Exystence
European Complex Systems Network of Excellence

COMPLEXITY WORKSHOP
HP Grenoble, Auditorium
November 28th 2005

Self Managed Systems - A Control Theory Perspective

RICHARD TAYLOR, Principal Scientist, Model Based Analysis Group, HP Laboratories.

Introduction
Self-managed systems are essentially closed loop control systems. For any control system, slow convergence, oscillation, chaotic behaviour or stuck modes are undesirable. It is argued that control functions and compositions should be restricted to those with known 'good' properties whose functional space can be demonstrated within cellular automata.

Background
I want to take an approach to complex and complicated systems which is highly utilitarian. I started to become interested in emergent behaviours about 16 or 17 years ago, when I became aware of things such as cellular automata which seemed to be quite interesting models of computation. Subsequently I developed switching systems that exploited particular properties of cellular automata to make them more amenable to dependable systems analysis. Over the last five years I have split my time between Hewlett Packard laboratories and Hewlett Packard services solving problems for customers.

The need for automatic control
I want to say something about complications versus complexities and the approach that I'm going to talk about also relies on work by one of my colleagues, Chris Tofts. We are already well aware that constructed systems, even small systems, have become more difficult for human beings to manage. For example, I recently spent time with two large European banks, attempting to work out how they can demonstrate Sarbanes-Oxley compliance and I came away thinking that the problem was an even bigger problem than I had previously thought.

Removing control from humans is a very rational thing to do for several reasons. Human beings have a very limited response rate, they are not good at repeated operation (which is one of the few things that computers are very good at) and they are expensive to maintain. You need a lot of them to get your response rate up, they are difficult to train and so we seek to replace them wherever appropriate.

Possibilities
What can we replace them with? We have to replace them with something that manages. To some extent every human being is a self-management system and it is worth reminding ourselves just what we mean by 'a self-managed system'. In engineering terms humans are closed loop control systems. They make an observation which tells them something about their existing state and they act on it. The observation does not tell the person everything about his or her person because that would be impossible. It gives them some information on which they can choose to act or not, but the information is limited and there are things over which they have no control. In the same way constructed self-managed systems have some ability to control some of the aspects of their behaviour. For example, the autopilot in an aircraft can control some aspects of the aircraft's behaviour but it cannot control the external environment.

Human beings demonstrate an ability to compute how to get from one state to another. Whether that is an implicit or an explicit set of operations need not worry us at the moment, but the important point is that conventional control systems require, if they are engineered, someone who is able to demonstrate with some degree of certainty, that what is achieved is what was intended. There is no point in putting in a control system that has such poorly defined properties; you do not achieve the desired end result. Once upon a time the systems that I dealt with were reasonably small real-time systems, now they are big banking systems, big managed service projects, so it is quite important that we understand the nature of the controller.

Requirements
The controller is the heart of a self managed system so it is worth asking what the properties of a good controller are. You need to have an ability to converge on a required solution over the whole input space. Someone mentioned the 'edge of chaos' in complex systems. I actually hate talking about the edge of chaos because I want to be well away from that kind of boundary, so that I can predict what properties are going to be within the input range of interest. That also requires adequate responsiveness, so I don't want to have to say that it takes five seconds to perform an action now, but two and a half hours at some point in the future.

We need to understand something about the stability of the system for stable inputs. We do not want unnecessary oscillations or snap-overs on small changes. Generally, I think most control engineers would say that this is pretty well understood; give them a few differential equations and they are as happy as Larry. The question I would like to ask is, what should replace this approach if we're going to derive control system properties in a complex system? If you have something with a complex nature that you drop into the system how do you understand and constrain the consequences? That is a fundamental question that has to be asked. The man on the moon has a somewhat different appreciation of his susceptibility to failure than somebody playing with a toy robot back on Earth. We need to ask ourselves the question, 'should we, philosophically, morally and practically, employ control systems which we cannot analyse?' Should we insist that we restrict ourselves to mathematically tractable controllers as things that are understood? Talk to the engineers making air buses and they are going to say, 'we want to be able to analyse the system'.

Approaches

If we take the air bus industry approach we want to understand the system we are trying to control, understand enough about the algorithms that we are using to be able to justify with a high degree of confidence that they will achieve stability. It will not always happen and there are going to be blackouts. In the case of air buses, for example, there are going to be problems with wind shear on runways. That means we have to introduce additional meta-control in order to prevent an aircraft attempting to land outside the parameters of its control system.

That is one approach, but another is to try to limit the response function to stable areas. Cellular automata (CA) are essentially simple, locally connected structures which, although not well understood, have certainly been explored by many people. Cells have simple transition rules based on their recent states and the states of neighbouring cells. In other words they communicate with their neighbours and do something based on what their neighbours did at some stage in the past. The 'game of life', which most people have seen, is the classic computer programme based on that principle. Cellular automata are often considered to demonstrate emergent behaviours because some of the patterns that emerge are extremely complicated. Sometimes they are chaotic and can be widely used as models for load balancing on parallel computer systems or other communications systems.

System behaviour

We can construct CA-like objects using computers which communicate with each other and such systems can exhibit four different basic behaviours. I use the term 'descriptive behaviour' lightly, because the distinctions are more like 'lies to small children' in the way that we describe what atoms are like in a child's first year at secondary school. So the system could evolve to an homogenous state where everything just flattens out. That would be one kind of behaviour. Then we might get evolution to some simple periodic structures, waves for example, continuously propagating at a constant rate through the medium. After that we might get more interesting aperiodic behaviour which is chaotic or we might get the generation of complex patterns of very localised structures which, for one reason or another, do not propagate.

Design approach

There is a backward or forward way to design. The forward way is not difficult and observation-based analysis gets you quite a long way in finding useful systems. The backward way however, is when the engineer has has a client who says, 'I want you to design a locally connected object that is going to behave in this way'. For many cellular automata you can only use the forward approach. It is analysis by trial and error because you have massive behaviour state spaces. You do not necessarily know how much of a state space you have looked at and you end up with a system which has spaces which you haven't explored, but you should do so in order to get the design right.

There are massive implications of, for example, a domino effect. About eighteen years ago there was a massive US East Coast telecommunications failure. One or two switches failed, cascade faults resulted and Bell telecommunications services were taken out for the best part of nineteen hours. There have also been more recent combinations of power and control cascade faults which have had a devastating effect on the United States. For modern communications infrastructure you now have to demonstrate the ability to damp oscillations if you have overload or failure.

If we take the dependability-based engineering approach to design then construction of systems requires an understanding. If we do the same as many people are doing in the complex systems world, which is setting up simulations and hoping that one of them is going to give a reliable, repeatable structure, then we will not satisfy the airbus people. We would be laughed out of court if we tried to implement such a controller in a large financial house because many large systems are simply too important to leave open to chaotic behaviour. The systems that we are now working on generate 70% of the UK gross domestic produce through service economy type activities. 18 or 19% of that is dependent on highly integrated human computer processing and physical controllers. If they fail it is very expensive and everybody gets extremely upset. We do not hear much about these kinds of commercial failures is because neither the supplier, nor the person whose business has obviously suffered through the development of inefficient or unreliable systems, wants to admit something that would affect their share price.

The other big problem that we have, especially in the biologically inspired computing world, is that people mistake an algorithm for something that can necessarily be implemented efficiently in a practical sense. The example that I always love is the so-called travelling salesman problem, being solved efficiently by ants. It is actually not the traveling salesman problem but a minimum spanning sub-tree problem, but leaving that objection aside, the point is that ants and silicon do not work the same way. If you want to solve the problem in silicon you have got to do it in a different way. Employing the art of the possible means that you can use a substrate and make the algorithm amenable to implementation within a real system, but you have to be very careful about it.

The challenge

I think that the challenges of specifying, building and managing such systems at the moment are massive. We are seeing constructed systems of the most complex kind that the world has ever seen. The connectivity in them means that you cannot leave them alone to develop undesirable emergent behaviour. You need to identify boundaries; control them as well as you can and understand the probabilities involved in different failure or success modes. You have to bet appropriately. I do think that the exploitation of large-scale emergent phenomena has got some role to play, but I also think we have to be very careful about not getting carried away and promising the earth too soon. The artificial intelligence (AI) community did that. People took enormous sums of money from the Government and the Military and, though there were some early successes, people soon began to realise it was more complicated than they first thought. The approach that we have been taking within HP comes back to understanding a control system because (a) we would rather make money than lose money and (b), our customers need to be able to rely on the systems that we construct, manage and ultimately have to take down or migrate to some other form.

Our approach is via something called Systems Sciences which in its philosophical approaches, appears to be somewhat similar to complexity science. It says at one end we have quantum chromo dynamics and at the other we have to explain why Fred will go over and have a cappuccino with extra milk in two days time. We have to take the notion of control systems seriously, because all of our businesses are effectively control systems, but we want to understand what we prod, why do we prod it and how can we prod it in order to move it in the direction that we wish. We will then say that we have services, of which many aspects are truly complex.

We do, however, need to draw a boundary round complex behaviour and put into place appropriate recovery mechanisms, so we can move back into stable operating regimes. We have to reason about different aspects of the systems, whether they be people or organisations and know where the limits are. We can do research on social interaction and knowledge management as Barnardo Huberman is doing. If we wish to achieve X then we can apply a scientific theory, set up an investigation and look at the results. We might observe feedback in the system and, depending on whether it was good or not, we might achieve management at some level. The other way to exert control is to plug wires together, configure microprocessors, decide how many disks we are going to put in and decide what level of redundancy we need. That way we can be much more precise, but we do have to accept there is going to be a continuum of different types of science, and we have to be able to put boundaries in and understand how flexible those boundaries should be.

Discussion

Questioner 1. How generalisable are such controlled systems?

Richard: That is a very important question. Half of my work is spent in HP laboratories and half is spent with customers in HP services. Some of that work is extremely generalisable and we have been encouraging our systems engineers to apply appropriate mathematics and appropriate social engineering methodologies. We believe at the moment about 80% of the problems we experience can be solved by relatively low-level training and relatively low-level exposure to basic techniques, which means anything from basic q-theory at one end to organisational theory plus Systems Theory.

We are stumped by about 20% of the problems that we come up against in attempting to bound complexity, or attempting to understand the dynamics of particular systems. We can get a bright mathematician economist to explain the system as a one-off, but attempting a generalisation and then moving back from that general explanation to something new is extremely difficult. Philosophically this has always been a problem with engineered systems. It very easy to explain a specific instance of something, but generalising becomes more difficult. From the other end computer scientists have some very abstract generalisations which become completely intractable as soon as you attempt to scale them up and apply them to something real. It's not an easy problem.

We are learning a lot about the way we should be applying this work and we have some quite distinct categories of system engineering. We also understand more about the need to allow multiple stakeholders in a system to understand the impact they have on it, or what one particular part of the system has on other parts. You learn as you go along, but all too often in big systems projects, you get a consultant coming in and saying, 'you have to re-engineer the system'. They then throw in some specifications and say, 'go and build it' and in that way you lose information. You lose the ability to link effect at one end to possible cause at the other.

One of the things that emerged from the work we have been doing at HP is called rapid scenario planning. It is a formal mathematical underpinning of scenario planning discussions that uses language appropriate to different groups of stakeholders. They can then assess whether they actually understand the system, understand whether they pass the right information forward and whether they're getting the right information back. It enables them to assess the agility requirements of the system that they are going to put into place and it has led to different approaches in the way that we are building things.

Questioner 2. How often do you hear customers complaining about IT complexity and what do you say to them?

Richard: One of the reasons that there are fewer customers lamenting about IT complexity is that they are outsourcing it to us, but of course that means we have to solve the problems. As soon as you move from simply selling people servers to selling a more complicated system, in which there are complex stacks of both basic software and business logic, and you write notional business objectives into the logic, you begin to see the engineering teams lamenting it.

Questioner 3. You talked about different systems that have controllers and the realities of intervening in such systems, but you did not tell us the purpose of their design. What can we learn from this design experience and translate into best practices or design practices? At the moment we just design our particular systems and instrument them to make them self controllable so that they stay within the design boundaries.

Richard: If anyone is interested there is an open paper on the HP Labs website called *Business as a Control System*. What we have been advocating is that if you start off thinking about the systems as control systems to begin with, you begin to do things properly. HP has always been a fantastic instrumentation company and we can measure anything about a system, but you have to know what the measurement data means. If we are going to invest in a country in order to increase the prosperity of that country, we do not start by measuring the amount of traffic on the roads. It may be an indicator of greater prosperity in one area or greater commerce in one area, but if we simply look at that data and say reducing congestion will maximise our return, then we will not get very far.

We have to look higher up and start taking a proper economic view. If we think about the system as an economic control system, we begin to see that underneath that there is a whole hierarchy of other control systems, all of which need to be appropriately instrumented. If you start with a top-down abstraction and the motto, 'design top-down, build bottom-up' which has always been a favourite saying of mine, then you begin to get proper root cause analysis. That means you begin to understand how what is happening below impacts on your high-level objectives.

I believe that many of the large successful service organisations are now taking a control-based approach in which they are saying:

- Here are my business objectives, some of which mean that I understand my agility.
- Here are the sets are things that I am going to need to understand at the business level.
- Here are sets of things that I'm going to have to understand the process level.
- Here are the sets of things I'm going to need to understand at the IT level.

If you build the appropriate abstraction interfaces you have the ability to rip one layer out and replace it with another or perhaps run one concurrently with another without having to disturb too much of your system.

Questioner 4. This is a question about the distinction between a complicated and a complex system. I heard recently that in a car there are 100,000 potential variables. At what level do you describe the system? Do you think that the distinction can be made at the level of the controller?

Richard: Yes partially, I think the distinction is at the level of understanding of the interface. If an LED fails on the car dashboard it doesn't make the car turn over and crash. What I have is a set of systems of systems in which there are very carefully controlled interfaces. I am not reliant on understanding the explicit state of the whole vehicle in order to understand the primary modes of behaviour. But one of the issues about many so-called complex systems is that we don't understand the boundaries or haven't been able to put the boundaries into place. That makes it extremely difficult to reason in any sensible manner about the whole system.

Questioner 5. I think it is a case of discovering a layer of abstraction that is relevant. You have to care about your tyre pressures, but you don't need a whole theory of how rubber reacts against the pavement, and anything that fails does not necessarily crash the car. A full spectrum of dynamics from micro and macro might be an interesting intellectual exercise, but I wonder if that is useful when you try to make these things do something?

Richard: I think it is more a case of necessary 'lies to small children'. It is hard to teach eight or nine year old children about physics. Suddenly solid objects apparently have lots of space in them containing little balls. But that explains a bit more about why things do what they do. Then these little balls are made of the other little balls with others whizzing round them and that explains a bit more. When you get to University you realise it was all lies, but understanding building blocks allows a child to understand the castle that Daddy is building. If they need to understand how to construct a material that is going to be tough and chewable, they might need to take a slightly more material science orientated view. If we ask what the appropriate level of abstraction is in these models then it is like asking what the appropriate lies to small children are. I think our deep understanding of biology might actually have been damaged and held back by the fact that we have advanced our understanding of physics so far. I think 'lies to small children' is an appropriate way to go in treating these complex systems as systems of systems.