The 11th White House Papers
Graduate Research in Cognitive and Computing Sciences at Sussex

Editor:
Fabrice P. Retkowsky

CSRP

October 1

ISSN 1 0- 1 2

UNIVERSITY OF

Cognitive Science Research Papers

THE ELEVENTH WHITE HOUSE PAPERS

CSRP 495

Graduate Research
in Cognitive and Computing Sciences
at Sussex

October 1998

Preface

At the sle of horns a small village situated not far from aywards eath stand a few white buildings hese buildings which sometimes act as Sussex University's conference centre are most of the time used as a playground by rabbits owever every year they are disturbed by a congregation of COGS research students n accordance with this time honoured tradition 1 saw the 11th sle of horns wor shop where COGS students gathered to present their wor share some ideas and spend a lot of time socialising

he hite ouse Papers are a conclusion to this year's wor shop You will find here some articles as well as some shorter papers written by PhD students in the last few months he aim is to show which domains we are interested in and to give a rough idea to new students of what's to come

e would li e to than Matthew ennessy and the COGS Graduate Research Centre for funding the o wor shop as well as all the PhD students who contributed to these hite ouse Papers but particularly John alloran for being the Post Graduate Representative in Cogs and organizing the sle Of horns wor shop

Fabrice Ret ows y

o the who Don't Do what they're old

Contents

n Ret ows y F P Ed 1
he 11th hite ouse Papers Cognitive Science Research Paper
School of Cognitive Co puting Sciences University of Susse

Odd bites into bananas don't make you blind
Learning about simplicity and attribute addition

Hilan Bensusan
hilanb@cogs.susx.ac.uk

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

he commonsense message from Emma's story above is that successful induction requires experience in an environment. Emma hasn't performed enough inductions to realise that in our environment odd

A problem descriptor generates values for the descriptor vector that is partly based on the working representation of the problem and partly on the way the baseline learner generates the working representation

A bias pool manager classifies the training problems and applies the learners in the bias pool to the test problems according to the meta learning classifier. The training problems are classified in terms of the most accurate bias or, in case of more than one bias with the same accuracy, in terms of si[illegible] he fo˛˛owing two trees have shape of and 1 respective˛y

```
        d
    b                 b
        e
a                 a
        f
    c                 c
        g
```

Homogeneity: he nu ber of ˛eaves divided by tree shape

Balance of the tree: Given a˛˛ the possib˛e va˛ues for ( ) ca˛cu˛ate ( ) as

$$(\quad) = \quad *$$

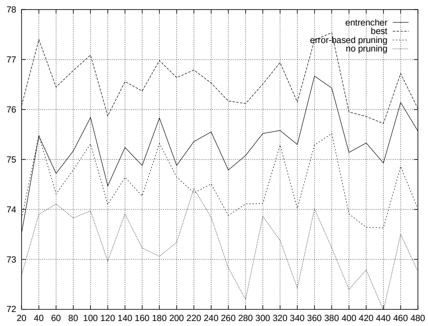
where is the nu ber of ti es occurs in the set of a˛˛ the ˛eaves of the tree he ba˛ance is then easured by the fo˛˛owing su for a˛˛ the possib˛e va˛ues for ( )

$$\sum (\quad) \log (( ) ( ) ( ) ( ))$$

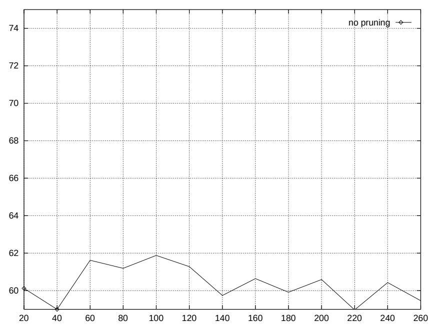
problems reported in (Thrun et al. 1991). Given the original 6 attributes of the MONK's problems there are [illegible] possible classifications. For the current experiments a number of classifications were chosen and [illegible] problems (composed by training and a test sets) were constructed for each classification. The ML learning system was trained on an increasing number of problems and tested on different test sets of problems.

2 Simplicity biases

The Occam razor is a popular and widespread bias: whenever possible prefer simplest hypotheses. In many contexts both scientists and laymen would appeal to simplicity to decide between different alternatives. One can ground the use of the razor on some sort



78
77
76
75
74
73
72
20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340 360 380 400 420 440 460 480
entrencher
best
error-based pruning
no pruning



74
72
70
68
66
64
62
60
20 40 60 80 100 120 140 160 180 200 220 240 260
no pruning

4 Representation bias

he input representation of a problem essentially influences the overall learning performance Craven Shavlik 1 Since learning biases deal in representations a good amount of effort has been put in finding appropriate strategies to improve the input description of a problem such that learning becomes easier for a given bias his effort is often conceived as part of the learning process since a better input representation is expected to emerge from the interaction between the problem and the learner Matheus 1 Rendell Cho 1 ne Michalski 1 Learners that have the ability to search for different input descriptions are often called constructive learners Constructive induction is now a common strategy to improve learning performance as one can easily see that the ability to redescribe the problem often increase the C dimension of a learning system

n this paper shall consider different strategies for attribute addition a form of constructive induction he fundamental element of a system that re represents by adding new attributes is a set of [illegible] will then have to choose between the constructors and the simple baseline learner with no attribute added. If no new attribute is taken to be needed, the original representation of the problem is considered to be such that none of the existing constructors could improve it.

Now, shall refer to a problem as falling into the domain of expertise of a given constructor whenever the new attributes built by the constructor promote a learning test accuracy of at least . Of course, a problem can fall into the domain of expertise of more than one constructor.

5 Learning to choose constructors

For these experiments, MONK'S SPACE classifications designed to generate problems that would fall into the domain of expertise of each constructor were generated. For each classification, problems with 1 training instances were generated. Performance of the constructors were measured in a test set composed of the whole classified instance space (instances). The meta-learning system was trained on an increasing number of problems and then tested on non-overlapping test sets of problems. The performances was again compared to the best possible bias in the constructor pool. Here, as before, the axis is the accuracy in the test sets and the axis represents the training set sizes. Once again, each point represents the average of runs under a fixed training set size.

Now, since there are overlaps between domains of expertise, the difference of performance between the best and the second best constructor may be so small such that no clear signal exists for [illegible].

single instance might be enough. Meta learning can therefore be seen as a force driving machine learning from [illegible]'s naive conjecturing to the informed induction of Wi[illegible]'s chemist.

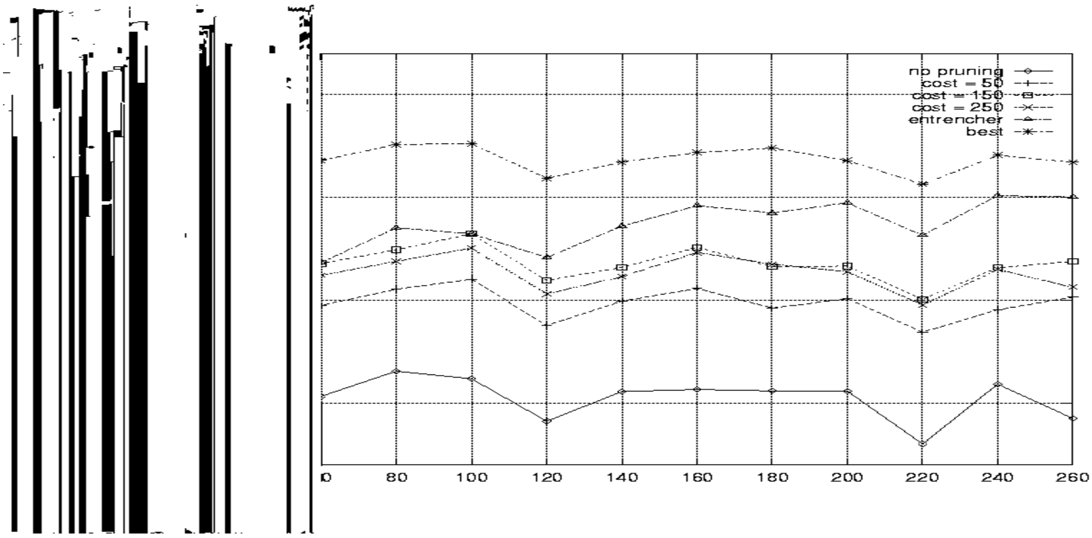
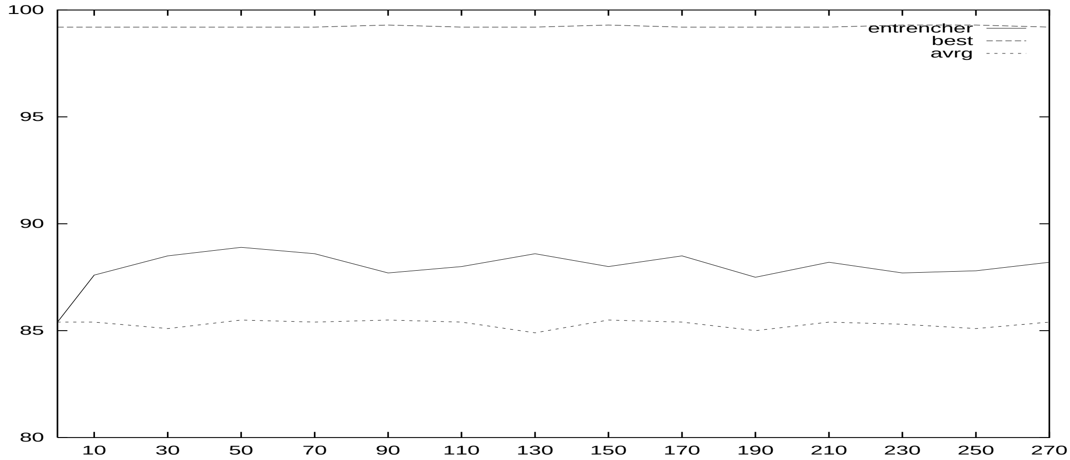


Figure Learning how [illegible]ch cost co[illegible]e ity pruning is required



100
95
90
85
80
10
30
50
70
90
110
130
150
170
190
210
230
250
270
entrencher
best
avrg

References

Bensusan 1 God doesn't always shave with Occam Razor – learning when and how to prune n ede ec C ouveiro C Eds
pp 11 –1 Berin Springer

Bensusan i ia 1 Learning to earn booean tas s by decision tree descriptors n So en id G Eds
pp 1–11 rague Czech epubic

Bu A Ehrenfeucht A ausser D ar th 1 Learnabiity and the apni Chervonen is di ension
–

Bu A Ehrenfeucht A ausser D ar th 1 Occa razor
02 –

Chan Stofo 1 E peri nts on tistrategy earning by a earning n urphy   azzani   1   1   d of   Constructive induction of   f n concepts for discri   ators in decision trees   n   San ateo CA QSA organ auf   n

urphy   azzani   1   E p oring the decision forest An e   rica i   estigation of occa   razor in decision tree induction

aga o G   auss er D   1   Boo ean feature discovery in e   rica   earning 1–

Quin an J   1   nduction of decision trees   1   1–

Quin an J   1   2   organ auf   n San ateo CA QSA

Quin an J   ivest   L   1   nferring decision trees using the ini   Description Length rincip e   An ear y version appeared as   LCS   echnica report   LCS   Septe   er 1

2.2 Design

Both e peri ments consisted of seven conditions hese invo ved the recognition of faces which were

- 1 Un manipu ated
- B urred Gaussian fi ter r = pi e s
- Scra m ed re arrange ment of horizonta face strips
- nverted
- Scra m ed and nverted
- 6 B urred and Scra m ed
- B urred and nverted

A between sub ects design was e p oyed and sub ects were rando y assigned to each condition Sub ects were assigned to the sa m condition for both e peri ments and the order in which sub ects co p eted the two tas s was counterba anced

2.3 Materials

For e peri ment 1 photographs of ce ebrities fa mous in the U served as targets Distractor faces were individua y matched to target faces on the basis of age hair co our and ength and qua ity of i mage

For e peri ment two photographs were ta en of 44 students at University Co ege London a f the faces were rando y designated targets and one un manipu ated photo was used in a study phase whi st the other manipu ated was shown during the test phase One photo of each of the re maining faces was matched to each target face and these served as distractors in the test phase

2.4 Procedure

Faces in e peri ment 1 were presented to sub ects one at a ti m in a rando m order Sub ects had to decide whether each face was a ce ebrity or a non entity he accuracy of their choice as we as their reaction ti m was recorded n e peri ment sub ects fi rst received a study phase in which they viewed un manipu ated versions of the target faces one at a ti m for seconds he test phase of the e peri ment para e ed that used in e peri ment 1

3 Results

3.1 Stati44E7F8(e4(.3)5.64311(e)5.64311(i)14.9784(v)18E8J2f)4.23177(a)27(u)-42014(t)-6.930

2.rke.231777.4mloexrgr

3e1p h s5.643314(n)-4.10014(s)-246686(o)-4.11137(l)-6.92958(l)-6.93441(u)--4.10091(n)-4.

n Ret ows y F P Ed 1
he 11th hite ouse Papers Cognitive Science Research Paper
School of Cognitive Co puting Sciences University of Susse

From an Encyclopedia to a Teaching Space: Using the Web in Schools

Ann Light

controlling scope personal and group hyperte ts class networ s school intranet and ultimately a public site   nevitably it would depend on the prevailing ethos of school and the image that senior staff wanted to pro ect

So far there has been little discussion of these issues as there has been little mention of the benefits of a hands on approach   hen the ES did publish an article on how schools might use websites it was entirely about promotion into the community Flanagan 1

o conclude there are many uses for the latest media in classroom and so far this message has not been as widely disseminated as the media themselves One of the most e citing aspects the chance for students to participate in the creation of content has had the least attention of all   his paper presents an argument for why this needs addressing   he creation of Media Studies as a sub ect area was a slow bottom up response to the changing nature of society Something more is needed this time

n et ows y Ed 1
he 11th hite ouse apers Cognitive Science esearch aper
Schoo of Cognitive Co uting Sciences niversity of Susse

3D Interactive Learning Environment

Nuno Otero
nunop@cogs.susx.ac.uk

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

1 Introduction

his te t presents the proposa for a research progra to be deve oped on the app ication of virtua environ nts to earning t wi estab ish the in goa s of the study integrate the n e isting wor and propose the ans to achieve such ob ectives

n the second section we wi present so of the st in uentia proposa s on the c assi cation of graphica si ations a ong with so discussion upon a typo ogy for co uter graphics syste and its usefu ness his topic is i ortant since it wi enab e us to c arify the potentia interactivity properties that these ind of syste have

hen we wi re ect about the e isting conceptua fra wor s to dea with the prob e f earning p1

te pouen11 1 tch d i a 1 1 o 111

stereoscopic head trac ed disp ays hand body trac ing and binaura sound is an i ersive ti sensory e perience pp

- Autonomy refers to the reacting and acting capacity that the computational model have to events and stimulus. This means that we could distinguish, for example, between a system that limits itself to be read and another that alters his narration in accordance to user actions.

- Interaction refers to "the degree of access to model parameters at runtime" (pp 1). Here Zeltzer (1) points out that the degrees of freedom exhibited by the system to user interaction should be closely designed to meet task requirements.

- Presence refers to a certain feeling emerging from our capacity to act in a world, which is closely related, in graphical simulations, to the fidelity of the sensory input and output data.

These three properties form a cube in which the different system can be compared. For example, a system with all levels high should be "fully autonomous agents and objects that act and react according to the state of simulation and that are equally responsive to the actions of the human participant" (pp 1). Zeltzer (1) admits that he still finds difficult to rigorously quantify his proposed properties. In fact, it seems that autonomy and presence are somewhat tricky concepts: the first because it seems difficult to clarify strict boundaries between systems and their autonomy, especially if they are not reacting to the same stimulus; the second because the implicit notion of presence does not clearly take into account the subjective dimension and possible different weightnings that different sensory cues have on the feeling of presence that, additionally, might also vary from task to task.

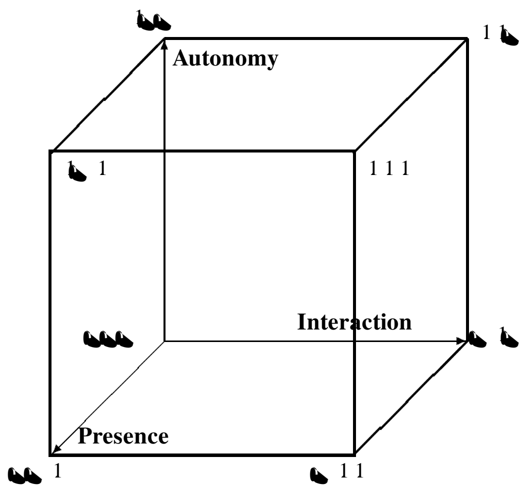


Figure 1: Adapted from Zeltzer (1 , pp 1)

Although difficult, a typology of graphical computer system

- technical fidelity the degree of realistic rendering colours texture motion etc that raises the question of their relative importance to different tasks and activities
- representational familiarity is the environment familiar?
- representational reality is the world possible?

- Immediacy of control the medium that the system uses where a set of natural behaviours for interface interaction corresponds to a more immediate system
- Presence our so many times referred that the authors consider to require subjective and objective measures

In a careful analysis one can find common issues in some of the frameworks revised and a different proposal should be towards integration Kalawsky 1 refers the similarities between Ellis and Zeltzer's proposals In fact we can see that the differences established by Ellis about objects and actors resembles Zeltzer's concept of autonomy and that Zeltzer's interaction implies the same issues has Ellis's dynamics The difference resides on the fact that Ellis expands his concepts considering that virtual environments are space metaphors and Zeltzer only pretends to cl achieved n particu ar the sense of presence can be di mi of information from one type of external representation to another, integrating and understanding the relationships between them. For example, Rogers and Scaife (1997) consider the interplay of concrete external representations and more abstract external representations in the comprehension of food webs. Considering that a certain way of presenting information allows a better comprehension is not enough (Scaife and Rogers 1996).

Perhaps in a usability perspective, let us now look to the general tasks that a user can perform in a VLE and their cognitive requirements. This perspective can aid us to have a more fine-grained understanding of the activities that can be implemented in a VLE and to realis

a [5] re detai ed cognitive ana ysis of graphica representations he e terna cognition approach wi be discussed further ahead

4.1 Types of learning in VEs

ic ens and Ba er 1 consider four types of earning tha hite oc et a 1 6 consider that research on educationa app ications of E's has been concerned with situation awareness or sensory or s i s here is no detai ed research assessing the re ationship between the structure and for of a E and the nature of the conceptua earning that ta es p ace he authors a so report Dede Loftin Sa z an Ca houn ob it and egian 1 research on e pirica eva uation of the effectiveness of E for ing conceptions and consider this wor va u ab e in the sense that it raises questions about the design of appropriate da ities for instructing the do ains or concepts

here are severa e a p es of E syste that were bui t for conceptua earning going fro bio ogy dyna ic and interactive representations here is a c ear n interactivity it may be possible to provide children with a more effective way of understanding and reconstructing the formal notations used to describe the concepts (Rogers and Scaife 1 pp

As we will refer these design principles can be used in con unction with the ones that Scaife and Rogers 1 consider at least in the initial phases of the research e thin that these design principles can be further specified and this will be one of the general goals of my wor

6 Lines of research

he analytical framework that a m constructing for my research involves

- find good dimensions for the characterization of graphical situ ations in order to understand what are the ey interactivity properties that these systems promote
- consider the characteristics that Scaife and Rogers 1 6 and Rogers and Scaife 1 propose for e ternal representations as main framework for the analysis of ERs
- use the design dimensions referred by Scaife and Rogers 1 6 and ic ens 1 and ic ens and Ba er 1 for the raise of pragmatical design research
- use the general tas s for Es that ic ens and Ba er 1 propose as w

object is central to the understanding of the problem, but manipulating a certain object in the archaeological site may not have any informational gain unless the recognition of the object is dependant on its manipulation. It is even possible that for focusing the learner attention on the relevant information a good design decision for the archaeological site would be to limit the possibilities of manipulating the object, considering the alternative of presenting information by a simple click of an input device.

The second problem is more confined and less prone to multiple interpretations. It is a matter of understanding how can we make more explicit the construction of an abstract representation through the use of more explicit and perceptually driven representations. The main issue is to know how to build the concrete representations and how to link them with the abstract modes.

It is clear, however, that the comparison between the two problems is not a conclusive one since I can not guarantee the perfect correspondence of the two. This should be seen more as two case studies with semi-controlled variables.

6.2 The archaeology site exploration

The problem that I want to investigate through the archaeology site exploration is the usefulness of more realistic simulations. More specifically, I want to understand the benefits of allowing a learner to explore a simulation of a certain location or environment that otherwise he would not be able to experience. In some sense, I am addressing the problem of what are the benefits of providing a concrete representation with high levels of interactivity in a domain that people can have difficulties to experience. This also considers the importance of providing levels of realism of the representation and realistic here must be separated in two different issues:

- the pictorial realism or technical fidelity according to Whitelock et al., 1996;
- the interactivity realism which should include the notion

when one considers complex domains. The complexity of the representation itself can be another problem. Since complex representations involve different cognitive properties in complex dynamics. So it can be the case that we can not avoid analysing complex perceptual problem investigating for example if the subject recognises the particular affordances of the representational object or if the object promotes good or bad analogies.

6.4 The basic interactivity properties in the VEs

The interactivity properties being investigated are:

- Visualizing: the benefits of the possibility to display information using an aditional dimension the 3D display. The use of 3D representations will have different goals in the problem. For the archaeological site will be a way to promote realism but for the stereographic site it will help the understanding of the problem states.
-

i ropou os 1 irtua environ ents in science e

Zhang J 1 . he nature of e terna representations in prob e so ving

n Ret ows y F P Ed 1
he 11th hite ouse Papers Cognitive Science Research Paper
Schoo of Cognitive Co puting Sciences University of Susse

Developing an experiment workbench to study software reuse from a cognitive perspective

Fabrice Retkowsky
fabricer@cogs.susx.ac.uk

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

Abstract Software reuse as a promising programming technique has led to many technological developments But it also involves programmers' cognition and different theories

Figure 1 E peri mental setup

3 The experiment

3.1 Design and materials

he e peri ment consisted in as ing 1 Java beginners to program si mple class by reusing a class from the Java A pac ages

he Java A pac ages are standard pac ages of classes written by Sun the creators of Java

4 Results

4.1 General remarks

The subjects all followed the same pattern of programming. First, they read and tried to understand the problem description (it took 1 minute on average). Then they looked for a component to reuse ([illegible] minutes on average, though some subjects didn't search at all) and finally did the programming. Only two tests (out of [illegible]) required more than [illegible] minutes.

Variable	PhoneNumber	PhoneNumberFormat
Total time	1[illegible] mins	16 [illegible] mins
No. of pages when searching	[illegible] pages	[illegible] pages
No. of pages when programming	[illegible] pages	[illegible] pages
Quality of the code	[illegible]	[illegible]
Percentage of reusers	[illegible]	[illegible]
Suitability of the components	[illegible]	1[illegible]
Quantity of reuse	[illegible]	[illegible]

Table Results for the PhoneNumber and the PhoneNumberFormat tasks

more API pages while programming probably because the programming was longer. The quality of the resulting code is the same for both tasks though the eval

Finally the situation produced better programs mainly because led to less reuse and programs based on reuse were judged of lesser quality see 4.

4.5 Expertise

hough the subjects were all beginners there were initially two measurements of their expertise in Java

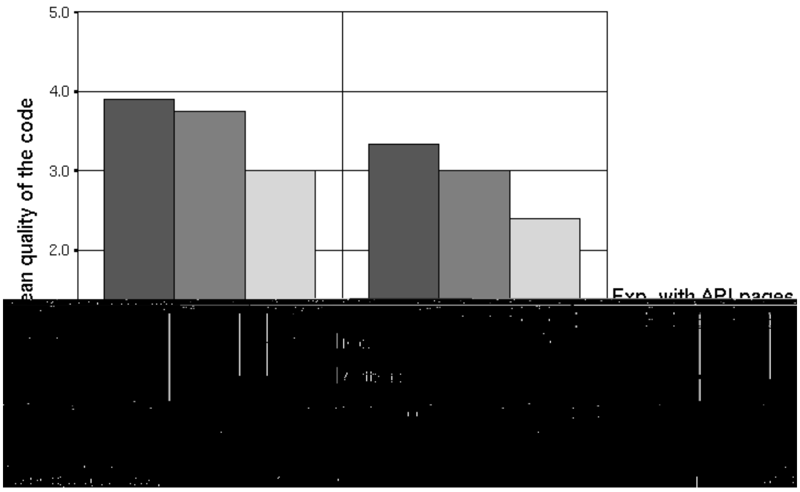


Figure 4. Experts underperform whether they reuse or not

probably read the descriptions more thoroughly when searching, as opposed to 'beginners' who just browse, and that they knew how to use the API pages as a programming help.

5 Consequences for the Design

Once the subjects completed their two tasks, they were asked three open-ended questions about software reuse:

- What are, in your opinion, the good aspects of the API pages as a reuse tool?
- What are, in your opinion, the bad aspects of the API pages as a software reuse tool?
- What should a perfect reuse tool look like?

The answers we collected can be found in Section 7. From these answers, and from the points we made in the numerical analysis, we can draw some guidelines for the design of the reuse tool. Some of these guidelines are already met by the initial design, some lead to a few modifications.

Component description As we saw before, the reuse tool will be based on a set of modules. One of the most important modules is the component description. What appeared from the experiment's

Therefore we will base the first version of the component description module on the API documentation. Yet the subjects suggested a few modifications:

- the class and package names should also be self explanatory for beginners
- it should have less technical terminology
- it should also describe the code itself, and make it easily accessible, or even include it in the description – particularly for the 'Understanding' stage
- it should include some examples
- the packages should have a description as well
- it should be less complicated and shorter. This is easily feasible for the search stage: the experiment proved subjects don't use the 'Methods' level.

Navigation Since navigation is an important issue, we initially designed a complete and efficient set of navigation tools. The subjects reminded us that the navigation should be very simple, i.e. like the API in HTML, and that:

- it should provide something so that users don't get lost
- the Search stage should actually have a search tool
- it should always suggest alternative possibilities, so that the user does not get trapped in one not so good solution
- it should assist but not be intrusive

As a consequence, it was decided to keep the navigation tools to a minimum, that is, a bar menu and a small wizard that allows quick navigation between the four stages of reuse.

Structure Finally, the system should include some editing tools, to specialize and integrate the components, and a built in compiler, which was lacking from the experiment's rudimentary reuse setup. These were not planned at first, but will be included in the Specializaton and Integration stages.

6 The next steps in the development of the tool

Since this experiment was completed, we have developed a mock-up of the user interface, and have had it tested by a few possible users. The next step consists in designing the experiment tool kit, and then programming the whole reuse workbench for real.

Once the system is completed, we will perform the experiment described in this paper again, but using our tool instead of the rudimentary reuse system used here. This will have two aims:

- to test wether there are any major flaws in the design of the tool, or whether subjects have any problem in using this kind of integrated tool that guides and assists them
- and to know wether, in a simple configuration based on the API documentation, our system already brings some kind of benefits.

Finally, we will develop and compare some new sets of modules, for example to evaluate alternative search techniques or documentation styles.

7 Appendix 1: The four tasks

7.1 Task A1

rite a honeNu ber N class by reusing a Java A class

- A N ob ect wi contain a te ephone nu ber such as 1
- t wi be initia ized using a String para eter i e
- t wi have a toString ethod which wi give bac a String such as 1

You HAVE to reuse a Java API class to write this class.

7.2 Task A2

ere is the honeList class which is used in an 'Organizer' progra

- t is basica y a ector of honeNu ber ob ects
- he 'Organizer' progra creates such honeLists adds honeNu bers to the re ove honeNu bers and print

```
public class PhoneList
{
    int MaxSize = 3;
    PhoneNumber[] PhoneArray = new PhoneNumber[MaxSize];
    int NbNumbers = 0;
// position 1 for PhoneArray[0]

    PhoneList()
    {
// creates two default numbers
        PhoneNumber OneNumber = new PhoneNumber("1111111111");
        PhoneNumber NineNumber = new PhoneNumber("9999999999");

        this.addNumber(OneNumber);
        this.addNumber(NineNumber);
        this.printNumbers();
    }

    public boolean addNumber(PhoneNumber aNumber)
    {
        if (NbNumbers == MaxSize)
    return false;
        PhoneArray[NbNumbers]  = aNumber;
        NbNumbers++;
        return true;
    }

    public boolean removeNumber(int position)
    {
        int i;

        if (position > NbNumbers)
    return false;
        if (position == NbNumbers)
    {
        PhoneArray[position] = null;
        NbNumbers--;
        return true;
    }
        for (i=position; i<NbNumbers; i++)
    PhoneArray[i-1] = PhoneArray[i];
        NbNumbers--;
        return true;
    }

    public void printNumbers()
    {
        int i;

        if (NbNumbers == 0)
    System.out.println("Empty List");
        else
    for (i=0; i<NbNumbers; i++)
        System.out.println("Phone n. "+i+": "+PhoneArray[i].toString());
    }
}
```

7.3 Task B1

> rite a [illegible] hone[illegible]u [illegible]ber[illegible]or [illegible] [illegible] c[illegible]ass by reusing a Java A[illegible] c[illegible]ass
>
> - A[illegible] ob ect wi[illegible] be ab[illegible]e to for [illegible] so [illegible] strings
> - t wi[illegible] for e a [illegible]e for [illegible] [illegible] into Brighton [illegible]
>
> **You HAVE to reuse a Java API class to write this class.**

7.4 Task B2

```
import java.awt.*;
import java.applet.*;

public class PhoneWidget extends Applet
{

// The interface attributes
    TextField input = new import1lic
```

References

Bur hardt J M Detienne F 1 A La reuti isation en genie ogicie une de nition d'un cadre de

n Ret ows y F P Ed 1
he 11th hite ouse Papers Cognitive Science Research Paper
School of Cognitive Computing Sciences University of Susse

Structural Knowledge in Prolog

Pablo Romero*

juanr@cogs.susx.ac.uk

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

Abstract Knowledge about which parts of the program me t are important to e perienced programmers might give clues about the nature of their structural knowledge and also might be useful for the design of instructional system for programming here have been sev eral studies that have suggested that programmers of proced

here are several models of structural nowledge proposed for Prolog however these models are not supported by psychological evidence he e periment described in this paper tried to find out which

of program comprehension for Prolog that could characterise the flow of information and the temporal ordering of information relationships of this process. The information relationships that they consid

According to Brooks (1) and Wiedenbeck (1 6) programmers use beacons in the code to guide their comprehension process. Davies (1 1 4) suggested that these beacons can be considered as the external analogue of the internally represented focal structures of the programmer's structural knowledge. This structural knowledge seems to be based in the idea of programming plans for the case of procedural languages, but this might not be the case for Prolog. It could be that either the idea of a plan has not counterpart in Prolog or the nature of plans is very different in this language.

3.2 Design

As mentioned before, this is not a hypothesis testing experi

```
   do_sort
/             /

do_sort SortedList
          write 'enter sorting data'
          read Key
          ne  t_va_ue Key List
          bubb_esort List SortedList

ne  t_va_ue stop

ne  t_va_ue Key  Key|Rest
          write 'enter sorting data'
          read NewKey
          ne  t_va_ue NewKey Rest

bubb_esort SortedList SortedList
          verify_sorted SortedList

bubb_esort List SortedList
          swap List List1
          bubb_esort List1 SortedList

verify_sorted

verify_sorted  X

verify_sorted  X Y|Rest
          X =< Y
          verify_sorted  Y|Rest

swap  X Y|Rest   Y X|Rest
          X > Y

swap  Z|Rest   Z|Rest1
          swap Rest Rest1
```

Figure 4 A version of the bubble sort program schemas related to data structure information. Finally, control flow relations seem to be highlighted by the points were recursion takes place.

Finally, as the experimental task includes the identification of the program functionality, 'disguised' versions of these programs have to be presented to the subjects. The criteria to 'disguise' these programs is similar to the one used by Wiedenbeck (1986).

3.5 Procedure

The programmer subjects of this experiment performed three similar sessions. In each session, they were given a hardcopy of the experimental program and were asked to study and memorise it. This study period lasted minutes. After this, the subjects were given minutes to recall and write down what they could remember of the program. Finally, these subjects used another period of minutes to write down a short explanation of what, according to them, the program does. These estimated times were calculated following the same proportions as the times for the Wiedenbeck (1986) experiment.

The control group followed a slightly different procedure. They were not instructed to comprehend but only to memorise the program. Also, they were not asked to write down an explanation of what the program do.

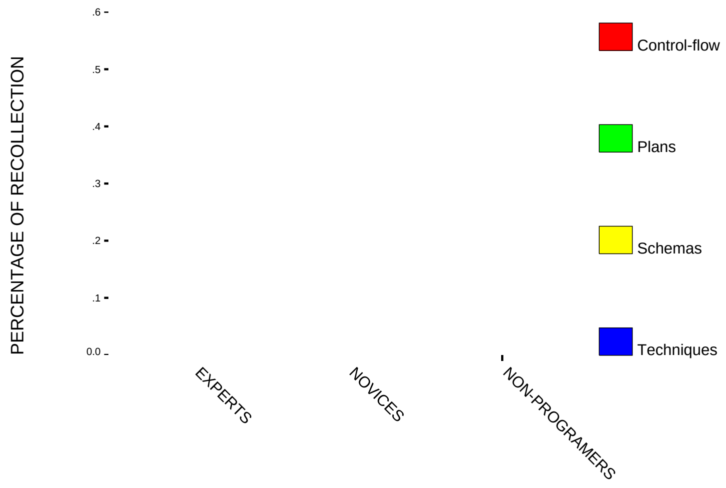
As this was a pen and paper exercise, the collected data was the hand written account of both the recollection and the explanation of the functionality of the program by the subjects. The recollection account was analysed for each one of the proposed structures. For this analysis, the success rate of recollection of each one of the structure's instances was calculated for every subject. These instances were considered as correctly recalled if the subject wrote a verbatim copy of the program code for these segments, however commas, dots, spaces and indentation were not considered in determining this success of recollection. Also, if the subject recalled a procedure or variable with a different name but was congruent through all the program with this modification, then the modification was not considered as an error.

As mentioned before, structure instances comprise several elements which could be scattered through several lines of the program. Only recollection of whole instances was considered as successful recollection. If, for example, of a particular instance was correctly written down, this instance was not considered as correctly recalled. This strict criteria for the recollection of instances was considered as appropriate because this study is interested in which structures, as chunks, are relevant to programmers.

The programmers' account of the program functionality was analysed in terms of its correctness. Each of the programs performs several functions. The program, for example, reads a list of values and obtains an average of these values and their minimum occurrence. The program validates a binary number and converts it to its decimal equivalent and the program (figure 4) reads a list of values and performs a sort procedure over them. The subjects' functionality description was required to mention these major functions in order to be considered correct. For example, for the program statements equivalent to 'read a list of values', 'obtains an average' and 'obtains the minimum' were searched in the subjects' description. Only if the three statements were identified the account was considered correct. A similar criteria was used for the other two programs.

3.6 Results

The data of this experiment was analysed in three parts. The first part dealt with the success rate in identifying the function of the program by the two groups of programmers. The second part was concerned with comparing the success percentages of recollection for the four structures taken into account. The third part compared the percentage of recollection of each structure versus the percentage of recollection of the program lines.


PERCENTAGE OF RECOLLECTION
.6
.5
.4
.3
.2
.1
0.0
EXPERTS
NOVICES
NON-PROGRAMERS
Control-flow
Plans
Schemas
Techniques

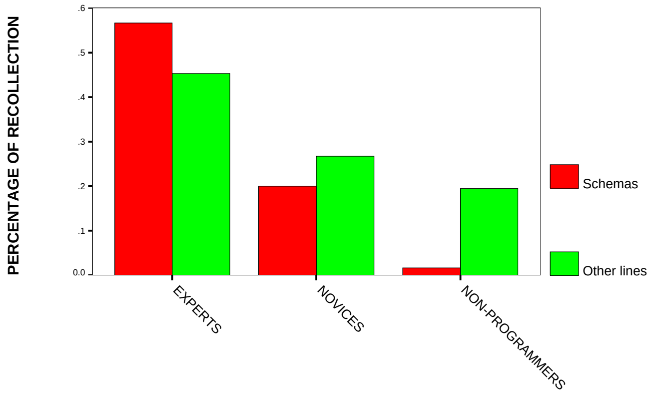


Figure Percentage of recollection for schemas and lines outside the

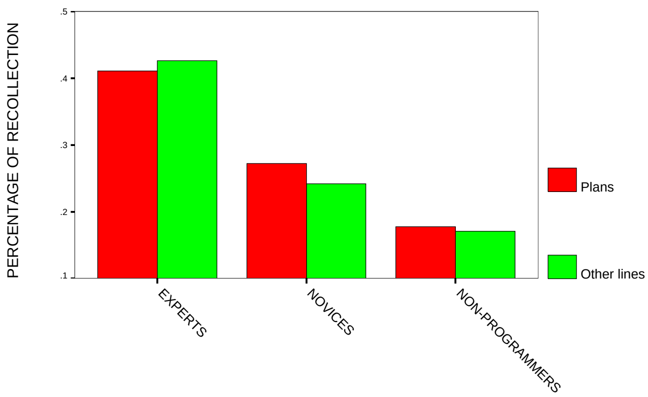


Figure Percentage of recollection for focal structures of plans and lines outside the these co parisons for sche and p ans he resu ts for techniques and contro ow oo very si ar to those for p ans

As direct co parisons between the groups of progra s had a ready been perfor d in the previous ana ysis and a so in order to avoid isation and size effects the statistica ana ysis for this third part of the study focused in the rate of change of the dif when considering only the bubble sort program, which is similar to the sort program Wiedenbeck uses, are basically the same to those obtained when taking into account the three programs. So it seems that the key difference is the programming language considered.

It seems reasonable to think that in absence of any other information, neither internal nor external documentation, and with variable and procedure names that do not help much to grasp the meaning of the program, patterns of typical operations performed over familiar data structures can be very important to start making sense of the code. This lack of documentation and meaningful variable names seems to be an important issue for Prolog. Green, Bellamy and Parker (1987) mention that Prolog, 'due to its poor role expressiveness', is specially sensitive to naming style. 'Salient variable names are almost the only method of making a Prolog program role expressive, and thereby revealing the plan structures' (Green et al., 1987). The obvious question is how naming style influences the program comprehension (mental model), or in other words, which aspect of the program structure relations are important at the beginning of the comprehension process for Prolog programmers.

The results of this experiment suggest that the mental model that Prolog programmers build at early stages of the comprehension process is different from the one that programmers of procedural languages construct. The former seems to be based on data structure relations while the latter, according to Wiedenbeck (1986) and Davies (1994), is related to functional information. This conclusion needs to be confirmed and its importance needs to be related to a more common programming task such as debugging or program modification. Only when it is established that these findings are important for actual programming tasks, this information could be applied to influence programming instruction and the design of programming tools.

References

Bergantz, D. & Hassell, J. (1991). Information relationships in PROLOG programs: how do programmers comprehend functionality? 1–

Bowles, A. & Brna, P. (1993). Programming plans and programming techniques. In

Brna, P., Bundy, A., Todd, T., Eisenstadt, M., Looi, C. K., Pain, H. (1991). Prolog programming techniques. 111–1

Brooks, R. (1983). Towards a theory of the comprehension of computer programs.

Davies, S. P. (1990). The nature and development of programming plans.

Davies, S. P. (1993). Models and theories of programming strategy.

Davies, S. P. (1994). Knowledge restructuring and the acquisition of programming expertise.

Gegg-Harrison, T. S. (1991). Learning Prolog in a schema-based environment. 1 –1

Gilmore, D. & Green, T. (1988). Programming plans and programming expertise.

Green, T. R. G., Bellamy, R. K. E., Parker, J. M. (1987). Parsing and Gnisrap: a model of device use. In Olson, G. M., Sheppard, S. & Soloway, E. (Eds.) pp. 1–1. Norwood, NJ: Ablex.

Ormerod, T. C. & Ball, L. J. (1996). An empirical evaluation of TEd, a techniques editor for Prolog programming. In pp. 1–16. Norwood, NJ: Ablex.

Pennington, N. (1987). Stimulus structures and mental representations in expert comprehension of computer programs. –1

Wiedenbeck, S. (1986). Beacons in computer program comprehension.

n Ret ows y F P Ed 1

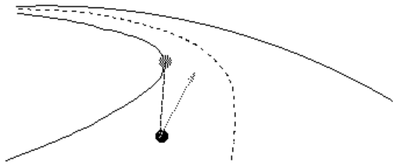


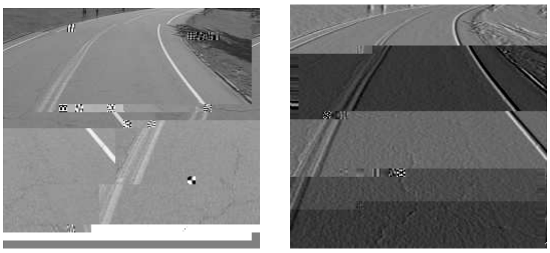
Figure 1 Tangent Point

scenarios to which the tracking algorithm is applied and analyses the effect of vibration to the tracking performance. Finally, Section 4 concludes the paper and reflects the main results.

2 Method

The tracking mechanism introduced in this paper makes use of well established and reliable image processing techniques and is designed to process image sequences with little scenery, such as shown in the images from Figure 6 and [illegible]. Image sequences containing more scenery, such as shown in Figure [illegible], can also be processed and produces reasonable results.

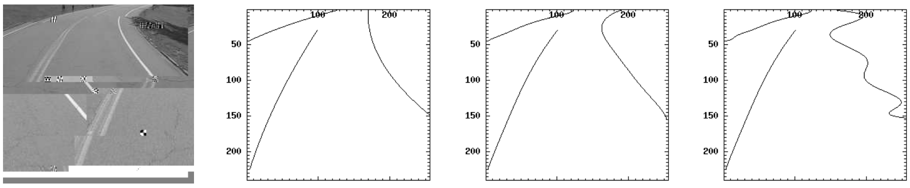
A tangent point forms the location where the driver's direction of gaze and the extreme inner point of a road bend touch but do not intersect. The road schematic in Figure 1 illustrates this. The black dot represents an approaching car and the grey dot represents the tangent point on the road bend. The dotted arrow indicates the car's current heading and the dashed line shows the driver's direction of gaze, which touches the tangent point.



a b

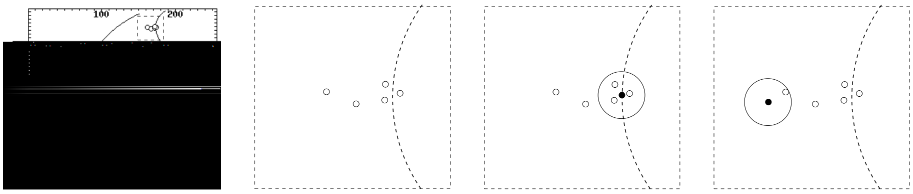
Figure S oothing and A is Differencing

of neighbouring c usters are ocated very c ose to each othe



a b c d

Figure Degree of Polynomial



a b c d

Figure 4 Tracking the Tangent Point

The extraction of potential tangent points from polynomials is a standard procedure and requires the first derivative (gradient). The tangent point (as shown in Figure 1) is the point in which the driver's direction of gaze and the extreme inner bend of the road touc

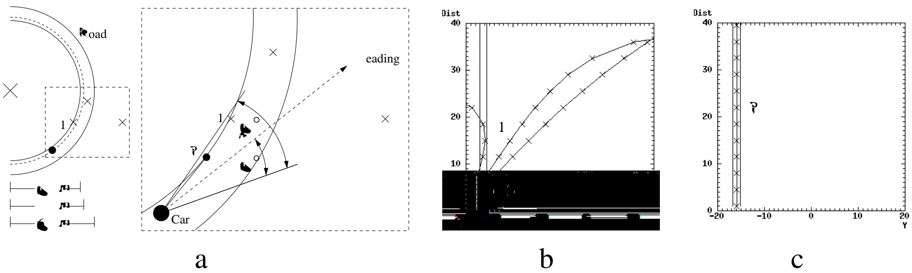
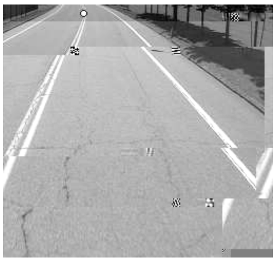


Figure he angent Point and Noise

a neighbouring range of at least two other tangent points in the history. A "neighbouring range" in the context of this program means that at least two tangent points from the history have to be within a close radius of the new tangent point. In (c) three previous tangent points are within the radius of the new



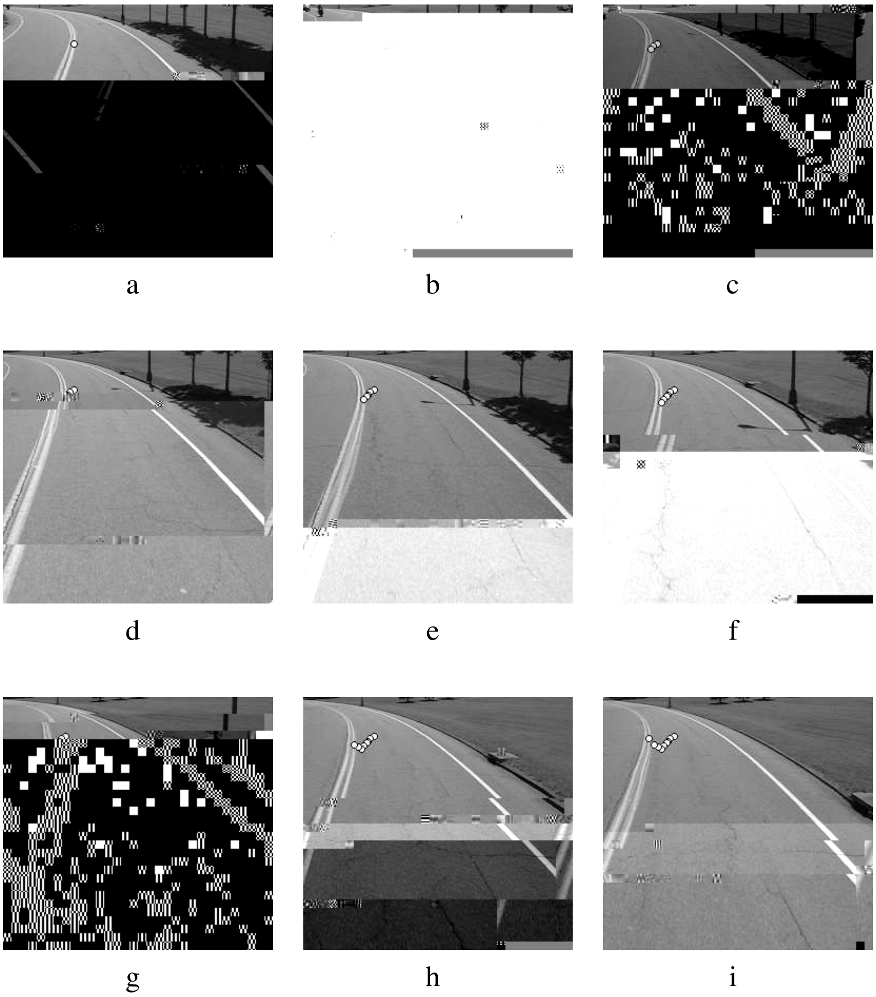
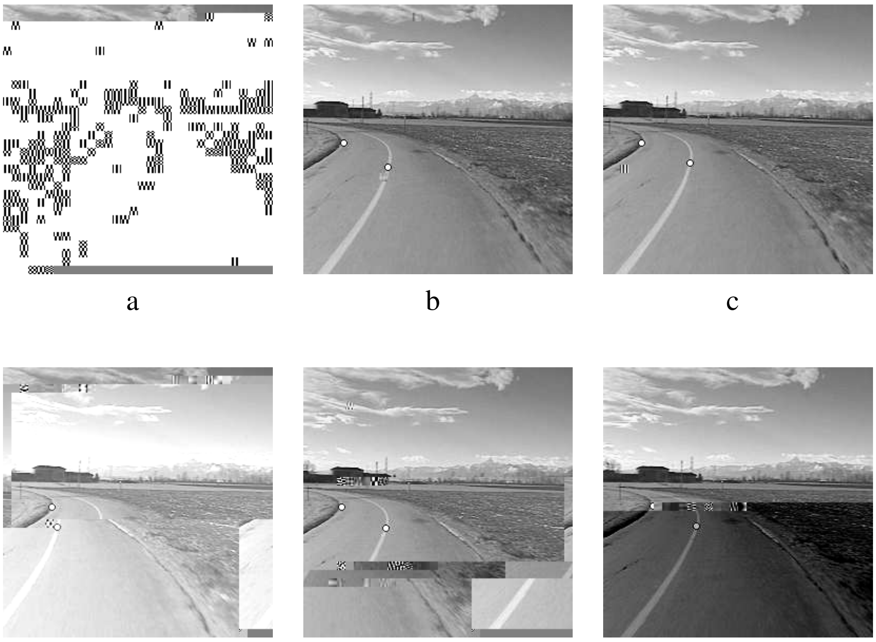


Figure C ustering of angent Points

3.3 Several Tangent Points

he sequence in Figure shows that the trac ing a gorith is not restricted to a set nu ber of tangent points he road shows two tangent points which cou d be fo owed by the driver the curve of the



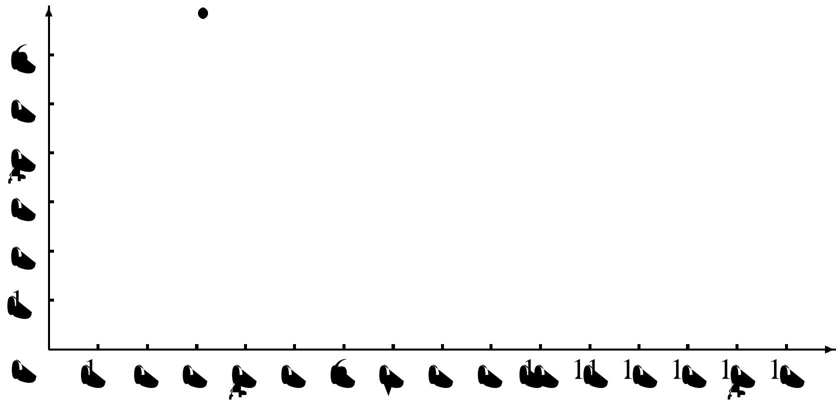
a b c

d e f

Figure Several Potential Tangent Points

the perfor ance of tangent point trac ing abi ity under increasing ca era vibration[1]

he sequence consists of 1 continuous road i ages in which the progra can detect 4 tangent points correct y and 1 tangent point incorrect y with no added ca era vibration he graph's vertica a is shows the a ount of detected i age regions and the horizonta a is shows the a ount of rando y added horizonta ca era vibration in units of pi e he so id b ac dots indicate the a ount of correct y



?o er eau D 1 ?a ph ?apid y adapting atera position hand er ech rep he ?obotics nstitute Carnegie ?e on ?niversity ?ittsburgh ?A 1 1 ?SA

?einhardt ? Soeder 1 ? edition o Deutscher aschenbuch er ag

Sera ?n C 1 ?re i inary e a ination of driver eye ? ations on rura roads nsight into safe driving ech rep ?niversity of ?ichigan ransport ?esearch nstitute ? ?

n Ret ows y F P Ed 1
he 11th hite ouse Papers Cognitive Science Research Paper
School of Cognitive Co puting Sciences Oniversity of Susse

Autopoiesis and Search

Oliver Sharpe
olivers@cogs.susx.ac.uk

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

1 Introduction

nown including which solution is the best solution and therefore if we are using the search space to find a solution to a real problem it is no longer useful to search the space as we already now which solution is the best owever if we still ran the same search algorithm on this same problem the dynamics of the search algorithm would be the same t has no way of nowing that we already now the best solution herefore if we want to me a definition of search processes that is based on the dynamics of the

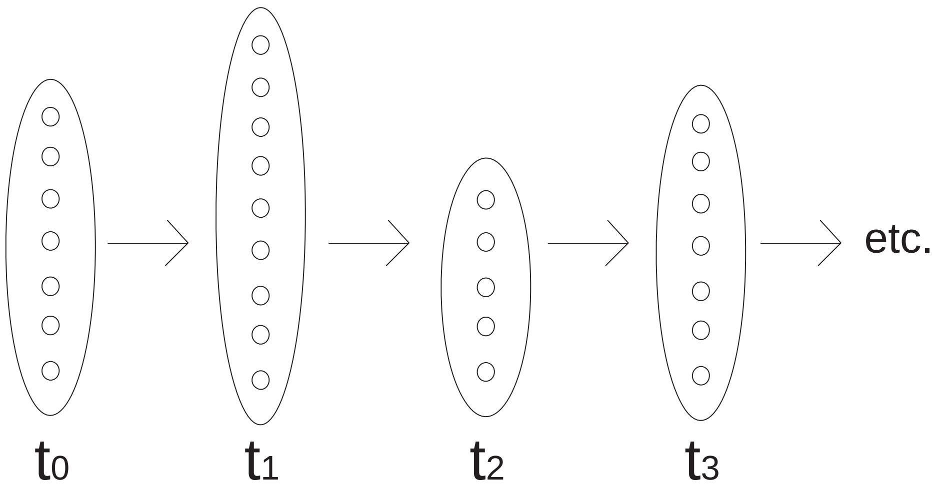
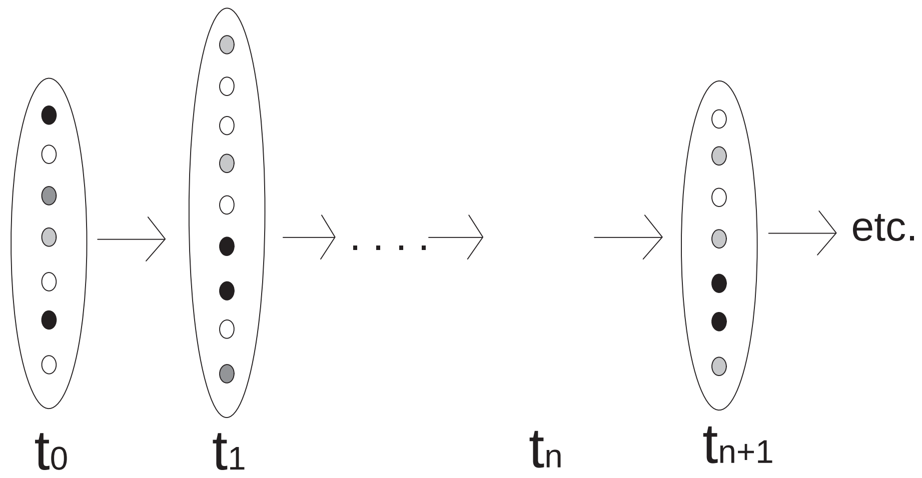


Figure 1 The structure of an active set trace

the structure of a history trace is a list (or time series) of objects. The history trace also shows us the order in which the search process performed work, as each evaluation is considered to be a unit of work. Hence the history trace shows us the work dynamics of a search process.

A significant aspect of the search process that the history trace ignores is the set of evaluated points from which the next candidate solution will be generated. Most search algorithms use hints gained from previously evaluated points to help direct the future search direction. However, most do not remember the whole of the history trace. Instead algorithms keep a sub


. . . .
etc.
t_0
t_1
t_n
t_{n+1}

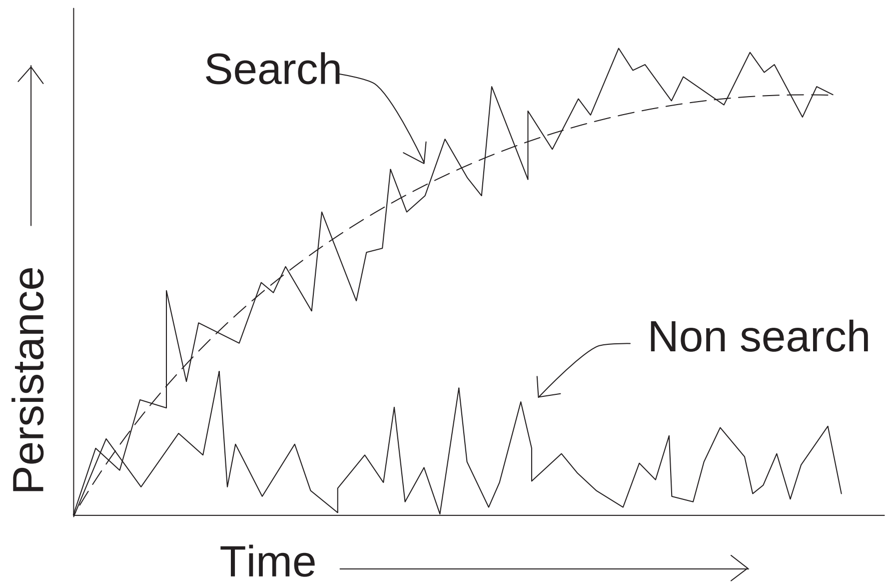
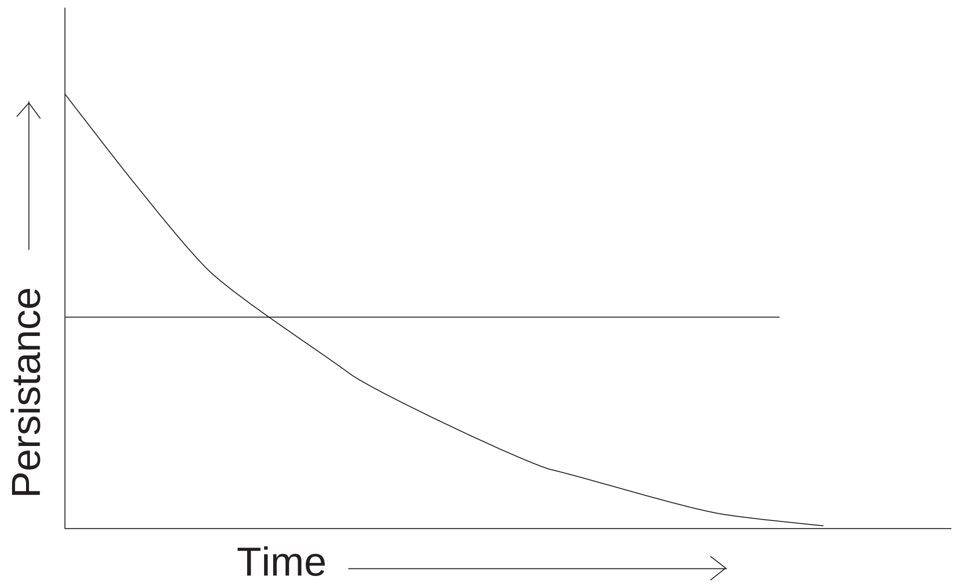


Figure Distinguishing between search and non search processes

Most important is that a random process will not fall into this definition and neither will the process of a hill climber on a needle in a haystack landscape or indeed flat landscape and I am very happy with that too. The dynamics of a hill climber on a needle in a haystack landscape is what one might call a termination dynamic where the active set trace is an ap


Persistance
Time

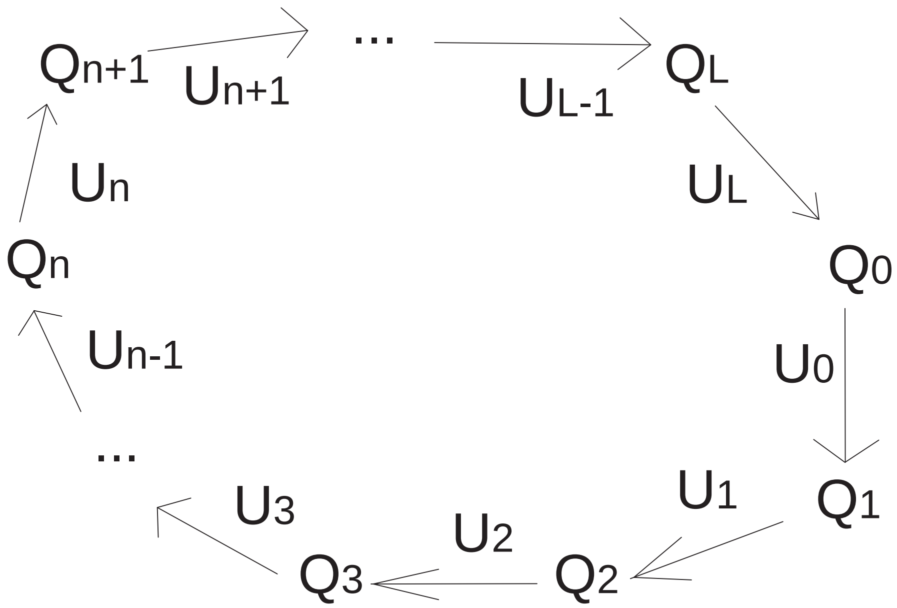
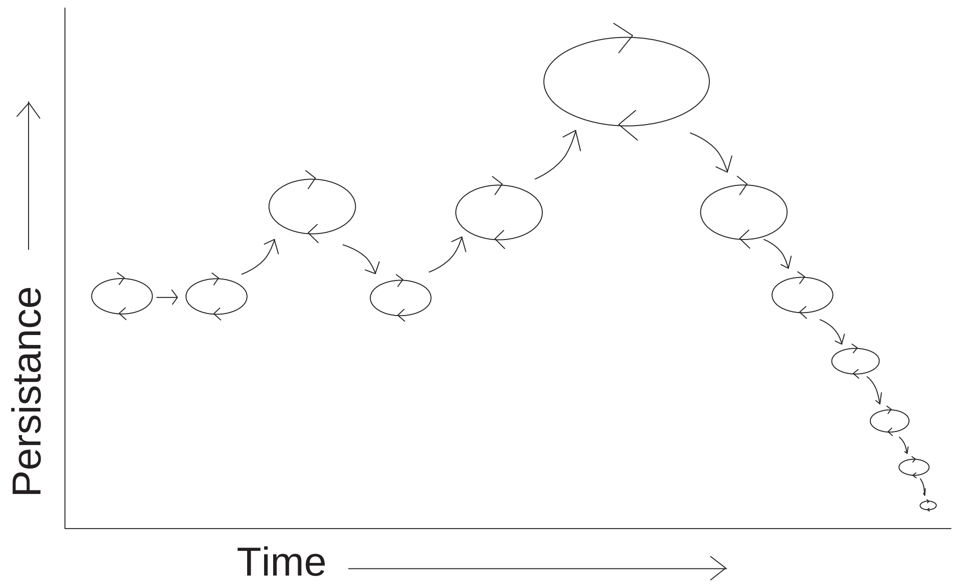
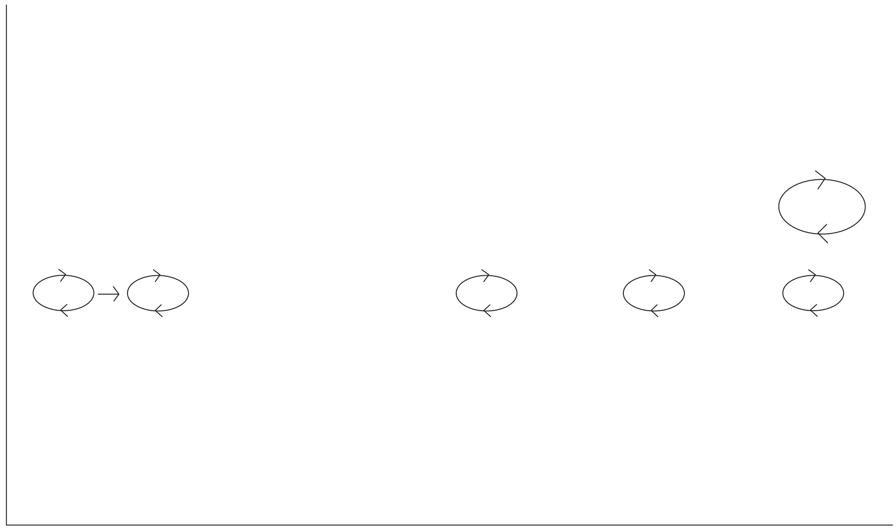


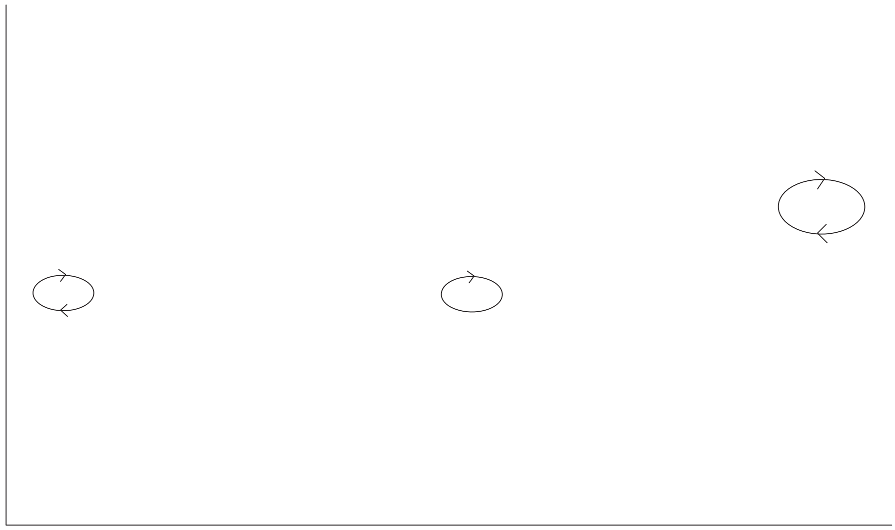
Figure 6 The formal structure of a Cyclically Persisting Relational Object (CPRO)

The model is as follows

1.0


Persistance
Time





that it was ‘playing’ in an approximation of infinite resources. Once life had filled the space available introduced the persistence ratchet he persistence ratch

The ultimate take home message of the paper is that if you want to study autopoiesis you have to be using a dynamical system that can support search processes either by allowing exponential growth or by using a search algorithm

References

Davis L Ed 1

n et ows y Ed 1
he 11th hite ouse apers Cognitive Science esearch aper
School of Cognitive Computing Sciences University of Susse

The Effect of Mood on the Accessibility of Reasons Why Positive or Negative Future Events Might Happen: An Application of Availability Heuristics to Worry-based Pessimism

Helen M. Startup & Graham C.L. Davey †
helenst@cogs.susx.ac.uk and grahamda@cogs.susx.ac.uk

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

Abstract Ob ectives
Because patho ogica worriers have unrea istica y high e pectations of negative events hap pening the present study investigated the effect of mood on the generation of reasons why future events might happen and on udge ents about the probabi ity of such events hap pening

Design
he study used a between sub ects design in which different groups of nonse ected sub ects were given either negative positive or neutra mood inductions

Methods
After separate mood inductions sub ns nducuc ne eci

1 Introduction

prevented than how it can be prevented

n a test of the app ication of the avai abi ity heuristic to e p aining worry re ated pessi is about future negative events ac eod i ia Be erian 1 1 found that severe worriers tended to generate ore reasons than nonworriers why putative future negative events ight happen and fewer reasons and complete some tasks. Finally, subjects were told that in order to alleviate boredom a short break would occur in the proceedings in which they would be asked to relax and listen to a short extract of music.

Stage 1 All subjects were asked to complete the Penn State Worry Ques

3 Results

3.1 Penn State Worry Questionnaire (PSWQ)

[illegible]ean score on the [illegible] Q for a[illegible] sub ects was [illegible] sd 1 [illegible] his co [illegible]res with a [illegible]an score of [illegible] found in genera[illegible] unse[illegible]ected sa [illegible]es of A [illegible]rican sub ects [illegible]=1 [illegible]o[illegible]ina Bor ovec 1 and a [illegible]an score of [illegible] found in ana[illegible]ogue c[illegible]inica[illegible] sa [illegible]es diagnosed as GAD by GAD Q screening [illegible]= [illegible]o[illegible]ina Bor ovec 1 [illegible] here was no signi[illegible]cant difference in [illegible] Q across the three [illegible]od induction groups [illegible] = 1 p= [illegible]ean [illegible] Q scores for the three groups were [illegible] sd 1 [illegible]1 [illegible] sd 11 [illegible] and [illegible] sd 1 for [illegible]egative [illegible]ositive and [illegible]eutra[illegible] Groups respective[illegible]y

3.2 Mood measures

[illegible]ab[illegible]e [illegible] shows the [illegible]an an iety sadness and happiness [illegible]asures for each Group both before and after the [illegible]od induction [illegible] hese were sub ected to a group negative vs positive vs neutra[illegible] ti [illegible] pre induction vs post induction ana[illegible]ysis of variance An iety ratings e hibited a signi[illegible]cant group ti [illegible] interaction [illegible] = [illegible] p[illegible] 1 A[illegible]though there was no signi[illegible]cant difference between groups on the pre induction [illegible]asure [illegible]= LSD a[illegible] ps[illegible] there was a signi[illegible]cant difference between groups on the post induction an iety [illegible]asure [illegible] = [illegible] p[illegible] 1 [illegible]he [illegible]egative Group reported signi[illegible]cant[illegible]y higher post induction an iety ratings than both the [illegible]ositive and the [illegible]eutra[illegible] Groups LSD both ps [illegible] here was no signi[illegible]cant difference in post induction an iety ratings between groups [illegible]ositive and [illegible]eu tra[illegible] [illegible]hese data were a[illegible]so sub ected to an an ious ti [illegible]1 an ious ti [illegible] within sub ects ana[illegible]ysis of variance [illegible]his e hibited a signi[illegible]cant an iety ti [illegible] interaction [illegible] = [illegible] p[illegible] re[illegible]ected as a signi[illegible]cant increase in an iousness between ti [illegible]1 and ti [illegible] in the [illegible]egative Group [illegible]an diff=1 p[illegible] co [illegible]ared with a decrease in the [illegible]ositive and [illegible]eutra[illegible] Groups

[illegible]ood	An iety		Sadness		appiness	
	[illegible]re induction	[illegible]ost induction	[illegible]re induction	[illegible]ost induction	[illegible]re induction	[illegible]ost induction
[illegible]egative	[illegible]	[illegible] 1 [illegible]	1[illegible]	[illegible] 1 [illegible]	[illegible] 1 [illegible]	[illegible]

appiness ratings a so e hibited a signi cant group rating interaction F =1 p 1 re induction happiness ratings did not differ signi cant y across groups LSD a ps but did differ signi cant y between groups at the post induction stage F =1 p 1 appiness ratings for the egative Group were signi cant y ower at post induction than ratings for both the ositive and eutra Groups and the ositive Group showed higher post induction happiness ratings than the eutra Group LSD a ps hen sub ected to a within sub ect ana ysis of variance the happiness ti 1 vs ti group interaction was signi cant F 1 =1 p 1 re ecting the fact that happiness ratings go down in the egative Group an diff = p 1 but up in the ositive and eutra Groups

3.3 Reasons Task

Figure 1 not inc uded shows the response pro es' for each 1 1 case of the Positive Group than in the case of the Neutral Grou a negative mood to judge negative events as more likely than individuals in induced positive or neutral moods

MacLeod et al 1 1 have used this relationship between the accessibility of reasons why events might happen and judgements about the probability of the event occurring to explain why severe worriers have significantly higher estimates of bad future events happening to them than do nonworriers see also Tasey Borkovec 1 However their explanation is based on worriers having already elaborated these reasons and them being more readily accessible than reasons why negative future events might not happen The present results suggest that this relationship between number of articulated reasons and probability judgements can also be found in an unselected population of subjects when mood is manipulated This indicates that prior elaboration of reasons through chronic worrying is not a necessary condition for the simulation heuristic to account for event probability estimates but that mood also appears able to influence the accessibility of reasons why events might occur

The main effect of mood on the generation of reasons why future events might or might not happen appears to be a reciprocal one in which negative mood influences the number of reasons why a bad future event might happen and positive mood influences number of reasons why a positive future event might happen Neither mood state influences reasons why events might not happen The influence of mood on reason generation for incongruent scenarios may however be varied Whereas the response profile of negative mood subjects in the context of a bad scenario clearly differentiates from that of subjects in a neutral mood there is no significance difference between the profile of these groups in the context of the good scenario This suggetsts that a negative mood solely exerts its influence on reasons for an event occurring In contrast the pattern is more complex in the case of a positive mood The response profile of subjects in a positive mood in the context of a good scenario differed significantly from those in an induced neutral mood interestingly however there was also a significant negative difference in response profile for these groups in the context of the bad scenario This finding suggests that perhaps a positive mood has an inhibitory effect on incongruent scenarios unlike a negative mood that merely exerts its influence on pro reasons The findings from the negative and neutral mood conditions are consistent with the fact that people find it easier to retrieve reasons why an event would happen rather than retrieve reasons why an event would not happen Dunning Parpal 1 Indeed data from the neutral mood induction group demonstrate that in the absence of a mood manipulation pro reasons are significantly more readily generated than con reasons regardless of whether the event being considered is a positive or negative one Furthermore if a mood is congruent with the valency of the event for which reasons are being sought e g thinking of reasons why a bad event will happen while in a negative mood such reasons are likely to be more readily retrieved than if the mood and valency of the event are incongruent Blaney 1 6 Bower 1 1 Teasdale 1 and when pro reasons are more accessible this appears actively to inhibit con reasons Tversky Kahneman 1 Koch 1 These processes may well contribute to the effect of negative and neutral mood solely on the generation of pro reasons and not con reasons Moreoever given that reasons for a good event in a positive mood were offered in most abundance by the response group and also considering that individuals are prone to persist at processing which maintains a positive but not a negative mood Sinclair and Mark 1 Martin and Stoner 1 6 one might indeed expect an inhibitory effect of incongruent scenarios in a positive but not a negative mood

Processes which facilitate the elaboration or accessibility of reasons why bad or negative future events might happen will provide some explanation of why chronic worriers have such unrealistically high judgements about the likelihood of such events happening Tasey Borkovec 1 MacLeod 1 For example the catastrophising process characteraihaih scenarios about the worry topic Davey Levy 1 asey Bor ovec 1 and this iterative process a lows for both the generation and e aboration of reasons why bad future events ight happen Si i ar y the dysphoric and negative ood frequent y asso

Davey G C L Levy S 1 Catastrophic worrying persona inadequacy and a perseverative iterative sty e as features of the catastrophising process Sub itted

Davey G C L a ton J Farre J J Davidson S 1 So characteristics of worry Evidence for worrying and an iety as separate constructs ersona ity ndividua Differences 1 1 1

Davey G C L Jubb Ca ron C 1 Catastrophic worrying as a function of changes in prob e o ving con dence Cognitive herapy esearch

Davey G C L a is F Capuzzo 1 he pheno no ogy of non patho ogica worry A pre i inary investigation n G C L Davey F a is Eds orrying erspectives on theory assess nt and treat nt Chichester i ey

Dunning D arpa 1 enta addition versus nta subtraction in counterfactua reasoning On assessing the i pact of persona actions and ife events Journa of ersona ity Socia sycho ogy 1

och J S 1 Avai abi ity and interference in predictive udge nt Journa of E peri nta sycho ogy Learning e ry Cognition

Johnston Davey G C L 1 he psycho ogica i pac isoc i A anden B orse C 1 he orry Sca e as a asure of an iety a ng ho bound and co nity e der y Behavior herapy 1