. In *Proceedings of the Fourth European Conference on Planning*, Springer-Verlag, 1997.

6. UPnP. http://www.upnp.org.