E E E E E A E

Gr s r n Co n l
n Co p n n s ss

C 1

ors
r Bo r r
on n
A r n o pson

D r

Contents

Dedication

In: de Bourcier, Lemmen & Thompson eds., 1994

1. *Causation:* the mechanisms underlying that behaviour. These include the triggering environmental stimuli and the cognitive/neural/hormonal processes active in the animal.

2. *Development:* the ontogenetic sources of that behaviour. For example, birds often learn their mating songs from their parents.

3. *Evolutionary History:* the phylogenetic development of that behaviour. For instance, in principle, it should be possible to trace out the route by which initially incidental movements or responses, on the part of certain animals, were modified over evolutionary time to become ritualized, stereotypic signaling patterns.

4. *Function:* the adaptive consequences of that behaviour. That is, the behaviour is investigated in terms of the role it plays in contributing to the survival and reproductive prospects (Darwinian fitness) of an animal. For example, if we want to understand *why* the female of the scorpionfly, *Hylobittacus apicalis*, mates for longer with males who woo her with larger insects as courtship gifts, then we had better identify the contribution made by that behaviour to the female's adaptive success. In fact, it appears to be because the female's capacity to produce eggs is limited by the food

te Boekhorst and Hogeweg (1994) used a simulated eco-system, based on a natural habitat at Ketambe, to investigate the formation of travel parties in orang-utans. The results from the simulation suggested the hypothesis that these travel parties were emergent propert between ecology and behaviour. Our claim is merely that SBE provides a new way of asking old questions and, in time, has the potential to find some new questions to ask. The following paper in this collection describes an example of SBE at work.

r n s

Beer, R. (1990). *Intelligence as Adaptive Behavior: An Experiment in Computational Neuroethology*. Academic Press, San Diego, California.

Cliff, D. (1991). Computational neuroethology: a provisional manifesto. In Meyer, J.-A., & Wilson, S. W. (Eds.), *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pp. 29–39 Cambridge, Massachusetts. M.I.T. Press / Bradford Books.

Cliff, D., Husbands, P., Meyer, J.-A., & Wilson, S. W. (Eds.). (1994). *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, Cambridge, Massachusetts. M.I.T. Press / Bradford Books.

de Bourcier, P., & Wheeler, M. (1994). Signalling and territorial aggression: An investigation by means of synthetic behavioural ecology. In Cliff, D., Husbands, P., Meyer, J.-A., & Wilson, S. W. (Eds.), *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pp. 463–72 Cambridge, Massachusetts. M.I.T. Press / Bradford Books.

Grafen, A., & Johnstone, R. (1993). Why we need ESS signalling theory. *Philosophical Transactions of the Royal Society: Biological Sciences*, *340*, 245–250.

Koza, J. R., Rice, J. P., & Roughgarden, J. (1992). Evolution of food-foraging strategies for the caribbean *anolis* lizard using genetic programming. *Adaptive Behavior*, *1*, 171–200.

Krebs, J. R., & Davies, N. B. (1987). *An Introduction to Behavioural Ecology* (2nd edition). Blackwell Scientific, Oxford.

MacLennan, B., & Burghardt, G. (1994). Synthetic ethology and the evolution of cooperative communication. *Adaptive Behavior*, *2*(2), 161–188.

Maynard Smith, J. (1982). *Evolution and the Theory of Games*. Cambridge University Press, Cambridge.

Miller, G. F., & Cliff, D. (1994). Protean behavior in dynamic games: Arguments for the co-evolution of pursuit-evasion tactics. In Cliff, D., Husbands, P., Meyer, J.-A., & Wilson, S. W. (Eds.), *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pp. 4199 0 Td [(,)5.441 -13.5.441 -13.5.I

or or on g r du to Maynard t and Harp r and s g na n g as a b av or p r or d by a g nt X o ac surv v n an at *Moving* an an at ov s n any d r ct on t ncurs a ov nt cost *Repro-ducing* I an an at ac v s a n r y v t n t w as xua y r produc - o spr n an on y c d p ac d rando y n t wor d s v n t sa n t a n r y v as ac b r o t popu a t on ad at t start o t run and t corr spond n a ount o n r y s d duct d ro t par nt And na y *Fighting* F ts occur w n an ats touc - uc p ys ca co bat r su ts n a ar r duct on n t n r y v s o t part c pants-

I an an at s n r y v s n s to t n t at nd v dua s d d to av d d- nd r t s c r cu stanc s t s r ov d ro t wor d- as w as ncr as n t rou b rt s popu at on s z can d cr as t rou d at s-

Du to t act t at an ats os n r y on a r u ar bas s ood nd n s an ss nt a tas - o ncour a ora n b av or ac an at as a un r v ts d spos t on to ov towards ood w c s updat d at ac t st p- ur s ca cu at d by add n a constant to t d r nc b tw n t ax u nd v dua n r y v poss b - t va u at w c r product on ta s p ac and an an at s curr nt n r y v - us an an at s un r s nv rs y proport ona to ts n r y v -

xc ss o t r actua a gg r ss on t ost xtr b u rs ar t ost succ ss u g roup - t y ar
t ost adapt v t und r t s sp c co o g ca cond t ons- In act a t r t n t a s tt n g down
p r od o approx at y t st ps t H g ro

ta p Daw ns M- - Ar t r n ra pr nc p s o s na d s n - *Philosophical Transactions of the Royal Society: Biological Sciences* H H

r M and d Bourc r - - How not to urd r your n bor s n synt t c b av ora co o y to study a r ss v s na n - ub tt d to t ourna *Adaptive Behavior*-

Za av A- H - Mat s ct on a s ct on or a and cap- *Journal of Theoretical Biology* H H -

In: de Bourcier, Lemmen & Thompson eds., 1994 *he eventh h te House Papers Graduate esearch n the Cogn t ve Co put ng c ences at ussex* University of Sussex, School of Cognitive & Computing Sciences, Brighton, UK. Research Paper CSRP 350.

Dynamic Fitness Landscapes

Seth Bullock
sethb@cogs.sussex.ac.uk

School of Cognitive & Computing Sciences
University of Sussex
Brighton
BN1 9QH

Abstract : Genetic Algorithms (GAs) are typically thought to work on static fitness landscapes. In contrast, natural evolution works on fitness landscapes that change over evolutionary time as a result of co-evolution. Sexual selection and predator-prey evolution are

2 Landscapes

Optimisation techniques are often thought of as traversing

drives the female's mate preferences and the associated male traits, and in many cases drives them in such a way as to create organisms which have deviated from their strictly survival oriented ancestors in an attempt to satisfy the constantly changing demands of sexual selection.

Sexual selection is by no means the only example of such co-evolution. Predator-prey evolutionary dynamics also exhibit what behavioural ecologists have termed 'evolutionary arms races'. The development of higher acuity in a predator may be countered by the evolution of camouflage in a prey, teeth and claws provoke carapaces and scales, toxins demand antidote from scratch. Such scaffolding techniques are reminiscent of the parent-child interactions which facilitate infant development (Rutkowska, 1994).

However, the hand-cranked nature of such scaffolding requires the presence of a human designer ‘in the loop’ and, potentially, the tasks of specifying the incremental goals that allow evolution to reach solutions to complex problems could itself become as problematic as designing the agents manually.

First attempts at utilising automatic co-evolutionary design includes work by David Hillis (1990) and Phil Robbins (1994), in which parasites are used to increase the performance of artificial agents, and Phil Husbands (1993) at the University of Sussex, in which the co-evolution of shop-floor schedules was explored. Such work, however, is in its infancy.

Before the full potential of co-evolutionary design techniques can be realised, the burgeoning body of work exploring artificial co-evolution must be consolidated. At Sussex, studies of predator-prey co-evolution (Miller & Cliff, 1994), sexual selection (Miller, 1994), and parental imprinting (Todd & Miller, 1993), have already been carried out and further research seems both worthwhile and inevitable. Open questions, such as the paucity of true co-evolution in natural predator-prey ecologies, in comparison to the relative abundance of such evolutionary dynamics in parasitic relationships, seem amenable to investigation through the artificial means employed within this style q6349(.)8.905]TJ -ti

world. It would probably not work.) Hence, an EHW robot controller which was evolved for a high-speed simulated environment, and then slowed down to operate in the real world, will not be making maximal use of the available hardware. This is because it is capable of producing the same behaviour in a world which is running faster, and the resources needed to a immunity to voltage and temperature fluctuations.” All of these points indicate that it is more appropriate to evolve asynchronous circuits for robot control than clocked ones. In fact, the reason that asynchronous circuits are little used is because of their difficulty of design — artificial evolution may be the answer not only in robotics, but in other areas of electronics too.

The evolution of circuits (either clocked or asynchronous) in real time allows maximal exploitation and accommodation of the natural temporal behaviour of the implementation (its physics). In addition, *asynchronous* circuits evolved in real time can put to use dynamics which would be considered as transients between clock-ticks in a clocked system. Circuits evolved in real time could be worth waiting for.

4 Conclusion

I have argued that when intrinsically evolving hardware to control a robot, the use of high-speed environment simulation to accelerate the evolutionary process imposes limitations on the nature of the circuit

[11] D. Mange. Wetware as a bridge between computer engineering and biology. In *Proceedings of the 2nd European Conference on Artificial Life (ECAL93)*, pages 658–667, Brussels, May 24-26 1993.

[12] Daniel Mange and André Stauffer. *Artificial Life and Virtual Reality*, chapter "Introduction to Embryonics: Towards new self-repairing and self-reproducing hardware based on biological-like properties", pages 61–72. John Wiley, Chichester, England, 1994.

[13] P. Marchal, C. Piguet, D. Mange, A. Stauffer, and S. Durand. Achieving von Neumann's dream: Artificial life on silicon. In

Miller, G. F., & Cliff, D. (1994). Protean behavior in dynamic games: Arguments for the co-evolution of pursuit-evasion tactics. In Cliff, D., Husbands, P., Meyer

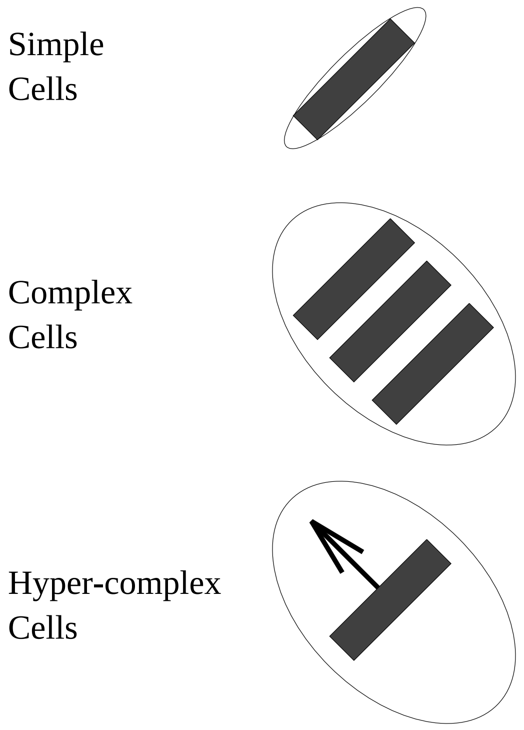


Figure 3: Popular classification of V1 neurons into simple, complex and hypercomplex cells. The grey bar indicates the optimal stimulus for the cell. The ellipse indicates the extent of the cell's receptive field. In the case of the complex cell, the bar can be in one of many positions – three possible positions are shown here. Finally, in the hypercomplex cell, the arrow indicate

As well as the neurophysiological experiments, another approach to understanding the development of the visual system is by computer simulation. Typically, these simulations involve modelling neural networks to see if they can self organise to produce neurons with similar response properties as real visual neurons. The first main simulation of this type was produced by von der Malsburg (1973), initiating many other models over the following twenty years. Here at Sussex, many simulations, modelling simple cells (Barrow, 1987), complex cells (Barrow & Bray, 1992b), colour blobs (Barrow & Bray, 1992a) and ocular dominance stripes (Goodhill, 1992) have been performed.

Given the importance of motion processing that has been mentioned earlier, it is surprising that the modelling of motion cells has been quite rare, with the exception of (Wang, Mathur, & Koch, 1990) and (Nowlan & Sejnowski, 1993). (Specific models of MT and MST development have been produced, but these typically assume certain functions for the retina, V1 and V2, eg. (Sereno & Sereno, 1991; Tanaka & Shinbata, 1994).) This is now a topic of research here at Sussex.

4 Conclusion.

An overview of the importance of motion processing has been given, along with the architecture of a mammalian visual system. The existence of a magnocellular pathway that deals with (amongst other things) motion processing seems to reinforce the importance of motion. The final section outlined two main theories for the development of the visual system: (1) genetic specification or (2) activity driven self organisation. It is likely that a combination of both of these processes are needed to produce a normal visual system, by working on different aspects of the problem. First, genetic specification can initially describe the overall topography and connectivity of retinal inputs and cortical areas. Second, the activity driven

Wiesel, T., & Hubel, D. (1963). Single cell responses in striate cortex of kittens deprived of vision in one eye.. *Journal of Neurophysiology*, *26*, 1003–1017.

Zeki, S. (1993). *A Vision of the Brasy203R10 10.B-4.10463ioi*

“society is not a mere sum of individuals Rather the system formed by their association represents a specific reality which has it’s own characteristics

This superstructure so constrains the behavioural opportunities of the individual that it is seen as essentially guiding or causing that behaviour

Within social sciences there has been a robust defence against this way of thinking From J S Mill in the 19th century to modern individualists many are unhappy with the possibility of macroscopic structures being causal of microscopic properties Two possible refutations seem to run as follows

The first is to argue that these high level structures have no goals of their own They merely inherit their seeming intentionality from the intentional humans who comprise them

But against this the holist believes that structures may be acting according to their own agendas No one wants an economic recession or a bout of inflation perhaps no one intends a moral norm to collapse Yet these things happen How is this to be squared with the idea that the intention to behave has come from the low level ? One solution is the *great man* theory That somewhere there is someone who secretly did want the change and this time has got his way Cognitive scientists may recognise this as a parallel to the idea of the grandmother sensor[1] or undischarged homunculus

A second strategy individualists use against structuralists is to ask directly how does it work ? How can this high level system cause these low level people to do things ? The answer demanded is a low level causal one a story of who did what to whom When presented with such they can then argue that this is obviously only a story about individuals and their behaviour In other words this is a reductionist argument of the kind which those cognitive scientists who wish to preserve the mind from being mere brain behaviour are continuously fighting against

These are rough parallels but nevertheless our first glance into the distorting mirror of social science has been enlightening Because the mirror reverses the levels that we normally consider intentional and non intentional it might help us understand the opposing viewpoint to our usual one

From our intuitions about folk psychology we tend to reject the dogma of the holistic social scientist But from our intuitions as cognitive researchers about both homuncular decomposition and reductive materialism we are tempted to deny the arguments which the individualist social scientists use to deny high level structures Any argument which seems to deny high level structure in favour of individual action is dangerously close to one which would collapse the psychological into the neurochemical By contrast any argument which could pump enough hot air into the mind to keep it afloat above the brain could probably launch a few leviathan like macroscopic enti[illegible] which will be familiar to some and mysterious to others the first of these is a question of *ontology* and the second *epistemology* as a Class War and should therefore do A Both allow an observer to tell the same kind of explanatory stories about why X performed A because A was the rational choice and both can make the same kind of predictions assuming X is rational he will perform A   Both are theories that there are situations or relationships which agents can find themselves in whose very nature makes a particular actor's behaviour meaningful or more predictable

Thinking in terms of high level structures can occur when borrowing explanations from biological sciences Evolutionary theory is already shot through with intuitions that are top down and often criti cised for being based on tautology Concepts such as fitness are only defined in terms of a circle of high level entities No genotype is fit merely by virtue of its internal structure Rather it is fit with relation to the phenotype and other genotypes and environment Co evolution or the evolutionary arms race is a high level structure introduced to provide explanations of current properties

Evolutionary roboticists who attempt to evolve solutions to problems find themselves thinking in terms of the evolutionary   niche   and trying to produce behaviour by designing the environment and fitness function within which a particular control system will evolve Hence there is some acknowledge ment of a world which is prior to and causally responsible fo planations of agency when we seem to be making such progress through scientific and reductionist ap

In: de Bourcier, Lemmen & Thompson eds., 1994 *The Seventh White House Papers: Graduate Research in the Cognitive & Computing Sciences at Sussex.* University of Sussex, School of Cognitive & Computing Sciences, Brighton, UK. Research Paper CSRP 350.

Creativity in Writing and Music

Rafael Perez y Perez
rafaelp@cogs.susyx.ac.uk

School of Cognitive & Computing Sciences
University of Sussex

Mario Vargas Llosa (1966) describes how experiences in Vict

n Bou n T o p on [illegible] *n o s* [illegible] *p rs*
[illegible] *s*[illegible] *r n Co n* / *Co p n* [illegible] *n* [illegible] *ss* [illegible] n v y o
Su S ool o ogn v o pu ng S n B g on p
S [illegible]

o b o ng v l u *n* u lly no n o b on ou n bu *on n* o n ng nvolv n o u on ow o ng o o v on n

n po o on n l w n on o on ou n o n un n ng by w ol ogn z T u u lly o *p rson* *on n* bu b u p on oo u ong no on n ogn on p o u n olog *or n s* *on n* n w pl ub p on l on n w o g n l on n op ng o g o on n by ub y o w ol o g n n l n lu ng o g n l on n ppl bl o w ol ogn z

3 Conclusion

ogn v n n o n n o on ou n S n ogn z w ol n bo y

Dow ll T on n o p p u l p n *osop* [illegible] [illegible] *r r* *44* [illegible]

S l [illegible] *s o r o* [illegible] *n* T b g A

In: de Bourcier, Lemmen & Thompson eds., 1994, *The Seventh White House Papers Graduate Research in the Cognitive & Computing Sciences at Sussex* University of Sussex, School of Cognitive & Computing Sciences, Brighton, UK. Research Paper CSRP 350.

Problems Caused by Ineffective Communication in Requirements Engineering

Amer Al-Rawas
ameral@cogs.susx.ac.uk

School of Cognitive & Computing Sciences
University of Sussex
Brighton
BN1 9QH

Abstract Large software projects involve many participants exchanging information through complex and recursive interactions. Effective communication is of vital importance throughout the project life cycle and particularly during the early phase of requirements specification. The various stakeholders must be able to communicate their requirements to the analysts, and the analysts need to be able to communicate the specifications they generate back to the stakeholders for validation. This paper describes some of the problems of communication between disparate communities involved in the specification activities.

1 Introduction

It is widely recognised that communication problems are a major factor in the delay and failure of software projects (e.g. see Curtis, Krasner, & Iscoe, 1988). This is especially true of "socio-technical" software systems, which must exist in a complex organisational setting. Communication is at its worst during the early phase of requirements specification where user and developers, in most cases, meet for the first time.

5 Unstated Assumptions

The problems described in this paper provide the motivation for my PhD research. My research aim is to come up with an approach that will help in facilitating better communication without the need to introduce new methods or notations. It is hoped that such an approach will overcome some of the problems described in this paper.

References

Curtis, B., Krasner, H., & Iscoe, N. (1988). A Field Study of the Software Design Process for Large Systems. *Co   un cat ons of the ACM*, 31(11).

Dasgupta, S. (1991). *Des gn heory and Co puter c ence.* New York, Cambridge University Press.

Easterbrook, S. M. (1991). Resolving Conflicts Between Domain Descriptions with Computer-Supported Negotiation. *Know edge Acqu s t on An Internat ona Journa* , 3, 255-289.

Easterbrook, S. M. (1993). Domain Modelling with Hierarchies of Alternative Viewpoints. In *Proceed ngs F rst IEEE Internat ona   y pos u   on Requ re   ents Eng neer ng*, San Diego, California, 4-6 January 1993:

Finkelstein, A. (1991). Reviewing and Correcting Specifications. In *he Fourth Annua Conference on Co puters and the   r t ng Process*, University of Sussex, Brighton, U.K.

Finkelstein, A., Kramer, J., Nuseibeh, B., Finkelstein, L., & Goedicke, M. (1992). Viewpoints: a framework for integrating multiple perspectives in system development. *Internat ona Journa of oftware Eng neer ng and Know edge Eng neer ng*, 2(1), 31-58.

Finkelstein, A. C. W., Easterbrook, S. M., Kramer, J., & Nuseibeh, B. (1993). Requirements Engineering Through Viewpoints. In *Proceed ngs of the DRA Co oqu u   on Requ re   ents Eng neer ng*, Malvern, UK: Defence Research Agency.

Gotel, O. & Finkelstein, A. (1993). *An Ana ys s of the Requ re   ents raceab   ty Prob e* . Forthcoming.

Hymes, C. M. & Olson, G. M. (1992). Unblocking Brainstorming Through the Use of a Simple Group Editor. In *C C   9   ACM 99 Conference on Co puter- upported Cooperat ve   or* . ”Sharing 1602(s)-312.362(f)

In: de Bourcier, Lemmen & Thompson eds., 1994 ***The Seventh White House Papers: Graduate Research in the Cognitive & Computing Sciences at Sussex.*** University of Sussex, School of Cognitive & Computing Sciences, Brighton, UK. Research Paper CSRP 350.

n n . . n n n n n

n

n . . . .

n . n n .

n . . . .

n

A . This paper discusses the notion of informality in HCI, leading to the design of informal interfaces. Such an interface exhibits tolerance in its input and variance in its output. Informal interface representations are internally composed of informal objects that are a combination of a prototype, such as a straight line, and associated informal dimensions such as shakiness and thickness. In an informal interface it is the gist of human-computer interaction, instead of a higher level of formalism, which is paramount. Internal representations of informal objects can be decomposed, manipulated, and recomposed. An example is given of a software tool that has been developed 11137(c)5.6453137(u)-e50914(m)10.876(p)-.64(d)-267.117(a)-6.93404(
(r)4.23177(a)5.6431857s)-5.52048(e)-4.952271(m)10.87177e04(a3(1(d)-223.281(h)-4.10914(e)-257.364((c)5.67511(i)-6.931(74311(t 1

ophy. Some work relates indirectly to informal interfaces, in that researchers have been experimenting with different concepts behind the human-computer interaction by using more intuitive graphical interfaces.

ˆn n n . n n.

In art and design there has been some work in analysing the principles behind sketching [5]. The authors explain how Leonardo da Vinci advocated the use of “untidy indeterminacies” for working out compositions, because he believed that sketches stimulated visual invention. Research from cognitive psychology [1] suggests that this is the case, in the way that a mental-imagery model is used by the human brain. It is suggested [16] that the brain can create a mental image of a sketch, and then apply processes to alter or enhance that image to useful and creative effect. Negroponte [14] notes that “Sketch recognition is as much a metaphor as fact. It is illustrative of an interest in those areas of design marked by vagary, inconsistency and ambiguity. While these characteristics are the anathema of algorithms, they are the essence of design.” Lohse [12] indicates how research into cognitive models for the perception and understanding of graphs can be applied to informalism; rough-sketch representations of graphs are inherently interesting as informal objects. One of the key concepts from informal interfaces is the relaxation of the invariance of output by computers, so it is revealing to study how people perceive and process meaning from graphical information. Lohse describes a computer program UCIE (Understanding Cognitive Information Engineering), which models the underlying perceptual and cognitive processes used by people to decode information from a graph, and considers results from

4 ˆ n ˆ n.

The input mechanism in the case of a stylus is as follows: a flow of pen-ink is input from the stylus position and both displayed on the screen and also stored in an int n

1. Boden, M. The Creative Mind; Myths and Mechanisms. Basic B

In: de Bourcier, Lemmen & Thompson eds., 1994 *[illegible]* University of Sussex, School of Cognitive & Computing Sciences, Brighton, UK. Research Paper CSRP 350.

A Proposal for the Detection of Software Interactions

Joseph A. Wood*
joew@cogs.susx.ac.uk

School of Cognitive & Computing Sciences
University of Sussex
Brighton
BN1 9QH

Abstract This is a position paper, which presents my views on software interactions, why they cause complications and what can be done to overcome these obstacles. A notion of interaction is defined, and the concept of "misfocusing" between groups is introduced. Finally, a brief outline of how an automatic tool might tackle this problem is described.

1 Introduction

I have lost count of the projects I have seen where various modules work in isolation, but the integrated modules do not work as expected. All such systems were designed and reviewed; so what went wrong? This paper examines what I believe are some of the underlying reasons for this situation.

Before progressing further, I should explain that I am interested in what is called Programming-in-the-Large, (de Remer and Kron, 1976). I assume that all experienced software engineers can produce small programs correctly.

1.1 Paper's Structure

Section 1.2 presents an example of a software interaction from a large industrial project. Section 2 briefly explains my usage of the term software interaction. Section 3 looks at the kinds of interactions that cause problems in large software projects. Section 4 considers how these problems might be handled, and finally section 5 outlines my future research plans.

1.2 Software Interactions: an example

Before attempting to explain interactions, let me give a real example from one project. As part of a large multi-processor Ada[1] project, one group supplied the message handling (MH) capability for use by other ("user") groups. Being written in Ada the interface to MH had been made available and the meaning of each data-type had been defined. Message handling worked in isolation; the user modules compiled with

2 What is a Software Interaction?

I have loosely used the term interaction above, without giving it a meaning. This section examines the nature of interactions in more detail.

shared maths libraries do not cause problems; all engineers have a shared (common) understanding of `sin` (say).

How does such a situation arise? All too easily, both parties think that the concept is obvious, and the user simply wants to use the provided service. I regard this as a type of misfocusing. By focusing I mean that an engineer's attention is focused on a particular activity, and other (non-central) issues are only peripheral. Hence as long as the periphery looks OK, no further notice is taken.

Many kinds of material interaction are now checked for in the later stages of a project, for example, type checking. However, the early stages of project development are less well supported, and in particular there are no checks for abstract interactions. How does this impact on software architecture?

Firstly, in the early stages of designing a system, i.e. when the architecture is being developed, attention focuses on the kinds and uses of services, not on the specifics of an interface. That is, designers are more interested in the broad nature of components rather than the exact details of how a service is provided. For example, message handling shall provide facilities for creating and sending messages to other parts of the system. Observed defects from this focusing are "we cannot provide this service because the information is not available". Hence, interfaces are broadened or shared data areas become a little more

In: de Bourcier, Lemmen & Thompson eds., 1994 *v nt Hous p rs Gr u t s r nt Co n Co put n s t uss* University of Sussex, School of Cognitive & Computing Sciences, Brighton, UK. Research Paper CSRP 350.

o a an n _ n bu _n o M P o a _n an ua

an_ a a
an_ o u a u

oo o o _ o pu_n _n
_ o u
_ on

Ab a We propose an intelligent debugging system with a knowledge representation that is independent from the programming language of a particular program to be debugged. We employ a knowledge representation technique called the Plan Calculus which has been developed by Rich [Rich, 1981b]. In order to debug a program we translate a given program to a surface plan representation, parse the surface plan to understand the overall function of the program, and use near-miss information to known plans to locate bugs and repair them.

Since our system will locate bugs by manipulating programming knowledge which is independent from any programming language, it can be used to debug programs written in any procedural language provided that the front end to the system is maintained. Our system is aimed at debugging student's ML programs and can be incorporated into an ITS (Intelligent Tutoring System) system as an Expert module or can be used as it stands.

n o u _on

The aim of our system is not the general task of debugging which notoriously is beyond the state of the art, but the simple task of finding bugs and repairing them in student programs for known exercises. Our research proposal is implementing an intelligent debugging system for ML based on the plan calculus formalism. The plan calculus is a knowledge representation formalism for representing programs and programming knowledge such as algorithms and data structur pre-stored in the plan and reference libraries, the Bug Detection module attempts to locate any logical errors in the student program. Subsequently it reports them (if any) either to the student so that he/she

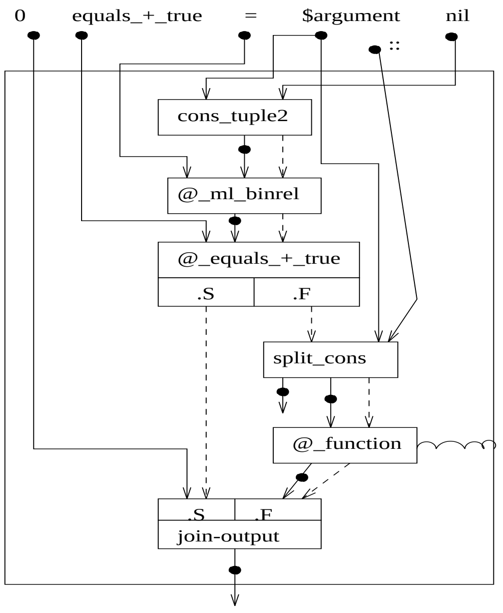


Figure 1: surface plan of the student buggy program

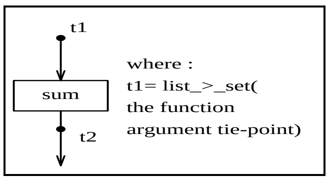
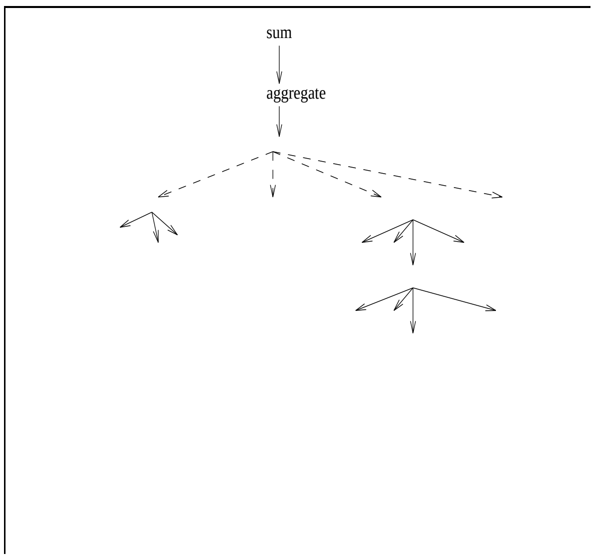
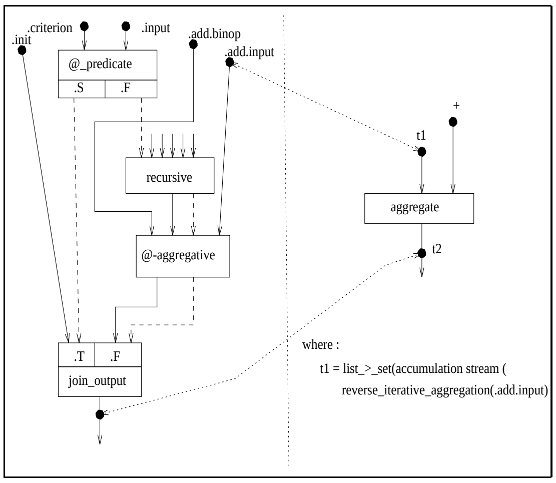


Figure 2: sum goal

to in section 2 and realizes that these are not the cases to proceed with. Then it moves to case 5 and figures out that this is the one. That is, the Bug Detection module could not find a complete or partial plan either in the complete chart or in the partial chart. As described earlier, it fetches the corresponding rules for this goal from the plan library and selects those rules which are compatible with the current goal.[7] Then it propagates any instantiated tie-points from the current goal to those rules and takes a rule. Now the right hand side of this rule becomes the current goal to proceed with. When a rule succeeds it discards the remaining rules for the current target, otherwise it tries the next rule in this category. For this example there is only one rule which is the *r* *t* rule (see figure 4) and the module sets it as the current goal and attempts to debug the gi.4398 Td 6(e)5.64311ng e e e e8I1e3-(F)8.0533(o)-4.1Ta1 3369.55 -7160.318-4.10914(r)4


sum
aggregate



where :

t1 = list_>_set(accumulation stream (
reverse_iterative_aggregation(.add.input)

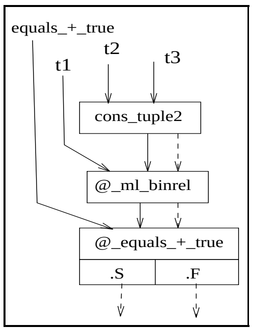
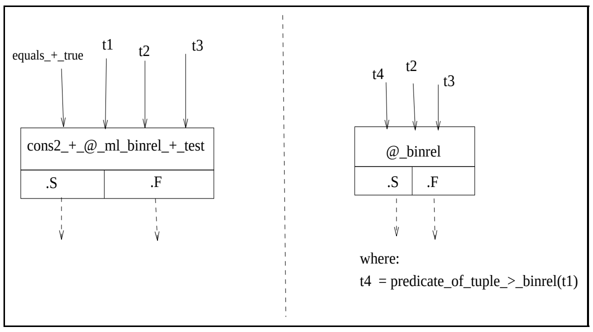


Figure 9: cons2_+_@_ml_binrel_+_test plan



equals_+_true
t1
t2
t3
cons2_+_@_ml_binrel_+_test
.S
.F
t4
t2
t3
@_binrel
.S
.F
where:
t4 = predicate_of_tuple_>_binrel(t1)

Figure 10: cons2_+_@_ml_binrel_+_test_as_@_binrel overlay

form of a hash table which contains the plan type and its score. In this case it chooses the _ *r t_y* sub-plan to proceed with. Now the current goal for the module is the _ *r t_y* plan. Debugging this falls under case 5 (because there is no complete or partial plan in the chart) which in turn directs the debugging the program for _ *_nfun t_on*. Figure 6 shows the corresponding overlay. There are four rules corresponding to this plan in the plan library and the Bug Detection module chooses a rule from the extracted set which is related to the current target. If this choice led to a discrepancy with what has been asserted, then the module backtracks the process to select the next rule according to its heuristic again. This process continues until it succeeds or it abandons the whole process because of contradictions.

At this stage, the module selects *ons _ _ _fun* plan (see figure 8) as the current goal and debugs the program for it. This falls into case 5 again, but this time all of its sub-plans are primitive plans. Therefore, it asserts that the student program has failed to produce such plans and it creates a corresponding plan for each of the sub-plans involved and passes them to the plan recognition module. This invocation entails the addition of the sub-plans to the complete chart as well as the instantiation of other pertinent plans such as *ons _+_ _fun*, _ *_nfun t_on*, and _ the complete chart.

Since the whole debugging process is recursive, at this stage of the analysis, the debugging process completes and recursion unwinds. Since the highest-level goal (i.e., *su*) plan is found in the complete chart, the whole debugging process is terminated. This means that the Bug Detection module found the highest-level goal but only by repairing the student program. In the final stage, the Bug Detection module reports on what has been done, for example: stating that _ *nfun t on* plan was missing and has been created and added to the chart, implying that the student had missed a binary operation (i.e., +) to sum up the 'head' of the list in the way back of the recursion.

This example shows how the Bug Detection module locates and repairs the bug. For the interest of the reader we have included the graphical representation of the plans and overlays referred in figure 3. We omitted their logical definition for the sake of clarity. Interested readers concerned with the logical foundation of the plans and overlays in the plan calculus are referred to [Rich, 1981b] and [Rich, 1981a].

4 on u_on

In this short paper we delineated the overall strategies use

In: de Bourcier, Lemmen & Thompson eds., 1994 *The Seventh White House Papers: Graduate Research in the Cognitive & Computing Sciences at Sussex.* University of Sussex, School of Cognitive & Computing Sciences, Brighton, UK. Research Paper CSRP 350.

Goa For u ation as an AI Research Issue

Re edios de Dios Bu os
re edios cogs susx ac uk

Schoo of Cognitive Co puting Sciences
niversity of Sussex
Brighton
BN1 QH

Abstract Although research in AI Planning systems has made considerable progress from its humble beginnings, much is still to be desired. A lot of research work remains to be explored to address several unresolved issues and problems. An important research issue in AI Planning systems that needs to be addressed, investigated, and solved concerns the problem of endowing an artificial resource-bounded rational agent with the ability to formulate its own goals, as it navigates a world that is characterizable as complex, dynamic and uncertain. This paper cites and discusses the main reasons and issues as to why goal formulation is an important research topic to be tackled in Artificial Intelligence. First, a definition of goals is given. Next, several reasons are identified as to why goal formulation is a research issue in AI. A summary and analysis of related work on goal formulation is then presented. Lastly, a list of research questions that need to be resolved is enumerated.

1 Introduction

Research work in AI Planning systems has progressively evolved from the simple operation of identifying and generating possible action sequences to achieve a set of specified goals (classical AI Planning) to the more sophisticated process of integrating the different functions of planning, execution, monitoring/control and learning. Moreover, the characteristics and environment of the domain of experimentation and application have gradually metamorphosed from simple, static and predictable to complex, dynamic and uncertain.

However, although research in AI Planning systems has made considerable progress from its humble beginnings, much is still to be desired. An important research issue in AI Planning systems that needs to

importance, i.e. which goal should be accomplished first.

However, due to the dynamism and uncertainty of the environment, decisions that were made previously (in terms of plans and goals) may be affected and become inapplicable later. During such trying and unexpected situations, the agent must be able to decide quickly, react and respond appropriately, be able to cope up with the new constraints and conditions, regain full control of the situation, and resume its normal functions.

Wh t

be detected given a set of premises are applied on all information. [Lizotte & Moulin, 1990]

nreso ved Issues on Goa For u ation

Analysis of the above cited works indicates that by and large, research on goal formulation has concentrated on the goal detection aspects. Goal detection is the process of signalling or reminding the system (planner) that it has a goal(s). The goal is usually detected when a situation or condition that gives rise to the goal is sensed by the planner. Although such a method is valid and effective, it is most probably not the only means to detect the occurrences of goals. Also, further elaboration is needed on the types of situations that emerge and how such situations give rise to goals.

Another important issue that needs further study is the reasoning and decision-making process that is undertaken when evaluating whether a newly detected goal sh goals. It must reason and decide what goals to achieve and when to achieve them. It must be able to detect its own goals, assess their feasibility, prioritize them, evaluate their validity (continuation, termination, suspension, modification) and modify them in the light of present circumstances. An intelligent agent's success in pursuing its goal-directed activities will largely depend on the behavior it exhibits during the formulation of goals. These above-cited reasons justify the research issue that an agent should be endowed with the capability of formulating its own goals.

References

Allen, J. F. (1979), *A Plan-Based Approach to Speech Act Recognition*, Doctoral dissertation, Department of Computer Science, University of Toronto. Also available as Technical Report No. 131, University of Toronto.

Chin, D. N. (1988), *Intelligent Agents as a Basis for Natural Language*, UCB:CSD-88- 396, Division of Computer Science, (EECS), University of California, Berkeley, CA.

Faletti, J. (1982), PANDORA – A program for Doing Common Sense Planning in Complex Situations, in *Proceedings of the Second Annual National Conference on Artificial Intelligence*, pp. 185-188, August, PA.

Ford, M.E. (1992), *Motivating Humans Goals, Emotions, and Personal Agency Beliefs*, USA: Sage Publications, Inc.

Wilensky, R. (1983), *Planning and Understanding A Computational Approach to Human Reasoning*,

In: de Bourcier, Lemmen & Thompson eds., 1994 ***The Seventh White House Papers: Graduate Research in the Cognitive & Computing Sciences at Sussex.*** University of Sussex, School of Cognitive & Computing Sciences, Brighton, UK. Research Paper CSRP 350.

Structural Extensions for the ELKA Model

Ricardo Garza M. [*]
ricardom@cogs.susx.ac.uk

School of Cognitive & Computing Sciences
University of Sussex
Brighton
BN1 9QH

Abstract In this paper the forked link is used in the construction of ex

[9] J. Iivari, Relationships, Aggregations and Complex Objects, Information Modelling and Knowledge Bases III, S. Ohsuga et al eds, IOS Press, 1992.

In d Bourc r L n o pson ds

n v rs ty o
uss x c oo o Co n t v Co put n c nc s Br ton K s arc ap r
C

Feature Extraction Using Wavelets for the Classification of Human *in Vivo* NMR Spectra

Rosemary Tate

Data

Processing the Spectra

Eac sp ctru was r pr s nt d by a data v ctor o n t _ ' r proc ss n was carr d out to a t sp ctra co pat b ' p a s- w c ad s t y d anot r rando sa p was s ct d' n t v tar ans w r nc ud d- t r su ts w r not so ood and o t two tra n n s ts- and o bot t st s ts- w c ac nc ud d two v tar ans- w r c ass d corr ct y' s poor r r su t was probab y du to two actors rst y t nu b r o v tar ans ay b too s a or t to b nc ud d n suc an ana ys s- and s cond y b caus t d r nc s b tw n t d ts o v tar ans and t ot r two roups s not n ar y as r at' s was s own by t act t at a ost a o t sc ass cat ons w r o n vor s b n c ass d as v tar ans or v c v rsa' s r su ts w r v ry s ar to t os obta n d w n p a var ab s w r us d'

Conclusions

s study d onstrat s an app cat on o wav ts or atur s ct on n t c ass cat on o data' t s part cu ar sp ctra cou d b c ass d qua y succ ss u y by ot r ans- us o t wav t trans or r ov d t n d or s ct n and quant y n p a s- t us r duc n t a ount o pr proc ss n n d d' s cou d b v ry us u w n auto at d proc ss n s r qu r d' oca s d prop rty o t wav t trans or - apart ro ts advanta s n n od n t p a s- a so a ows us to d nt y t ost portant contr butory atur s to c ass cat on' s sp ctra w r unusua or u an sp ctra n t at t quant cat on o t p a s was r at v y asy' s s not n ra y t cas ' r su ts ro t s(' _ (' _ r(' a(' onta s p t q ' a(' r(' t('

In: de Bourcier, Lemmen & Thompson eds., 1994 *The Seventh White House Papers: Graduate Research in the Cognitive & Computing Sciences at Sussex.* University of Sussex, School of Cognitive & Computing Sciences, Brighton, UK. Research Paper CSRP 350.

MML, a Modelling Language with Dynamic Selection of Methods

Vicente Guerrero-Rojo*
vicenter@rsuna.cogs.susx.ac.uk

School of Cognitive & Computing Sciences
University of Sussex
Brighton
BN1 9QH

Abstract Second generation knowledge based systems often incorporate multiple problem solving methods. Up to day, there is a need for modelling languages capable of handling, invoking, evaluating and choosing multiple methods at run-time. There are several modelling languages with such capabilities. With them it is possible to develop robust, more flexible and less brittle systems. Unfortunately, those languages are not flexible enough to cope with the behaviour of the systems when more methods are incorporated. In this paper we propose a new modelling language which overcomes these shortcomings.

1 Introduction

Second generation knowledge based systems (systems) often incorporate multiple problem solving methods. The advantages of having multiple methods in a system have become apparent: robustness [Simmons 93], flexibility [Vanwekenhuysen, Rademakers 90], broader kind of reasoning [Delouis 93], less brittleness, reusability [Punch, Chandrasekaran 93].

The decision about which method to use is very much an open problem [David et al. 93]. Nowadays there is a need for modelling languages capable of handling, invoking, evaluating and choosing multiple methods at run-time [Chandrasekaran, Johnson 93].

There are several modelling languages with such capabilities. With them it is possible to develop robust, more flexible and less brittle systems. These capabi

2 MML - Multiple Method Language

The MML is a task-independent modelling language for the explicit representation of systems with flexible control strategies which allow dynamic selection of methods. MML is being designed as an initiative to overcome some of the shortcomings that current modelling

```
[define                propose
properties:
   [type               task]
   [goal 'The goal  of this  task is the  allocation (solution)  of ...]
   [input              components resources]
   [output             allocations]
   [control-terms      NameOfMethod]
abstract structures:
   [associated-methods  decomposition random-init sequential]
   [satisfaction-crit   allocations  not = empty ]
   [preferences         [random-init all] [sequential decomposition]]
   [code
             collect-methods(associated-methods) -> SetOfMethods;
             while SetOfMethods /= [] do
                 select-a-method(SetOfMethods, appropriateness-crit)
                                   ->   NameOfMethod;
                 applicable(NameOfMethod) -> Applicable;
                 if Applicable = true then
                     apply-method(NameOfMethod);
                     test-satisfaction(satisfaction-crit) -> TaskSuccess;
                     if TaskSuccess = true then
                         return;
```

```
[define             decomposition
properties:
   [type            method]
   [goal            'To allocate components into resources using ...']
   [input           components resources]
   [output          allocations]
   [m-type          non-terminal]
   [ck-type         procedural]
   [backtracks      false]
   [structure       assemble-plan assign-resources]
   [control-terms   NameOfMethod]
abstract structures:
   [applicable-crit
                    components-value       exist and
                    resources-value        exist ]
   [appropriateness-crit
            time =       bound    and
            a-plan       exist    and
            problem-type in [sisyphus researchers]
   [code            ...
                    debug(m, 'Executing decomposition');
                    getdomain( components, value) -> Components;
                    getdomain( resources, value) -> Resources;
                    for Res in Resources do
                        putdomain(store, allocations, Res, []);
                    endfor;
                    ...]
activities:
   [applicable( applicable-crit) -> boolean]
   [apply-method( code)   using code-interpreter]
control strategies:
]
```

Figure 2: Decomposition method in the Sisyphus problem

```
[define                     single-method
properties:
   [goal                    'call an object-level method']
   [type                    meta-method]
   [ck-type                 procedural]
   [m-type                  meta-method]
   [structure  take-method applicable
               apply-method test-satisfaction]
   [control-terms       NameOfMethod]
abstract structures:
   [code
    vars NameOfMethod, Applicable, TaskSuccess;

     take-method(associated-methods) -> NameOfMethod;
     applicable(NameOfMethod) -> Applicable;
     if Applicable then
         apply-method(NameOfMethod);
         test-satisfaction(satisfaction-crit) -> TaskSuccess;
         if TaskSuccess = true then
            return;
         endif;
     endif;
    ]
activities:
   [apply-m-method( code) using code-interpreter]
control strategies:
].
```

Figure 3: Single-method meta-method in the Sisyphus problem

in a modelling language (e.55526(e)-e.559 Td [(apply-metho)999.81(t)-6.93181(h)-4.10914(o)-4.11137reTJ 5931.–ilod
ñ14(s)-22422(.55526(e)-e)5.64422TJ -356.63137(f)-661(1-e)5.64422(re -356.63)-4.11277203 /R1

- The language distinguishes between control knowledge for decomposition of methods and control knowledge for method-related activities.

MML has been designed to capture some of the more desirable features required for solving problems with multiple methods. Initial experiments with the Sisyphus problem indicates that MML is not only a more flexible system but also it has led to an improvement in the specification of the knowledge about methods.

Further work will involve extending the range of methods defined in MML and evaluating its mod-

[Vanwekenhuysen, Rademakers 90] Vanwekenhuysen J., Rademakers P. Mapping a Knowledge Level Analysis onto a Computational Framework, in: Aiello Loigia C. (Editor), Proceedings of the 9th European Conference on AI, 1990.

cases. Hobbs acknowledges this, but the percentage of correct antecedents found (83%) is still encourphonetically expressed su 76.5602 72C-4.10914(pj-6.9307(l)5.64422(d)5.644311t)-6.93181(i.-2097.945(N-15.2868(uc

segments, with the same local topic; and **subsegments**, which contain subtopics related to the segment topics.

The annotation is made up of eight **slots**. The three first slots refer to the segment or subsegment. The first one specifies whether the unit is a segment or subsegment, together with a number that fits the unit into a sequence within a fragment - in the case of segments - or within a segment - in the case of subsegments. The second slot specifies a discourse function from a set which is kept as small as possible. It is not possible to discuss every option for each slot in this paper for reasons of space. The third slot specifies a topic for the segment or subsegment.

In: de Bourcier, Lemmen & Thompson eds., 1994 ***The Seventh White House Papers: Graduate Research in the Cognitive & Computing Sciences at Sussex.*** University of Sussex, School of Cognitive & Computing Sciences, Brighton, UK. Research Paper CSRP 350.

Doing a PhD with Hindsight

Julian M. L. Budd and Eevi E. Beck

Make friends

Lastly, don't become isolated from other research students. It's possible they have already encountered your problem and know how to solve it. Moreover they can give you support when things are not going well and you need cheering up.

Eevi [2]:

Go to conferences

If you can, go to conferences. Discuss your work and your ideas with people there. If you don't know anyone before you go, it'll be a good opportunity to meet people in your field. Chances are that you'll find that some people are really interested in what you're trying to do, which can be a great boost to your confidence! Also, seeing that there's nothing unusual in lots of people not being interested can be good: it's not you or your work there's anything wrong with, it's just the reaction that everyone gets. Some are going to be interested, many not.

Be critical of criticism

If someone – your supervisor, another student, a prestiguous name, or anyone else – criticises your work, try not to take it personally. Evaluate it calmly, if you can: does it make sense to you? Has she (or he) misunderstood what you're trying to do? Are there aspects you need to clarify, like what limitations you have set on your work? If you think they're misguided, might there still be something to their criticism; an angle you hadn't thought about, or a pitfall you're in danger of falling into?

If it makes sense to you, follow advice. If it doesn't, try to work out why it doesn't, and defend it (in later conversations, in your thesis, or wherever). In an existence where feedback on your work is a scarce resource, try to learn something from it either way. At the same time, if you feel radical, ***be*** radical. It's ***your*** thesis. In fact, many established researchers expect PhD students to be the ones to rock the boat of established truths in the discipline. If you are convinced you have grounds to do that (or will have grounds), then go for it. Follow your convictions!

Enjoy life!

To stay healthy, physically and emotionally, you have to make sure you don't just think thesis (yours and others') all the time. Do your favourite sport or start another one; spend time with friends; set up your life so you're regularly and frequently reminded that there are other things which are more important than your thesis. If you have children, you're probably getting reminded of this already, but many a single preson has become a social hermit, and many a relationship has felt the strain after a while of one or both doing a PhD. Try to make your weekly routine include things you really enjoy which have nothing to do with academic work.

Keep your distance

The interest of people who're not doing a PhD and never have done one is not going to last if you're constantly talking thesis. That's ***why*** I think it's really important to spend time with others – you're forced to talk and therefore think about other things.

Keeping a distance from your thesis makes it less likely that you'll turn into a social hermit. What it also can do is help you work better when you are working: occas the work I did at times of prolonged immersion was... well, I might have interesting ideas, but it was often ***not*** what I needed to get on with to meet the deadlines I'd set. Coming back to it after a day out walking (or whatever), I'd often only then realise how I was taking myself off on a tangent which I could ill afford.

Don't kill yourself

This one is no joke. People ***have***