THE EIGHTH WHITE HOUSE PAPERS
Graduate Research in the Cognitive and Computing Sciences at Sussex

editors:
A. Jonathan Howell & Joseph A. Wood

CSRP 390

November 1995

ISSN 1350-3162

UNIVERSITY OF SUSSEX AT BRIGHTON

Cognitive Science
Research Papers

THE EIGHTH WHITE HOUSE PAPERS

Graduate Research in the Cognitive and Computing Sciences at Sussex

CSRP 390

editors:

A. Jonathan Howell & Joseph A. Wood

November 1995

Contents

Preface

Each year since 1988, COGS graduate students have been meeting at Sussex University's conference centre, the White House, located at the Isle of Thorns, near Haywards Heath. Over several days, students are given the opportunity to give presentations on their work, exchange ideas, and most importantly, socialise. Out of this annual event arises a collection of short papers that have come to be known as the White House Papers.

This summer, all postgraduate students at COGS were invited to submit papers of around 2000 words for inclusion in the Eighth White House Papers. The resulting collection reflects work in many diverse areas of research, such as philosophy, behavioural ecology, computer vision, linguistics, medical informatics, HCI, software design, and artificial life.

Many people have sacrificed their time and effort to organise the Isle of Thorns Workshop and the White House Papers. The editors would like to thank the secretaries, postgraduate students, and members of the faculty for their help in making the workshop and papers a success. Thanks also to Prof. Matthew Hennessy for financial support and special thanks to Jo Brook, for her continuous dedication to the organisation of the Isle of Thorns Workshop and White House Papers. Her help and extensive knowledge of the 'art' of LaTeX proved invaluable in the preparation of this collection.

Jonathan Howell
Joseph Wood
November 1995

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

From Genotype to Neural Network through Hierarchical Organisation

Guillaume Barreau
guillaum@cogs.susx.ac.uk

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

Abstract Until now Artificial Life has paid very little attention to the problems of developmental biology. After listing some of the reasons why these problems might be worth investigating we present a computational model which, when combined with a Genetic Algorithm, can evolve the architecture of Artificial Neural Networks performing the sensory-motor control of an autonomous agent. The model mirrors the hierarchical organisation of multi-cellular organisms. For each level modelled we have preserved the wealth of interactions between the elements that compose it, as far as this was computationally feasible. Preliminary results are reported and the model is compared to some related work.

1 Introduction

> After the publication of Darwin's *Origin of Species*, but before the general acceptance of Weismann's views, problems of evolution and development were inextricably bound up with one another. One consequence of Weismann's concept of the separation of the germ line and soma was to make it possible to understand genetics, and hence evolution, without understanding development. In the short run this was an immensely valuable contribution, because the problems of heredity proved to be soluble, whereas those of development apparently were not. . . . My own view is that development remains one of the most important problems of biology, and that we shall need new concepts before we can understand it. Maynard-Smith (1982)

The influence of Weissman's legacy has certainly reached deeply into Artificial Life (AL). To convince ourselves of it, it is enough to consider the omnipresence of evolution and the almost complete lack of attention paid to developmental issues in this field. It could be argued that there is also an inherent reason for that absence: the problems of biological form and development are too embedded in chemistry to interest a discipline whose aim is to discover universal principles of life which are independent of any particular medium.

Motivating the computational model that will be presented here is the conviction that some of the deep issues underlying development such as information exchange in multi-cellular organisms, irreversibility in pattern formation, hierarchical and distributed control and the relations of information and form should be brought together with more traditional evolution-related issues.

In fact, developmental issues are permeating AL, but only insiduously, through the combination of Neural Networks (NNs) and Genetic Algorithms (GAs). Some workers have seen in this alliance a novel and powerful methodology for the synthesis of information systems, such as robot controllers, difficult to design with traditional methods (Beer & Gallagher, 1991; Gruau, 1992; Cliff et al., 1993; Harvey

[*] This paper was originally presented as a poster at the fourth conference of the European Society for Philosophy and Psychology in Oxford.

et al., 1992; Nolfi et al., 1994). Others, bearing in mind that NNs and GAs preserve the most important features of nervous systems and evolution, have seen in the combination of the two, a chance to gain some understanding, of how complex and intricate structures like nervous systems can originate and be perfected by natural selection (Nolfi & Parisi, 1992; Dellaert & Beer, 1994; Miller & Cliff, 1994). In both cases, researchers have been confronted with the problem of representing the connectionist systems they want to evolve in the form most suited for manipulation by a GA. We show in the next section that this is an important problem the solution of which cannot rely on intuition.

2 Evolution and the Genotype/Phenotype Relationship

Let us examine some important consequences of the choice of a particular Genotype/Phenotype relationship for any evolutionary process.

First of all this relationship defines the boundaries of the search space accessible to evolution. Suppose a neural net is recovered from a bitstring in the following way: the string is read as an integer and an equal number of neurons is chained in a feed-forward fashion. This encoding imposes the serious constraint that only chains of neurons are obtained even though there is no redundancy in the code i.e. distinct genotypes always give rise to distinct phenotypes.

How easy it is to define the set of possible phenotypes depends essentially on how reversible the transformation is. If working backwards from phenotype to genotype is simple (as in the previous illustrative example) then the boundaries of the search space are easily determined. If on the other hand, the transformation is an irreversible one, determining whether a particular phenotype could be the product of *some* genotype is by definition impossible. As a consequence, our knowledge of what is within the reach of evolution becomes blurred. The biological relationship between living forms and their genes, clearly falls in the latter category.

The redundancy of the resulting code is a related and important characteristic. It can be vaguely defined as the number of distinct genotypes that give rise to the same phenotype. Being correlated to the frequency of neutral mutations it clearly influences the evolutionary process.

Fitness landscapes have been recognised as a useful conceptual tool for studying evolutionary dynamics (Kauffman & Levine, 1987). This landscape results from the juxtaposition of phenotypes according to an arrangement of their genotypes at genetic level based on one point mutation. Because this factor clearly spans across the phenotypic and genotypic level, it will clearly ultimately result from the relationship uniting the two.

With crossing-over, beneficial genes at different loci which have arisen independently in different individuals of the population can, at some point in the future, find themselves combined in the same individual. The chances of this combination being successful, i.e. providing an advantage over genotypes possessing only one of the two genes, is central to the efficiency of the crossing-over operation. The way genes interact to produce phenotype and subsequently fitness, is again, a direct consequence of the Genotype/Phenotype relationship at hand.

3 Artificial Development of a Neural Network

We propose here an instance of such a relationship that respects some of the hierarchical and temporally dependent features of biological development. The end product is a neural architecture capable of performing the sensory-motor coupling of a simulated robot of the type described in Cliff et al. (1993). It is in other words a number of nodes interconnected through oriented links. Some of those nodes will be committed to receiving input from a particular portion of the visual field of the robot; two output nodes will control the instantaneous speed of each of the wheels of the robot. Many nodes will be intermediary units with an arbitrary number of incoming and outgoing connections, including recurrent ones. The problem of assigning weights to those links will be ignored for the time being. It will be dealt with later, once we know what range of topologies can be achieved.

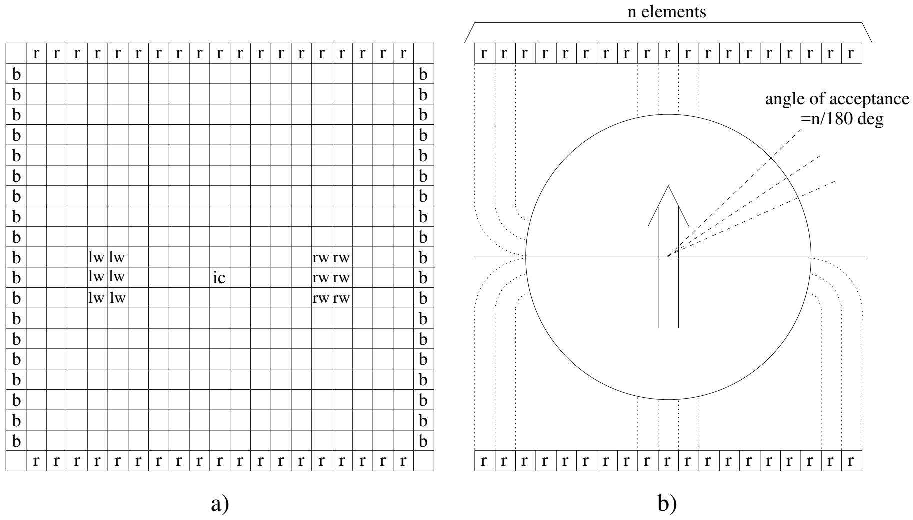

Figure 1: (a) The grid world showing the initial cell before development starts (r=retina; b=border; lw=left wheel; rw=right wheel; ic=initial cell). (b) The relationship between the *r* slots and the retina of the animat.

The spatial background

The process of simulated development takes place in a idealised space, namely a two-dimensional grid. Some of the slots that compose the grid contain static symbols. These symbols set some kind of boundary conditions to the process: the two dimensional structure of the grid is somehow isomorphic to the shape of the robot as is illustrated in Fig. 1.b. Each slot containing an r stands for a particular element of the retina of the animat. All slots containing lw stand for the neuron that will control the speed of the left wheel and similarly with the right wheel. The b symbols are inert indicators of the limits of the grid.

What is on the Grid?

In the development of most tissues, the building blocks of adult shape are the cells. In neurogenesis however, axons and dendrites of a same neuron can decouple their behaviour and function as autonomous units responding to the particular stimuli provided by their immediate surroundings. For this reason, we chose, in this system, to model the interactions of entities more alike to neurites than to neurons. Each modelled neurite is mobile but is localised at any given time in a slot of the grid. Conversely, every slot that does not contain a static element (r,lw,rw,b) can contain a maximum of one neurite.

Neurites come into existence by duplication of previously existing neurites, and all neurites that coexist on the grid are descendents of the initial cell that exists at the time development starts.

Neurite Behaviour

Every neurite belongs to a particular neuron which might or might not have other neurites on the grid. We have therefore two types of division.

The first type, which we call branching, is the creation by a neurite of another neurite belonging to the *same* neuron. It is equivalent to the tip of an axon or dendrite branching so that a new tip is created. The behaviour of the new tip is to some extent independent of that of the preexisting one.

The second type of division is more alike to biological cell division. It creates a new neuron and, to start with, a single associated neurite. In real neurogenesis, a cell that has grown dendrites or axons cannot divide anymore. In our system however a neuron which already has a rich arborescence is still capable of producing new cells.

In addition to division, every neurite has additional behaviours it can perform:

- It can die and disappear from the surface of the grid.
- It can move to a neighbouring cell provided that this cell is empty.
- It can connect to another neurite if there is one in its immediate neighbourhood. If both neurites belong to the same neuron, that logically leads to a recurrent connection.
- It can connect to a retinal element or to a motor unit if it is next to one.
- It can remain idle.

The Causes of Neurite Behaviour

Which behaviour a neurite will perform at a particular time is the result of both its genotype (which is the same for all the neurite of that net), its current neighbourhood and its history. Since all neurites that interact have the same genotype, only the two last factors are responsible for their difference in behaviour. The history of a neurite is mediated by the change in internal states which effectively provide the neurite with some kind of memory. The internal state of a neurite is the conjunction of three arrays of binary elements, each of which is characterised by the kind of events that causes it to change its state.

We call the slowest of these arrays the lineage register or L register. It is only updated at cell division and therefore carries information that is purely related to the lineage of the cell.

The second array, called the communication register or C register, is updated every time the cell comes into direct contact with any type of entity whether another neurite or a static landmark.

The third one is called the metabolic register or M register and is updated at every elementary clock cycle of the system. Because the M register changes all the time, the cell has the potential to follow a complex trajectory even though its neighbourhood remains empty.

From Genotype to Neurite Behaviour

The rules that dictate how the internal state should be updated is what is specified by the genotype. For every binary element of the registers, the genotype determines what are the factors relevant to its update and what should be the value of the update for any particular configuration of those factors. Each register is then updated at regular (M register) or irregular (L and C registers) intervals in a similar way to a Random Boolean Network (Kauffman, 1993).

4 Results and Work in Progress

This work aims at contributing to the design of autonomous agents through evolutionary techniques. One aim is to evolve a visual system capable of motion detection. This would allow interesting comparisons with the neural circuitry known to perform that function in insects (Franceschini, Pichon, & Blanes, 1992). Furthermore, a solution to this problem probably requires some periodicity in the connectivity, and this is quite a typical feature of many developmental patterns which would be interesting to see our system reproduce.

As a preliminary study, we have tested our developmental transformation with a fitness function designed to test the capacity of the system to react to a selection pressure related to the form of the network rather than its function. More precisely, the function used gave the maximum score of 140 for a network having exactly 40 neurons and an average of 3 connections each, this score decreasing linearly as you moved away from the optimum. In addition, an extra point was added for each connection with the retina. A spatialised Genetic Algorithm with asynchronous replacement of genotypes was used. The population size was 400 and the chromosomes allowed the encoding of 100 metabolic links. The probability of mutation per bit was set to 0.005.

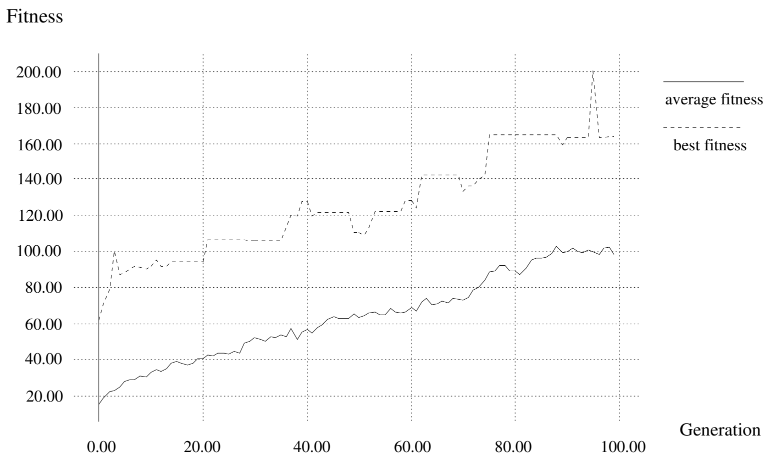

Figure 2:

The result a typical run is illustrated in Fig. 2.

We can see that the system responded immediately to the selection pressure and that it has evolved fairly rapidly to a reasonable fitness. The best networks evolved had almost exactly the desired number of nodes and connectivity and up to 60 connections with retina elements. This is an encouraging result but it is obviously too early to evaluate the practicality of this system for the evolution of Neural Networks.

5 Relation with Existing Work

Many solutions have been proposed for the encoding of Neural Networks in a form suitable for Genetic Algorithms. For a fairly complete picture, the reader can consult Kodjabachian and Meyer (1994) as well as Husbands, Harvey, Cliff, and Miller (1994). We will shortly discuss two of the existing solutions because they also have at their heart cell-like agents operating in parallel and ultimately responsible for the the final pattern.

The cells in Gruau (1992) all mature into neurons. Their defining parameters at any moment in time are a pointer to a symbol of the tree-shaped genotype and a set of connections with other cells. The cells are not in any kind of spatial relationship with each other. At any moment in time a cell executes the operation corresponding to the symbol it is pointing to and from there moves on to the next symbol. Because the tree is read from root to leaves, distinct sequences of operations can be followed by different cells so that, in fact, cells use different parts of the genotype.

(Dellaert & Beer, 1994) have proposed a more biologically minded model of the cell. Their system and ours share a number of properties. Cells occupy a definite position in space, they exchange information with those that happen to be their neighbour and they can self-replicate. Furthermore, in both cases, the relationship of a cell with its genome is a diffuse one which is not reducible to any well-defined subpart of the chromosome. This is done in both systems through the use of RBNs.

They are however some important differences between the two systems:

- the final product of development is, in our system, a network rather than a pattern of differentiated cells.
- the cells have been endowed with a more extensive behavioural repertoire, giving them for example the ability to move extensively in two dimensions.
- the Random Boolean Networks driving the cells are allowed variable numbers of inputs per node and use a special set of boolean functions.

- all Random Boolean Networks will produce a network whereas Dellaert and Beer have discarded all networks which did not have a point attractor.

We feel that the properties of development which are important for our purpose, stem from all the mechanisms which affect the way information flows in the system. It is therefore important to leave a maximum of flexibility to the kind of information exchanged but also to the way this information is distributed, exchanged and updated across the system.

References

Beer, R., & Gallagher, J. (1991). Evolving dynamical neural networks for adaptive behaviour. *Adaptive behavior*, *1*(1), 91–122.

Cliff, D., Harvey, I., & Husbands, P. (1993). Explorations in evolutionary robotics. *Adaptive behavior*, *2*(1), 73–110.

Dellaert, F., & Beer, R. (1994). Toward an evolvable model of development for autonomous agent synthesis. In Brooks, & Maes (Eds.), *Artificial Life IV*, pp. 246–257.

Franceschini, N., Pichon, J., & Blanes, C. (1992). From insect vision to robot vision. *Philosophical transactions: Biological sciences*, *337*(1281).

Gruau, F. (1992). Cellular encoding of genetic neural networks. Tech. rep. 92-21, Laboratoire d'Informatique du Parallelisme, Ecole Normale Superieure de Lyon.

Harvey, I., Husbands, P., & Cliff, D. (1992). Issues in evolutionary robotics. In Meyer, J., Roitblatt, H., & Wilson, S. (Eds.), *From animals to animats 2*. MIT Press.

Husbands, P., Harvey, I., Cliff, D., & Miller, G. (1994). The use of genetic algorithms for the development of sensory-motor control systems. In *Proceedings of the first conference: From perception to action*.

Kauffman, S. (1993). *The origins of order*. Oxford University Press.

Kauffman, S., & Levine, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of theoretical biology*, *128*(11).

Kodjabachian, J., & Meyer, J. (1994). Development, learning and evolution in animats. In *From perception to action*.

Maynard-Smith, J. (1982). *Evolution and the theory of games*. Cambridge University Press.

Miller, G., & Cliff, D. (1994). Protean behaviour in dynamic games: arguments for the co-evolution of pursuit-evasion tactics. In Cliff, D., Husbands, P., Meyer, J., & Wilson, S. (Eds.), *From animals to animats 3*. MIT Press.

Nolfi, S., Floreano, D., Miglino, O., & Mondada, F. (1994). How to evolve autonomous robots: different approaches in evolutionary robotics. In Brooks, & Maes (Eds.), *Artificial Life IV*, pp. 190–197.

Nolfi, S., & Parisi, D. (1992). Growing neural networks. In Langton, C. (Ed.), *Artificial Life III*.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

Induction Input Management

Hilan Bensusan
hilanb@cogs.susx.ac.uk

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

Abstract This brief work aims to suggest that it might be profitable to study the management operations in a training set for inductive problems in a unified way. By management operations it is meant the operations on the original set of attributes and examples that can be performed in order to increase the prospects of an accurate induced hypothesis. A classification of such operations is provided and some similarities among them are pointed out.

1 Introduction

Human inductive performance seems to profit from the ability to manage the input data. Humans can often evaluate when the data available is enough, when only a subset of data deserves consideration and when a redescription of the data would make induction easier. This sort of management work allows the induction process to concentrate on data where a sufficiently accurate hypothesis can be found in a reasonable amount of time. There seem to be a growing conviction that only by the artificial inductive systems with resources to automatically manage the input data can we expect to broaden the scope of solved induction problems.

The aim of this brief work is to suggest an outline of a unified framework to study input management in general and how it can be implemented in artificial systems. It will be assumed that input data for induction problems are given in the form of a set of examples described by means of a set of attributes. I shall restrict myself primarily to cases where all the examples are classified - instances of supervised learning. The set of examples, their respective attribute values and their classification compose a training set. Input management is therefore made by promoting any sort of profitable change in the training set.

2 Managing a training set

I shall propose a classification of the operations that can be applied to examples or attributes in a training set. They correspond to types of management operations that have been provided to a number of inductive systems. Given a set A = $\{A_1, \ldots, A_n\}$ of attributes and a set E = $\{E_1, \ldots, E_n\}$ of examples in a training set we can have the following possible sorts of operations:

- Attribute Selection: Chooses a subset of A. This is recommended whenever there are attributes that would not be useful to learn the target concept.
- Attribute Construction: Promotes a training set of A ∪ A' where A' is a set of new attributes built out of operations on subsets of the original attributes $A_1, \ldots, A_n$. The selection of operands among the original attributes is a special case of attribute selection. Attribute construction is the basic element of the so-called constructive induction.
- Attribute Destruction: Chooses a subset of A ∪ A'. This can be seen as a revision on the results of the preceding operation.

- Attribute Addition: Searches (somewhere) a set A^* of attributes and generate a training set of A ∪ A^*. This is the first non-redesciption operation for it finds new data bits and not merely a new way to represent the available data.

- Example Selection: Chooses a subset of E. An excessive number of examples can increase both the computational cost and the likelihood of overfitting.

- Example Construction: Merges examples from E and generate a set E' of examples that are added to the original examples of the training set. This is often used to group values of attributes and drives the attention of the learner to special similarities among the original examples.

- Example Destruction: Chooses a subset of E ∪ E'. Sometimes some or all the original examples should be eliminated to help the learner to find an accurate hypothesis.

- Example Addition: Searches (somewhere) a set E^* of examples to be added to the training set. This can be seen as the requirement for additional cases for induction or, alternatively, as the requirement for experimentation. This is the second non-redescription operation.

The management operations should not be seen as taking place before the induction starts. In fact, they often require some sort of information about the learning problem at hand. Most of the operations are then executed incrementally (gradual deletion of attributes, gradual construction of new ones, gradual grouping of attribute values etc.). The detection of the need of management operations, the choice of an adequate one and the search of convenient operands are activities to take place within the induction process for they should be concomitant to the search of an adequate hypothesis - the search for the best input data and the search for the best inductive hypothesis should be seen as two faces of the same search.

Since the operations take place during the search for an adequate hypothesis, they can be activated by previous learning attempts on the initial data. Management is composed of two separable routines - although they often are intertwined. The first is the detection-evaluation routine, that assess the quality of the current input data (the original input data or input data already processed by management operations). The second routine is the selection of the adequate management operation and a convenient set of operands. This two interrelated routines are the major elements of input management and only when they are possible an improvement of the input data can be generated.

In order to automatise the process, the need for input management should be detected by inspecting the original training set and the hypothesis induced from it. Common resources for detection involve the analysis of the shape of the hypothesis when considered as a surface over the instance space (see Rendell & Cho 1990), the cross-validation technique by which the accuracy of d(E) hypothesis generated by a subsets of the training examples are contrasted[1] (see Breiman et alii 1984) and multiple applications of the MDL (Minimum Description Length) principle (see Rissanen 1985) whereby we aim to detect when a generated hypothesis is too complex to be acceptable (it could, for instance, be more complex that the training set itself).

In most of the rest of this work I shall make some comments on the above listed management operations concentrating on the decision tree induction (C4.5 of Quinlan 1993).

The two non-redescription operations described might be used when a theory refinement is needed - more attributes and more examples can incrementally improve the already obtained hypothesis. Automatising attribute and example addition is hard to be done in general for they require a search in an attribute and example space - that search is the basis for the selection routine. The advantage of incremental learning, achieved by iterated example addition operations is nevertheless easier to appreciate. A small number of examples is often sufficient to learn a coarse hypothesis and sometimes a sufficient gradual refinement requires less resources and less examples than a batch learning from the entire set of examples would need. It is easy to see that a small A^* or E^* added to A or E can be responsible for a major change both in the size and in the shape of the induced tree.

[1] d(E) is the cardinality of E.

Inductive theory change should deal with the learned concepts - the currently entertained induced trees. When a new necessary example is added, a tree can be expanded in order to benefit from the new data (and eventually pruned afterwards) or a revision can be required. Revision is needed when the added example differs from a previous one only in its classification value. This can indicate one or more of the tree following situations: a) the new training set (composed by E $\cup$ E^*) is noisy; b) the target concept has a probabilistic character or c) the new training set has an insufficient number of attributes (there are, so to speak, hidden variables). In the third case the attribute addition operation module can be activated (if new attributes can be searched somewhere). In the first two situations, a new tree should be generated. This can be done by running the tree generator algorithm again. Alternatively, we can generate a new tree by replacing the values of the leafs and eventually shortening some branches. This avoids the costs of finding splitting points for the new training set and therefore can prove to be more efficient.

Other revision methods may profit from other, concomitant input management operations. This is the case of the recently proposed TGCI algorithm (Donoho & Rendell 1995). The two revision strategies considered above can achieve a considerable change both in the size and the shape of the induced tree but they are rather limited in promoting representation changes. The TGCI (for Theory Guided Constructive Induction), like MIRO for example (Drastal & Rastz 1989), makes use of the new examples to incrementally promote attribute construction and destruction - and therefore constructive induction. By this means, Donoho and Rendell expect to achieve representation flexibility - allow the revised theory to use a set of attributes different from the one used by the original one.

I turn now to the redescription management operations. Example selection is required when the number of examples are likely to make the learner induce too much structure. The windowing technique (see for instance Quinlan 1993, chap. 6) was originally designed to deal with training sets that have a number of examples too large to be handled by a bounded memory. It is also used to avoid overfitting. The initial window of examples is randomly selected and it is (randomly) incrementally increased until an acceptable degree of accuracy is reached. Since windowing is dissociated from memory considerations, a detection procedure should decide when windowing is required. Moreover, a more intelligent selection routine can be considered - the system can take an initial window of examples and an order of examples to increase it that are likely to be useful to learn the concept at hand.

Attribute selection deals with attribute irrelevance. It easy to see that, together with noise, attribute irrelevance is the most likely element to generate overfitted learned concepts. Irrelevance can be detected by a cross-validation on an increasing number of attributes. The procedure can be thought of either as similar to windowing or as a kind of pruning whereby a simpler (and supposedly more likely to be accurate in the test set) and still reasonably accurate (in the training set) concept is searched. Kohavi (1995), for instance, uses an incremental cross-validation to assess the quality of an attribute subset by the accuracy of their respective decision tables where some attributes and a default class replace a fully expanded decision tree (or set of rules) based on the whole set of attributes. A less costly alternative, without cross-validation, however, was introduced by Pfahringer (1995), using the MDL measure. In any case, the attribute subset to be found is a result of a tradeoff between simplicity and accuracy where detection and selection are indistinguishable parts of the search.

Example construction and destruction is needed when the target concept is more likely to be learnable from types of examples than from examples themselves. Grouping values of attributes is used to achieve a better ratio of training examples per attribute value and, as in the option C4.5 -s (see Quinlan 1993, chap. 7), it is made by merging relevantly similar examples in the input data. Often, but not always, the original examples need to be destroyed (example destruction). Examples construction is typically needed when the attributes values of a subset of examples are sufficiently (relevantly) similar and if they are treated separately, a too small number of examples will be available for each attribute value. When merging attribute values, the examples constructed copy the original examples concerning the values of all the other attributes. Grouping values and example construction can be profitably compared to constructing attributes (constructive induction) for in both cases the redescription of the input data that is achieved is likely to capture high-order regularities in the training set. As a clear example of this similarity, the MONK's problems (Thrun et alii 1991) can be mentioned. The MONK's are often thought of as problems to be handled successfully by some form of constructive induction - AQ17-DCI and AQ17-HCI are the early, and controversial examples. Now, although plain C4.5 performs very bad in these problems, the -s option allows a far better result (see Quinlan 1993, chap. 7).

Attribute construction and destruction are the kernel of what is called constructive induction. In constructive induction, a detection-evaluation component assess the quality of the current attributes and can be done either by somehow analysing the data alone or by making use of a selective (non-constructive) learner (as C4.5). The selection routine is the redescription part of the process - the choice of operands (set of previous attributes) and operators (operations among the selected operands) that will add new attributes to the training set. Additionally, we can provide the selection module of a destruction procedure whereby some (ground or constructed) attributes that are considered irrelevant are eliminated. There is a variety of alternative methods both for the detection-evaluation and the selection routines as there are different systems of constructive induction (see Kramer 1994 for a classification). Some systems are provided with algorithm-fixed operators - like FRINGE (Pagallo 1989, Pagallo & Haussler 1990). The hypothesis generated by a selective learner is assessed by the detection-evaluation and it is left to the selector simply the task of choosing the adequate operands on the basis on the selective learner output. In recent, more sophisticated constructive induction systems - such as CIPF2.0 (Pfahringer 1995a) a non-algorithmic fixed operator for constructing new attributes is present. A set of operators is then applied whenever they can be applied and the population of generated attributes is then assessed by the detection-evaluation module. This allows the system to select which operator is needed for each learning problem. In a system like CIPF2.0, each redescription step can involve different operators, allowing the final hypothesis to be represented with a greater richness. The system aims to minimise both the hypothesis to be generated and the amount of attribute construction required to achieve it. It takes, therefore, an approach based on the simplicity (via MDL) of both the learned hypothesis and the data redescription - it minimises both the amount of redescription and the size of the learned hypothesis. The cost to be minimised is the sum of the number of bits needed to encode the current hypothesis and the number of extra bits needed to send the examples of the training set that are not consistent with the encoded hypothesis. The formula used is the Quinlan's (1993) one - but an improvement is proposed in later works by Pfahringer - with an important addition to deal with constructed attributes. To compute the number of bits needed to be sent to a receiver that should be able to retrieve the training examples, it is supposed that the receiver is informed of the set of attributes to be used to describe the examples. CIPF2.0 computes the cost of encoding the constructions of new attributes out of the ground attributes and add this cost to the total cost to be minimised. By this means, the selection of constructors and the search for better attributes are integrated in the overall process of induction. Different hypotheses are assessed by the cost of sending them and the uncovered examples to a receiver that is informed of the original set of attributes (A) only.

3 A unified approach to input management?

Numerous aspects of input management could be further investigated from the unified perspective suggested here. It would be interesting, for example, to try to determine whether or not some types of management operations could be replaced by others without considerable loss in accuracy and efficiency. Perhaps, for instance, attribute construction can always be used instead of example construction. Another question to be considered is how efficient can be a method for detecting the need of input management in general (in contrast to detecting the need of particular input management operations).

It is clear that there are remarkable differences among the management operations mentioned here. It seems nevertheless that many features of the operations are similar - the most noticeable, perhaps, is that the techniques used for the detection-evaluation and selection routines that can be applied to one of them might also be applied to most of the others. How adequate is an unified approach depends therefore on the fruitfulness of examining these similarities.

References

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984), *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group.

Donoho, S. & Rendell, L. (1995), Rerepresenting and Restructuring Domain Theories: A Constructive Induction Approach, *Journal of Artificial Intelligence Research*, 2, pp. 411–446.

Drastal, G. & Raatz, S. (1989), Empirical Results on Learning in an Abstraction Space, *Proceedings of the 1989 IJCAI.*

Kohavi, R. (1995), The Power of Decision Tables, Paper submitted.
Paper available in: ftp://starry.stanford.edu/pub/ronnyk/tables.ps.

Kramer, S. (1994), CN2-MCI: A Two-step Method for Constructive Induction, *ML-COLT '94 Workshop on Constructive Induction and Change of Representation*.

Pagallo, G. (1989), Learning DNF by Decision Trees, in *Proceedings of the Eleventh Joint Conference on Artificial Intelligence*, Detroit, MI, US, pp. 639–644.

Pagallo, G. & Haussler, D. (1990), Boolean feature discovery in empirical learning. In *Machine Learning*, 5, pp. 71–100.

Pfahringer, B. (1995), Compression-Based Feature Subset Selection, Submitted to IJCAI95.
Paper available in: http://www.ai.univie.ac.at/cgi-bin/biblio-ora.

Pfahringer, B. (1995a), Robust Constructive Induction. In *Proceedings of the 18th German Conference on Artificial Intelligence (KI-94)*, Saarbruecken, Germany, 1994. Preliminary version available as Report TR-94–11, Austrian Research Institute for Artificial Intelligence, Vienna, 1994.

Quinlan, J. (1993), *C4,5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann

Rendell, L. & Cho, H. (1990), Empirical Learning as a Function of Concept Character, *Machine Learning*, 5, pp. 267–298.

Rissanen, C. (1985), Minimum Description Length Principle, In Koltz, S. & Johnson, N. (eds.), *Encyclopedia of Statistical Sciences*, Wiley, pp. 523–527.

Thrun, S., Bala, J., Bloendorn, E., Bratko, I., Cestnik, B., Cheng, J., De Jong, K., Dẑeroski, S., Fahlman, S., Fisher, D., Hamann, R., Kaufman, K., Keller, S., Kononenko, I., Kreuziger, J., Michalski, R., Mitchell, T., Pachovicz, P., Reich, Y., Vafaie, H., Van de Welde, W., Wentzel, W., Wnek, J. & Zhang, J. (1991), The Monk's Problems: a Performance Comparison of Different Learning Algorithms, *Technical Report CMU-CS-91–197*, Computer Science Department, Carnegie Mellon University.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

Automatic Debugging of Multiple-Function Programs

Changiz Delara
changiz@cogs.susx.ac.uk

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

Abstract This paper reports on an intelligent debugging system (based on the plan calculus formalism (Rich, 1981; Rich & Waters, 1990)) for automatically detecting and correcting semantic errors in novice student programs written in ML. Its computer implementation is called EMILY. We explain the overall structure of the system and give an overview of the adopted technique for debugging single function programs. We also discuss the approach that used to debug programs with multiple functions. Finally, future research work is pointed out.

1 Introduction

This paper reports on an intelligent debugging system (based on the plan calculus formalism (Rich, 1981; Rich & Waters, 1990)) for automatically detecting and correcting semantic errors in novice student programs written in ML. Its computer implementation is called EMILY. We explain the overall structure of the system and give an overview of the adopted technique for debugging single function programs. We also discuss the approach that used to debug programs with multiple functions. Finally, future research work is pointed out.

2 The Overall Structure of EMILY

EMILY consists of three modules and two knowledge bases. These are the translation module (translator), the program understanding module (the chart parser), the bug detection module (debugger), the plan library, and the reference library. Figure 1 shows the overall structure of the system. The translation module is responsible for translating a student ML program into its equivalent surface plan[1] representation.

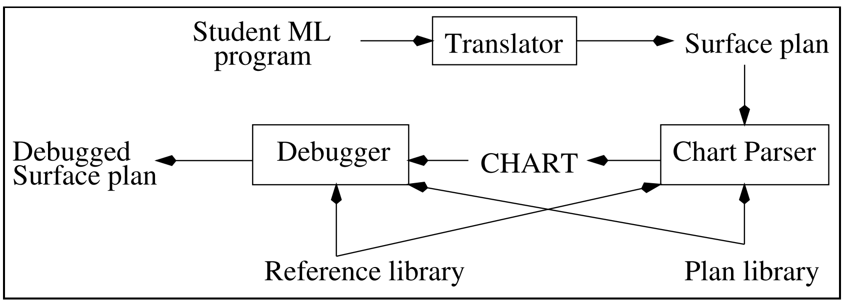

Figure 1: The overall structure of EMILY

[1]Representation of a program in terms of its primitive operations and data and control flow is called a surface plan.

The program understanding module accepts the surface plan of the student program and parses it against the plans in the plan library. It then generates a high-level semantic description of the program in terms of one plan or a small set of plans from the plan library. All assertions in this process are stored in a database called *"chart"* (Lutz, 1992). The debugger needs three types of information in order to start the debugging namely, a high-level goal specification (obtained from the reference library), the chart, and the plan library. Once equipped with this knowledge, the debugger attempts to locate the bug and repair it at the surface plan level.

3 An Overview of the Debugging of Single Function Programs

In order to debug a program EMILY must have enough programming knowledge[2] to enable it to understand the program. That is, it must be able to generate semantic descriptions of the program. This constitutes the chart. Having the the plan library and the chart, it then needs a high-level goal[3] specification of what the student program should achieve. When it gets this knowledge, it commences the debugging of the program.

EMILY checks whether the program is correct or not. If it finds an exact match to the high-level goal specification in the chart, it then postulates that the program is correct, prints an appropriate message and terminates the process.

Otherwise it checks whether the student program has solved a similar goal to the high-level goal. This may happen when (say) the high-level goal is sorting a list in ascending order, but the program has implemented the list in descending order. In such a case, the debugger finds the place where such discrepancy has occurred and removes those plans (operations). It then creates new correct plans and inserts the changes into the chart by means of the chart parser.

If neither of the above is the case, then the debugger examines the chart for the closest partial match for the goal (we refer to it as CPG). When such a plan is found, it then takes each un-instantiated sub-plan in turn, propagating any known information from the CPG (e.g., known controlling condition). It then sets it as a current active goal (we refer to it as CAG) and continues the debugging of the program.

When there is no partial match for the goal, the debugger uses the plan hierarchy for the goal (from the plan library) and descends the plan hierarchy (in top-down and depth-first manner) to get the children plans for the goal. Next, it examines the chart for them. If there are such partial plans, it chooses the *'best'*[4] one and sets it as the CAG.

If there is no such partial plan for the CAG, then it descends the plan hierarchy as discussed above until it finds one, or it reaches the bottom level plan (a plan whose component (sub-plans) are all primitive plans (surface level plans)). In this case the debugger creates a corresponding new plan for it (where all of its components are instantiated) and inserts it into the chart by invoking the chart parser. Note that this process recursively continues until the program is debugged.

Note also that whenever the debugger locates a semantic bug it repairs it as well. Note that the repair is done at the surface plan level, and that EMILY does not concern itself with the pedagogical aspect of such repairs. That is, presenting the repairs to students and tutoring them is left to the tutoring module of an ITS that will employ EMILY as its domain expert module. For more explanations see (Delara, 1994).

4 Gathering Supportive Debugging Information

In order to debug any program EMILY first checks if it is correct in terms of the high-level goal specification of the program. If it is correct then it prints an appropriate message and terminates the process, otherwise it prepares some supporting information before plunging into the debugging process. The sup-

[2]We have introduced sufficient domain specific and general purpose plans to the previous version of the plan library (which had about two hundred general purpose plans).

[3]The goal is expressed in the form of a plan from the plan library.

[4]The best partial plan means the plan which is related to the CAG and has maximum instantiated sub-plan(s), etc.

porting information can include the identification of buggy functions, and the ascertaining of functions' task in terms of plan(s) (as far as possible). That is, specifying what function has achieved which plan. Let us explain the process for gathering the supporting information first.

4.1 Identifying Buggy Functions

In order to find buggy function(s), EMILY gets the high-level goal specification from the reference library and examines the chart for the closest plan to the goal (i.e., CPG). When such a plan is found, it expands (recursively) all of its instantiated sub-plans, and any other plan in its plan hierarchy, to surface-plan level. We call such plans *already-covered* plans. Next, it identifies those functions that have been covered by the *already-covered* plans. This means, EMILY maps the *already-covered* plans to the surface plan of each defined function in the student program. A function is said to be covered when all elements of its surface plan exist in the *already-covered* plans. This mapping reveals which functions are covered by the instantiated components of the CPG. All the uncovered functions are considered to be buggy. If a function contains a "superfluous error" (Johnson, 1986) then this approach may not work and such a function is flagged as erroneous. We have not attempted to consider "stylistic error" (Murray, 1988) in a function in this process. Detecting the faulty functions is considered part of the debugging process, and such information can be rendered to the student or to the tutoring system.

4.2 Specifying Task of Functions

In order to identify the task of each function (in terms of a fully instantiated plan or a small set of fully instantiated plans) EMILY finds the closest partial plan to the given high-level goal specification (i.e., CPG) and it also takes the function name[5]. Next, it sets that function as a current active function (we refer to it as CAF), and it takes each known sub-plan of the CPG in turn and checks if it is implemented by a function other than the CAF. If it has been implemented by a function other than the CAF then it checks whether the plan does cover the function that has implemented it or not. If the plan covers the function that implemented it, then EMILY notes the fact, otherwise it assumes that the plan is implemented by the function but that it does not cover the function.

If a sub-plan of the CPG has not been implemented by any other function then EMILY checks its sub-plan(s) to find out which of them are implemented by any other function than the CAF. This process continues recursively for all the sub-plans of the CPG (and their sub-plans) in the plan hierarchy.

This is similar to what an expert programmer usually does when debugging multiple-function programs. That is, he/she first tries to understand each function (i.e., to recognize some plans and their relationships - this is similar to the task of the chart parser)) and then to come up with a high-level semantic description for each function involved in the program. He/she then uses these assertions in the debugging process wherever seems appropriate. Clearly, any violation to these assertions is considered as a sign of a bug. Needless to say, this is not the whole story of expert debugging rather, this is one of the sources of information that an expert programmer relies on.

5 Debugging Approach for Multiple-Function Programs

Having the current active goal and the current active function, EMILY starts debugging the program in the same way as it does for program with a single-function program or for a program with no function at all. For detail discussion on debugging techniques of such programs see section 3 and (Delara, 1994). Since the program has multiple functions, therefore EMILY must be cautious during the debugging. This means that EMILY must make sure that the CAG that it is debugging the program for should be implemented by the CAF or it should be implemented by another function.

Bugs in a multiple-function program may occur in various different ways. For example, a called function is correct and the place where the call is made is correct, but the argument(s) of the call are wrong

[5] Each plan (fully or partially instantiated plan) carries the name of function that it is implemented in.

or the wrong function is called by mistake (we assume the wrong called function is called somewhere else in the program and is correct). In this paper we only discuss these two cases mentioned. See the program at the end of this paper which is an example where the *is_vowel* function itself is correct, and also is correctly called somewhere else (i.e., in the *fem_def* function), but it is incorrectly called in the function *masc_def*: the first call is correct but the second call has the wrong argument.

5.1 Identifying a Call to a Function

During the debugging process, whenever EMILY tries to debug the current active function (CAF) for the current active goal (CAG), it first checks whether the CAG is actually implemented by the CAF or whether it is implemented by another function due to a call. To ascertain this, it examines the output of the CAG to see if it is instantiated. If it is, then it compares the output of CAG with the output of all the calls that have been made in the program. If it finds such a call then it postulates that the CAG ought to be implemented by the called function. As such, it finds the function name associated with this call and sets it as a CAF. In order to find such a function name, EMILY uses the information that the translation module has provided.

If the output of the current active goal (CAG) is not known then EMILY takes its input(s) and checks if there is a call whose argument is equal to the input or is a member of it. If there is one such call, then this means that the CAG ought to be implemented by the called function, and EMILY finds the associated name and sets it as current active function. If there are several such calls with the same input, then this means that there are several different calls to different functions with the same input argument(s). Next, EMILY should decide which of these functions actually implements the CAG. To do this, it finds the closest plan to the CAG by descending the plan hierarchy. When it finds such a plan, then it gets its function name from the plan and sets it as the CAF. If none of the above is the case then EMILY assumes the CAG ought to be implemented by the CAF that has been set prior to this analysis. In brief, for each un-instantiated sub-plan of a CAG, EMILY must specify whether it is implemented by current active function or by another function due to a call. This ascertains the CAF.

Having the current active goal (CAG) and current active function (CAF), EMILY recursively continues the debugging process until the high-level goal is achieved during the debugging process: this implies that the program is debugged. Note that EMILY's approach is similar to that of an expert programmer when debugging a program. That is, he/she first analyses the whole program as far as possible and then tries to debug the places where a call is made (calling function), and then zooms in to debug the called function[6].

5.2 Identifying a Call with the Wrong Argument

When EMILY realizes that a current active goal (CAG) ought to be implemented by a function other than the current active function(CAF), then it checks whether the task of that function is known or not. It does this by inspecting the information that has been gathered about functions' task. If the task of the function is known and is the same[7] as the CAG, then it examines that whether that function has been covered or not. If the CAG does not cover the corresponding function then EMILY gets the function name and sets it as the CAF and continues the debugging process. This happens when a function is partially correct, that is, the function is correct for some cases and probably does not work for other cases in its implementation.

If the current active goal (CAG) does cover the function then this implies that the called function is correct and that the bug is in the argument of the call. In this case EMILY gets the corresponding call and removes all plans (fully or partially instantiated) from the chart that have been introduced by this call. It then generates a new call to that function with the correct argument, while carrying any other information from the previous call.

[6]We do not claim this as a cognitive fact and it is a separate research issue.

[7]By ' same' we mean that both plans have the same structure but differ in input, output, etc.

5.3 The Activation of the New Call

When a new call to a function is generated then EMILY transfers this call (in the format of the plan calculus) as an input graph for the chart parser to parse. This entails the called function to be in-line expanded and its surface plan is parsed. Recall that whenever the chart parser parses any elements of the input graph (which is in from of surface plan), it checks whether it is a call to another function or not. If this is the case, then the parser takes the surface plan of the called function and parses that. After the chart parser has finished, then the current active goal (which EMILY was debugging the program for) will be generated, and left in the chart. If this results in the generation of the high-level goal that we started the debugging with, then the process recursively unwinds and the debugging process terminates, otherwise EMILY continues the debugging process recursively.

5.4 Dealing with a Wrong Function Call

If the task of the called function is different from the current active goal (CAG) and the plan representing its task has covered the function, then this means that the current call to the function is wrong - the student called a different function. In this case EMILY gets the corresponding call, and removes from the chart all plans (fully or partially instantiated) that have been introduced by this call. Then it compares the CAG with the known tasks of functions to find out which of them implements the same task as the CAG. If it finds such a function then it generates a new call to that function (which should be called). Having done that, the rest of process will continue as described above. It is worth stressing that when EMILY locates the bug it repairs it as well. The repair is done at the surface-plan level and EMILY does not concern itself with the pedagogical aspect of such repairs. That is, presenting the repairs to students and tutoring them is left to the tutoring module of an ITS that will employ EMILY as its domain expert module.

6 Experimenting with EMILY

We have been testing EMILY on real student ML programs. In order to do this we have taken a corpus of student programs. These programs were written to solve the problem of adding a definite article to an Italian noun. That is, the task is to take a given Italian noun, determine its gender, and add an appropriate definite article to the front of the noun. There are different rules for specifying each type of gender. There are also exception nouns to these rules, and these are provided for the students. These programs are written by students who took an ML functional programming course in autumn 1994 as their first assignment out of two assignments. Each program contains at least eight functions. An example program is shown at the end of this paper. So far, EMILY is able to understand several programs from the corpus of student programs, and is able to find the bug and fix it in the program shown below. This program is an example where one function is called more than once in another function, where the first call is correct whereas the second call is wrong (i.e., it is called with the wrong argument). More specifically, the *'is_vowel'* function (which is correct) is called twice in the function *'masc_def'*. The first call is correct while the argument to the second call is wrong.

7 Future Work

We intend to generalize our approach of debugging programs with multiple functions. That is, we are trying to formulate a theory of debugging of multiple function programs based on the plan recognition technique.

8 A sample program

```
val masc_fem_exc_list = [
("ambiente",      true),  ("mano",         false),
("animale",       true),  ("bestiame",     true),
("piazzale",      true),  ("brioche",      false),
("comunista",     true),  ("sale",         true),
("sole",          true),  ("totale",       true),
("carne",         false), ("chiave",       false),
("mare",          true),  ("radio",        false),
("mese",          true),  ("pane",         true),
("nome",          true),  ("turista",      true),
("paese",         true),  ("fine",         false),
("legge",         false), ("ponte",        true),
("piede",         true),  ("camice",       true),
("moto",          false), ("automobile",   false),
("biro",          false), ("alce",         true),
("programma",     true),  ("crisi",        false),
("stazione",      false) ];

fun is_vowel char = member char (explode "aeiou");

fun  fem_def string = if is_vowel(hd(explode string)) then
                           "l'"^string
                     else "la "^string;

fun masc_def string = if is_vowel(hd(explode string)) then
                           "l'"^string
                        else
                      if "s" = hd(explode string) andalso
                            not(is_vowel string) then
                                            "lo "^string
                      else if "z" = hd(explode string) then
                            "lo "^string
                      else "il "^string;

exception Unknown_gender
fun sgender x = case last(explode x) of "o" => true
                | "a" => false

                | _   => raise Unknown_gender;

exception Unknown_word
fun except (word,x) = if (mem x (word,true)) then true
                   else if (mem x (word,false)) then false
                   else raise Unknown_word;

fun ggender (noun,excptlist)
    = except(noun,excptlist) handle ? => sgender noun;
```

```
fun gender  noun = ggender (noun,masc_fem_exc_list);

fun singdef noun = if gender(noun) then masc_def(noun)
                                        else fem_def(noun);

singdef "banca";
```

9 Summary

In this paper we sketched the overall structure of our intelligent debugging system for student ML programs. We briefly discussed its debugging approach for single-function and multiple-function programs. We discussed how EMILY detects a call with a wrong argument and a wrong call to a function and how it fixes such bugs.

We are experimenting with EMILY's capabilities on real students' ML programs. In this regard we have included a sample of such programs, where each program consists of at least eight functions. We delineated the future work, where we are going to generalize the adopted debugging approach of EMILY to deal with other cases which may occur when debugging multiple function programs.

Acknowledgements

I am grateful to my academic supervisor Dr. Rudi Lutz for his supervision and willingness to discuss the material presented here. I appreciate the encouragement and comments that I have received from the organizing committee of the Isle of Thorns. Thanks also to Susi Ross for correcting my English and her comments. This research has been fully supported by the Ministry of Culture and Higher Education of the Government of Islamic Republic of IRAN.

References

Delara, C. (1994). Towards an intelligent debugging system for ml programming language.. In de Bourcier, P., Lemmen, R., & Thompson, A. (Eds.), *The Seventh White House Papers*, Vol. CSRP 350, pp. 60–68. University of Sussex, UK.

Johnson, W. L. (1986). *Intension-Based Diagnosis of Novice Programming Errors*. Morgan Kaufmann Pub. Inc., USA.

Lutz, R. (1992). *Towards an Intelligent Debugging System for Pascal Programs: On the Theory and Algorithms of Plan Recognition in Rich's Plan Calculus*. Ph.d. thesis, The Open University, Milton Keynes, UK.

Murray, W. R. (1988). *Automated Program Debugging for Intelligent Tutoring Systems*. Morgan Kaufman, San Mateo, USA.

Rich, C. (1981). *Inspection Methods in Programming*. technical report no. 604, Artificial Intelligence Laboratory, MIT.

Rich, C., & Waters, R. C. (1990). *The Programmer's Apprentice*. ACM Press, Baltimore MD.,USA.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

Cascade Correlation as a model of Representational Redescription

J K. Brook
joannab@cogs.susx.ac.uk

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

Abstract This paper relates to a talk given at the IOT 95 Postgraduate Workshop. It introduces the Representational Redescription Hypothesis — a position on learning and cognitive development, then motivates briefly the suitability of connectionism as a modelling framework. The remainder of the paper focuses on a particular connectionist architecture — cascade-correlation — which seems particularly promising as a model. As the speculative nature of the hypothesis leaves many aspects of the possible underlying mechanism open I will argue that although this makes it difficult to make firm claims about mechanistic detail, a model such as cascade-correlation may allow us to investigate such issues.

1 The Representational Redescription Hypothesis

The Representational Redescription Hypothesis (RRH) (Karmiloff-Smith, 1992) is an attempt to account for certain qualitative phenomena to be observed in development and in child- and adult learning, in particular the progression from knowing how to do something (procedural knowledge) to being able to reflect upon that knowledge, discuss it and manipulate it. What is important is not that performance at a task actually changes but rather that the status of the knowledge to the learner changes, i.e., the form in which it is understood and its integration into the rest of the learner's knowledge. Indeed overall task performance may actually worsen symptomatically as the roles of knowledge change and conflict.

In attempting to explain this progression, the hypothesis puts forward a series of *phases* during each of which knowledge is thought of as being represented and stored in different formats. Each of these is more explicit than the last (at the highest level this is linked with verbalisability) and allows progressively more flexibility in its use, whether verbal or non-verbal (e.g., in facilitating further learning).

The three phases are known as **I** (implicit) at which knowledge is procedurally represented and unavailable outside the original input–output mapping, **E1**, an intermediate level of the so-called 'explicitation' process, at which knowledge is more accessible than at the first phase, but still not verbalisable, and **E2/3**, conscious (and possibly verbalisable) explicit knowledge. Transition between these formats is hypothesised to involve the reiterative *redescription* of previous representations, which while not specified in detail, seems to 'reduce' the knowledge by discarding some of the original detail. It is also hypothesised that the generation of new formats is conservative or redundant — rather than each new format supplanting the last, representations form a hierarchy of levels at which the same knowledge is differently re-represented.

An important aspect of the hypothesis is the emphasis it places on endogenous (or internally driven) change; although learning happens with respect to external influences, representations are assumed to change 'off-line' after (at least some) initial competence at the task has been achieved, rather than in response to external pressures, such as the need to improve task performance. Rather than being provoked to improve by the presence of instability, the system is driven to reappropriate already stable states.

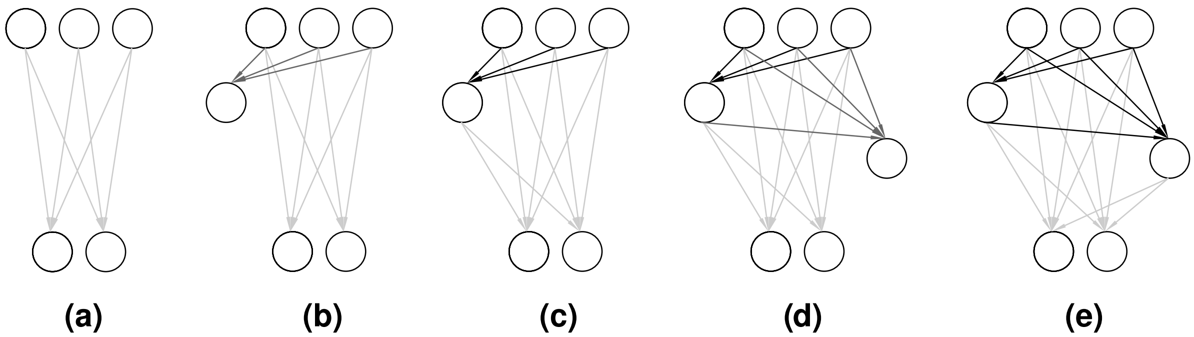

Figure 1: Learning phases in cascade-correlation over the course of two unit-recruitments. Arrows in light grey indicate connections which remain trainable using error-driven learning, those in darker grey indicate connections which are trained before installation using correlation-driven learning, and those in black indicate connections whose weights have been frozen. Input units are shown at the top. For clarity only one candidate unit is shown.

An informal example of this progression presented by Karmiloff-Smith (1992) is that of learning to play a musical piece. First one must learn to reach the initial mastery of having assembled notes and phrases into a continuous piece. Playing at this point is competent but relatively inflexible — maybe the volume of the whole piece may be adjusted but little else. The RRH has it that with time, redescription acts to increase the flexibility and accessibility of the knowledge and its components. This manifests itself in effects such as the ability to begin playing at arbitrary points during a piece, to add subtle emphases, and to improvise upon it.

1.1 The appeal of connectionism for modelling RR

The link between the hypothesis and connectionism is prompted by the observation (Karmiloff-Smith, 1992) of an apparent similarity between the opaqueness and embeddedness of procedural knowledge at the initial level and the kind of knowledge captured by a trained network. But as Clark and Karmiloff-Smith (1991) point out, standard error-driven networks do not spontaneously go beyond this success to further link, systematise and redeploy their knowledge as the RRH demands.

In response to this point, several (existing) schemes for augmenting standard networks have been proposed in the literature as models for RR (e.g., Clark and Karmiloff-Smith (1991) although see also Plunkett (1993)). These have generally involved structural manipulations on networks such as copying whole or partial networks during or after training and using various means to capture the idea of reduced re-representations.

Although a detailed survey of these proposals is beyond the scope of this paper, I will focus here on one scheme which brings out the strengths of a certain class of extensions to basic connectionist models — resource-phasing schemes — and also use it as a focus to discuss examples of issues which I believe must still be resolved in any attempt to interpret a connectionist model as capturing RR.

2 The Cascade-Correlation Architecture

The cascade-correlation architecture (Fahlman & Lebiere, 1990) (Figures 1(a)–(e)) is a multilayer supervised connectionist learning scheme which differs from the standard backpropagation model, for instance, in that although it has (multiple) hidden layers, the number of hidden units is not predetermined. Instead these are recruited as necessary to the progressive reduction of error. The net begins with only the user-specified input and output layers (Figure 1(a)) and tries to learn the task using error-driven learning. If this fails it then enters an off-line correlation-driven recruitment phase. After a candidacy phase (Figure 1(b)) in which only the incoming connections (from the inputs and any previous hidden units) to a pool of new hidden units are trained so as to maximise the correlation between their activations and the sum of the errors at the output layer, the best of these is installed in a separate layer and connected to the output units. On installation, the weights on the connections to the new hidden unit are frozen (Figures 1 (c) and (e)) and the input–output and hidden–output connections are trained to readjust overall performance. This process is repeated until either error has dropped to an acceptable level or any of various user-determined limits have been reached.

Each learning phase, whether error-driven (Figures 1 (a), (c) and (e)) or correlation-based (Figures 1 (b) and (d)), continues until training reaches 'stagnation', i.e., proportional improvements have been comparatively small over a (user-determined) number of epochs.

This architecture has been used by Shultz and his collaborators (Mareschal and Shultz (1993), Shultz and Schmidt (1991), Shultz, Buckingham, and Oshima-Takane (1994)) to construct developmental models involving the exhibition of a series of qualitative behavioural changes on tasks such as seriation, the balance-scale and the correct usage of personal pronouns respectively.

3 Cascade-correlation as a model of redescription

Cascade-correlation has been proposed (Brook (1993), Shultz (1994)) as a possible connectionist model of representational redescription. The main reasons given include the fact that it is an example of an incremental learning scheme, making it a suitable candidate for developmental modelling in general, that in unit-recruitment it incorporates a natural mechanism for supporting qualitative change, and more specifically that the changes in performance it passes through are both conservative and hierarchical, and involve alternations between phases of differently focused types of learning, transitions between which are prompted by the net's achievement of a stable state. I will consider each of these aspects in turn.

3.1 Incremental learning and developmental modelling

Incremental learning is a broad term which refers in connectionist theory to the idea that learning a task is best (or only) achieved through the 'staging' of learning, usually through manipulation of resources (or *resource phasing*). These resources may either be internal to the network, such as number of internal nodes or length of attentional window in recurrent nets, or external, such as ordering or constituency of training set. Resource-phasing works to facilitate the learning of complex tasks by making use of the fact that the initial inability of the network to deal with complexities, such as long-range dependencies for instance, can allow it to focus on the simpler aspects of the problem. And, as Clark (1993) puts it, in some cases these simpler aspects of the problem may happen to act as building blocks on which subsequent, more complete, solutions may depend. Incremental learning appeals to modellers more interested in capturing the path or trajectory through the process of learning of a task than in eventual successful performance in itself, since resource manipulation allows the trajectory through the learning process to be shaped. The hidden-unit-recruitment strategy of cascade-correlation is an example of resource-phasing, each step serving to increase the power of an initially very limited network.

3.2 Conservative changes in qualitative performance

Cascade-correlation preserves both previous structure and (partial) solutions in the frozen incoming weights to the hidden units. This gives the network the potential to return to previous solution states at least over the immediately succeeding unit recruitments. Shultz and Schmidt (1991) report that in their model of the balance-scale (torque-difference) task, around the time of a qualitative change in behaviour the network tended to go through a period where it would alternate between strategies.

The cascade architecture also resembles the hierarchical nature of the representations proposed by the RRH. The cascade architecture means that any new learning which takes place must happen with respect to all earlier, frozen, hidden structure.

3.3 Incorporation of alternating learning modes

Central to the RR model is the idea that learning within phases is driven by the need to make quantitative performance improvements, while the (redescriptive) process which brings about phase transitions is not directly error-driven and acts to bring about greater flexibility and explicitness. Cascade-correlation also incorporates two alternating learning modes, and like RR, one is error-driven and acts directly to try and improve performance while the other is indirect, occurs off-line, and is associated with biasing a (microscopic) quantitative change in processing power in the form of the recruitment of a new one-unit hidden layer. Each learning-mode change is also prompted by the net's having reached a stable state in the previous mode, which can be seen as corresponding to the stable states of (at least partial) task-mastery required for redescription in the RRH.

4 Interpretational Issues

The RRH is put forward as 'a framework — rather than a precise theory — for exploring possible generalities in developmental change across a range of domains.' (Karmiloff-Smith, 1994), and makes no detailed commitment to any possible mechanisms for redescription itself. As a consequence of this, despite the general fits to the hypothesis provided by models such as cascade-correlation, some issues of interpretation and mechanistic detail remain. In the remainder of this paper I will discuss two of these and suggest that the cascade-correlation architecture might provide us with a framework for exploring some of the open questions which remain.

4.1 Causes of representational change

There is some debate (see, for instance, Scutt and O'Hara (1993)) as to whether the RRH is justified in the strong emphasis it puts on representational change which occurs 'spontaneously' and 'off-line' to error-driven learning. Karmiloff-Smith responds that the model does not actually exclude externally driven representational change but (in contrast with many preceding theories) rather emphasises that which is internally driven, but clearly it is still difficult to factor out such external influences in natural learners.

A similar issue arises in the extent to which explicit mechanism or intervention is necessary to capture RR effects in networks. Plunkett (1993) argues that the drive towards systematicity and the corresponding representational reorganisation that attends it in standard backpropagation could already be said to be capturing the RR model in some sense. This gradual, parallel and emergent idea of RR effects contrasts with proposals such as that originally put forward in Clark and Karmiloff-Smith (1991) in which trained networks are copied and 'skeletonised' (Mozer & Smolensky, 1989) in an attempt to realise explicitly the idea of creating new and reduced representations.

The cascade-correlation architecture incorporates both standard error-driven learning as well as structural change. It thus includes explicit 'intervention' (in a sense), but in a manner which is relatively gradual, microscopic, and is interleaved with on-line learning. I would argue that it occupies an intermediate position on a continuum of degrees of explicit intervention, and in parameters controlling the extent of both externally and internally focused learning can be seen as acting as a tool for exploring the issue of the role of explicit intervention in shaping trajectories and giving rise to accessible representational formats.

4.2 Granularity of change

Shultz (1994) proposes that in interpreting cascade-correlation as a model of the RRH, the patterns of learning in each can be related directly, such that the first error-driven training phase in cascade-correlation corresponds to the within-phase learning leading to the the I-level, the subsequent correlation-driven training phase to the intermediate E1-level, and the next error-driven phase to the final explicit (E2/3) level. While, as Shultz argues, this correspondence does capture the way in which late-occurring errors appear as the learner (human or network) tries to reconcile new mappings and representations with old, preserved, patterns after a phase-change or change in processing power, there are several problems with this interpretation of cascade-correlation as a model of RR. Firstly there is an implication that the whole three-phase progress of RR can routinely be captured by a network which recruits only a single hidden unit and Shultz also has nothing to say about the interpretation of multiple recruitments for RR.

Another problem arises in that RR is assumed to involve the reiterative action of a redescriptive process. Although this process is constrained in some ways by 'the contents and level of explicitation of representations' (Karmiloff-Smith (1992), p. 25), in Shultz's reading, while the same process (error-driven learning) acts at the first and final phases, the intermediate phase involves a different process (correlation-driven learning), and it is not clear that this corresponds without further qualification to the idea of a common RR process as presented by Karmiloff-Smith.

As a consequence of the parallel drawn between the first phase of RR and standard connectionism discussed in section 1.1 above, it also seems that the proper (or at least initial) role of connectionism in attempts to model the RRH lies in trying to capture the initial implicit–explicit transition and the attendant increase in transferability and manipulability of knowledge, rather than in any claims to try and capture the final conscious, verbalisable level, or in any detail the whole (three-phase) RR model. On this basis and in the light of the results of Mareschal and Shultz (1993) and Shultz and Schmidt (1991) concerning the contribution of individual unit-recruitments to overall qualitative change, the most plausible level of granularity at which to interpret the structural change in cascade-correlation seems to be at the micro-level of a possible RR mechanism, in which multiple instances of the RR process accumulate to give larger qualitative changes.

5 Conclusion

In this paper I have introduced the Representational Redescription Hypothesis and a generative resource-phased connectionist architecture — cascade-correlation — which seems to provide a promising modelling framework. I have gone on to argue that despite the absence of detailed constraints from the RRH itself or a consensus amongst commentators concerning possible underlying mechanism, that cascade-correlation can provide a framework for exploring some of the issues raised by these debates.

Acknowledgements

I would like to thank Stephen Eglen for his comments on a draft of this paper.

References

Brook, J. K. (1993). Developmental connectionism and representational redescription. In Brook, J. K., & Arvanitis, T. N. (Eds.), *The Sixth White House Papers: Graduate Research in the Cognitive and Computing Sciences at Sussex*, CSRP 300.

Clark, A. (1993). Representational trajectories in connectionist learning. CSRP 292, School Of Cognitive and Computing Sciences, University Of Sussex, Brighton, UK.

Clark, A., & Karmiloff-Smith, A. (1991). The cognizer's innards: A psychological and philosophical perspective on the development of thought. CSRP 193, School Of Cognitive and Computing Sciences, University Of Sussex, Brighton, UK.

Fahlman, S. E., & Lebiere, C. (1990). The cascade-correlation learning architecture. CMU-CS-90-100, School Of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Karmiloff-Smith, A. (1992). *Beyond Modularity: A developmental perspective on cognitive science*. MIT Press, Cambridge, MA.

Karmiloff-Smith, A. (1994). Transforming a partially structured brain into a creative mind. *Behavioral and Brain Sciences*, *17*(4), 732–745.

Mareschal, D., & Shultz, T. R. (1993). A connectionist model of the development of seriation. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society* Hillsdale, NJ. Erlbaum.

Mozer, M., & Smolensky, P. (1989). Skeletonization: A technique for trimming the fat from a network via relevance assessment. In Touretzky, D. S. (Ed.), *Advances in Neural Information Processing Systems 1* San Mateo, CA.

Plunkett, K. (1993). Making nets work hard. *Mind and Language*, *8*(4).

Scutt, T., & O'Hara, K. (1993). 3, 2, 1 . . . we have cognition. *Mind and Language*, *8*(4).

Shultz, T. R. (1994). The challenge of representational redescription. *Behavioral and Brain Sciences*, *17*(4), 728–729.

Shultz, T. R., Buckingham, D., & Oshima-Takane, Y. (1994). A connectionist model of the learning of personal pronouns in english. In Hanson, S. J., Petsche, T., Kearns, M., & Rivest, R. L. (Eds.), *Computational learning theory and natural systems, Vol 2: Intersection between theory and experiment*, pp. 347–362 Cambridge, MA.

Shultz, T. R., & Schmidt, W. C. (1991). A cascade-correlation model of balance scale phenomena. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pp. 635–640 Hillsdale, NJ. Erlbaum.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

An Application of Artificial Intelligence Techniques to a Consumer Software Product

Ian Cullimore
ianc@cogs.susx.ac.uk

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

Abstract An implementation is discussed where the application of Artificial Intelligence techniques leads to improved performance in a consumer PC software product, with benefit to both the end-user and the development company.

1 Introduction

It is all too easy for Cognitive & Computing Science students and researchers to forget that there may be real-life application for their endeavours; perhaps one of the few fortunate side-effects of my being a part-time PhD researcher working in the PC computing industry for the remainder of my working day is that I have been able to inject some of my indirect research and learning into some of the commercial products in which I have been involved. In this case I was faced with the problem of easing the installation issues for end-users when attempting to install software to drive the new popular PCMCIA PC Card (Association, 1992) type of Ethernet and modem cards in portable laptop and notebook computers.

2 The Problem of Installation

PCMCIA cards (now more properly known as ”PC Cards”) have become very popular with portable notebook computer manufacturers and end-users alike; although originally intended as a new style of solid state memory storage, their acceptance really came as peripheral devices such as modems, LAN cards, and rotating disks became popular. Even though the PCMCIA organisation set up a technical committee and software subcommittees (in which I was involved) to architect software abstraction layers (known as Socket Services and Card Services (PCM, 1992)) to ease the problem of software drivers and applications interacting with PC Cards, the legacy nature of the PC architecture (Hogan, 1988; Norton, 1985) from its inception in the early 1980s by IBM has lead to many problems when trying to install software drivers for such cards.

Typically a device such as a modem or Ethernet card requires resources such as an available hardware interrupt (IRQ), and an open memory and I/O address range. Unfortunately the original IBM PC architecture allowed for little standardisation for the allocation of such resources (people in the universities, generally using the architecturally purer Apple Macintosh, may not be aware of such problems), it being done in an ad hoc manner by the end-user, in a trial and error style using DIP switches on the plug-in hardware boards and modifying installation parameters for the software drivers. The problem of resource allocation has been compounded with Microsoft’s Windows architecture (Schulman, Maxey, & Pietrek, 1992) often sitting on top of the DOS operating system and the underlying PC hardware architecture.

3 An Example of an Ethernet PC Card

Until recently, an end-user (whether novice or expert) would have to attempt to install a driver for an Ethernet PC Card in the following type of manner, in the case of, say, installing for a Novell network.

An ODI driver (Novell, 1992) is required to interface to the Ethernet card; this hardware interfacing ultimately has to be accomplished by communicating with the card through either I/O ports or a memory region. Also, a hardware interrupt is required for the card to be able to signal events (such as message reception) to the driver.

Unfortunately, the IBM PC's I/O and memory maps are far from well defined, and although many areas are generally understood to be reserved for certain types of devices or uses, the holes are usually a free-for-all and prone to conflicts. For example, the famous bottom 640K of memory for an IBM PC in real mode is reserved for RAM, the top 64K is reserved for the ROM BIOS, and some parts in between are regarded as reserved for ROM BIOS extensions and video controllers, but the usage of the areas remaining (generally between hexadecimal memory addresses 0xC0000 and 0xEFFFF) is not well controlled. In the same way the allocation of IRQs and I/O ports, although also partly adhering to some convention (such as I/O ports 0x3F8-3FF being reserved for COM port 1), is also not well controlled.

Pity the poor end-user who would thus, typically, have to manually edit a Novell-defined text file **net.cfg** containing configuration information for the driver's setup, on a trial-and-error basis until a working solution had been found (and having to load and unload device drivers at the same time as each new configuration was tried). Here is an example of such a net.cfg file:

```
link driver pccard
  frame ethernet_802.2
  int 5
  port 320
  mem d2000
```

4 A First Pass at Easing the Installation Problem

I was involved in architecting a new system for easing the installation problem (Xircom, 1995). Not only is this problem a significant one for experienced Information Systems personnel who historically have been responsible for installing such new hardware and software devices, but even more so now with PC Cards becoming popular consumer products, and thus purchased by novice end-users.

Our first version used the basic principle of the core installation code, hidden behind either a modern Windows or DOS graphics front-end, procedurally (in 'C') examining system resources and attempting to make informed procedural decisions about their availability or otherwise. This method was extremely successful, but prone to the expected problem of difficulty of maintainability.

5 A Second Attempt, Using AI Techniques

We decided to attempt to separate the two areas of (a) resource allocation analysis and (b) free resource analysis, and abstract out the task of a rules based system which had previously been too embedded in conventional procedural code. We found that resource allocation analysis could be performed very effectively using conventional C or C++ techniques (Borland International, 1994), albeit still with much embedded understanding of PC hardware and operating system architectures and resource usage. We also discovered a very effective way of attempting to ascertain a free set of required resources: the resource allocation analysis system would create a set of applicable rules (e.g. a file containing a set of Prolog (Inc., 1994; Bratko, 1986) predicates) for the chosen rules-based Expert System (e.g. Prolog), which was then fed into the Expert System engine along with our own set of precompiled rules-based constraints, again based on an understanding of PC hardware and operating system architectures and resource usage. In this manner we found that the Expert System had an extremely high success rate of determining correct resource parameters, with the added bonus of being much easier to maintain (and not just by the original development team, as it turned out). The rest of the installation software could then automatically create configuration files (such as net.cfg) for the end-user, there being very little or no end-user interaction at all in the whole process.

6 Conclusions

I hope to have shown that techniques from AI, which has perhaps been waning in its acceptability of late, can be applied to real-life consumer software applications problems resulting in the twin benefits of ease of use for the end-user, and better quality and maintainability of code by the development company.

References

Amzi! Inc. (1994). *Cogent Prolog 3.0 Interactive Development Environment*.

Borland International, Inc. (1994). *Borland C++ for Windows*.

Bratko, I. (1986). *Prolog Programming for Artificial Intelligence*. Addison-Wesley.

Hogan, T. (1988). *The Programmer's PC Sourcebook*. Microsoft Press.

Norton, P. (1985). *The Programmer's Guide to the IBM PC*. Microsoft Press.

Novell, Inc. (1992). *ODI Developer's LAN Driver Toolkit guide for DOS Workstation HSMs*.

Personal Computer Memory Card International Association (1992). *PCMCIA Standards*.

Schulman, A., Maxey, D., & Pietrek, M. (1992). *Undocumented Windows*. Addison-Wesley.

Xircom, Inc. (1995). *CreditCard Ethernet+Modem 28.8 Installation Software*.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

Multimedia interfaces and anaphora resolution

Marco Rocha
marco@cogs.susx.ac.uk

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

Abstract One way to try to overcome the difficulties involved in anaphora resolution when designing natural language interfaces is to build a capability for control by the user into the interface. The user can thus assess whether the interface is interpreting anaphoric references correctly. Multimedia technology offers new possibilities to integrate such control by giving the user feedback of coherence between the current input and the ongoing discourse. After a general discussion of matters concerning the integration of visual and discourse information, the paper will present two existing attempts to use multimedia techniques as described above. It will then discuss particularly thorny aspects of anaphora resolution which might be made easier by employing multimedia techniques, together with a few practical suggestions.

1 Introduction

As McKevitt (McKevitt, 1994) points out, one of the reasons for the skepticism about machine understanding of natural language probably results from the fact that, differently from human beings in real-life dialogues, systems hardly ever make use of visual information processed to support and disambiguate natural language input. At the methodological level, this may be a consequence of the reductionist influence, which hindered the integration of vision processing and natural language processing. A practical aspect is also likely to have played a part, as these are two very difficult AI problems which at first sight might seem even harder when attempts to create computational models for the integration of vision processing and natural language processing are envisaged.

However, the integration of information from these two sources can often simplify the individual tasks, and it seems now evident that a discourse understanding system which cannot deal with references to the visual world does not tap into a major source of relevant and often crucial information. It does not appear wise to persist in overlooking the fact that humans integrate both abilities to communicate. This is just the same for reading situations, as exemplified by captioned photographs. The reverse is also true in the capacity to dynamically construct scene descriptions, which could be a powerful tool in robotics (see (Nualláin, 1994)). The whole approach bears relation to the idea of symbol grounding, meaning the notion of a putative language of thought common to vision and language into which both systems would be mapped.

On the other hand, there is a correspondence problem when practical ways of achieving the integration of visual information with words are devised. The simple association of pictures with words does not amount to actual integration. Visual information must match phrases, entire sentences and ideally full discourse with complex descriptions of events. Therefore, knowledge bases containing language models and visual models are needed, even in systems which take either language or visual inputs but not both.

For instance, in natural language assisted graphics, several pictures can be associated with a single sentence, because presupositions are a fact of life in natural language. Thus, a container is presupposed in the sentence below:

(1)
The wine is on the table.

Similarly, a system for optical character recognition can benefit highly from syntactic knowledge structured into statistical models derived from part-of-speech transition probabilities. This can be further developed into systems which are able to fully analyse the structure of a document, such as Unysis' IDUS (see (Liewbowitz et al., 1994)), which integrates natural language processing and image processing to achieve optimum results. Other applications aim at map and diagram understanding as well as at a variety of tasks involving the transformation of language-based descriptions into image-based descriptions and vice-versa.

2 Multimedia systems and reference

The advent of multimedia systems enables the integration of data from various modalities, such as text, tables, diagrams, photos, involving various media, such as paper, electronic, audio and video. As a result, intelligent user interfaces have broken free from the strictly linguistic type of interaction to rely on the graphic depiction of the interface domain, with relevant concepts explicitly shown and available for direct manipulation (see (Sullivan & Tyler, 1991)). Therefore, interfaces do not have to process and hopefully interpret the meaning of a natural language input correctly in order to grasp user intents. As they involve manipulation, the actions of an user are immediately accessible for an intelligent system, which can thus interact with the user in easy and clear ways.

These advances relate directly to the correct identification of referents involved in a query of any kind. When saying *these*, a user can simultaneously click onto a certain word, area or node in order to single out accurately the object referred to. Hypertext nodes can make such integration even more powerful, allowing users to explore the data universe in question when they haven't realised precisely what they are looking for or where it is located in the system. Conversely, the problems of disorientation and cognitive overhead (Conklin, 1987), resulting from the existence of too many hypertext links, can be minimised by the use of natural language to express complex goal-oriented behaviour.

The CUBRICON system (see (Neal & Shapiro, 1991)), developed at Calspan-UB Research Center, accepts input from speech, keyboard and mouse device pointing to objects on a graphics display. Thus, the user can say *this SAM system* and point to the object he or she means, avoiding the need to specify in degrees of latitute and longitude what object is meant by the phrase. The system also generates a response where higlighting, colours, written and spoken natural language, tables and histograms are combined according to what the system deems as most appropriate to express the information, on the basis of its nature and characteristics. The system may choose to respond in natural language only, when the query demands a short and nontechnical response. An example is given below:

(2)
User: Is this [point] a steel plant?
CC : No, it is a munitions factory.

It may add graphics such as blinking or it may open a text box guiding the user to a different display with a table, for instance.

The ALFRESCO Interactive System (see (Zancanaro, Stock, & Strapparava, 1992)) is a prototype developed at the IRST - Istituto per la Ricerca Scientifica e Tecnologica, in Trento, Italy - as an interactive natural language-centred system for a user interested in frescoes and paintings. It runs on a workstation connected to a videodisk unit and a touch screen. The user interacts with the system by typing sentences, navigating the underlying hypertext and using the touch screen to point at images displayed. The user can point to a reproduction of a fresco and ask questions such as the one below, with the appropriate response:

(3)
Who is she?
Madonna.

ALFRESCO also provides a module to give feedback to the user of the coherence found between the current input sentence and the ongoing discourse. The user can thus interfere if misunderstandings occur. This feature may prove invaluable in case the referent of an anaphoric pronoun is identified incorrectly.

3 Two suggestions

One difficult problem within the context of anaphora resolution is identifying referents which are chunks of discourse. Such chunks vary in length, but they are not unfrequently quite long. The anaphor typically used for this kind of reference are the demonstratives *this* and *that*, although the personal pronoun *it* is employed for this form of anaphoric reference at times. A system would have initially to determine whether the referent is an object, expressed by a noun phrase, or a discourse chunk. This may be made a lot easier if the user is expected to use a pointing device to identify objects referred to. Whenever no pointing occurs, the referent is not an object, but a discourse chunk.

Two difficulties persist however. The first one is that the user may occasionally not point to anything in the screen, in spite of the fact that the referent intended is an object. This might be overcome by adding a procedure which would generate a natural language message whenever the demonstrative pronouns are used without association to pointing. The second is of course to identify the precise discourse chunk referred to once the type of anaphora is ascertained. A feedback module would have to rely on fully processed preceding discourse information to generate a referent candidate. It would be often necessary to present summaries containing the gist of preceding discourse as resolution options to the user. Feedback for this form of anaphora in multimedia interfaces clearly demands further research.

The second problem are collocations containing anaphors, such as *that's it* or *that's right*. It is often easier to process such references as units, and this is certainly true in the case of the first one, where separate referents for *that* and *it* are rarely the correct answer. A list of such collocations with typical procedures for resolution may be created after appropriate real-language corpus investigations. Such list would also help the system deal with the chunk-of-discourse anaphora type discussed above.

4 Conclusion

The idea of communicating system inferences concerning discourse processing to the user may go a long way to ease the burden of anaphora resolution in natural language understanding systems. It is important to note, however, that this burden cannot be simply shifted to the user as his or her responsibility. Before the advent of multimedia systems, this would be equal to return a metalinguistic query to the user, which would be certainly not desirable. The powerful graphics presently available have made constant feedback possible, without overburdening the user. The development of ways to present feedback information to users and enable them to interact with the system in discourse understanding may hold the key to successful natural language processing.

References

Conklin, J. (1987). Hypertext: an introduction and survey. *IEEE Computer*, *20*.

Liewbowitz, S., et al. (1994). Integrating natural language processing with document structure analysis. *Artificial Intelligence Review*, *8*(2-3).

McKevitt, P. (1994). Vision processing and natural language processing - editorial. *Artificial Intelligence Review*, *8*(2-3).

Neal, J. G., & Shapiro, S. C. (1991). Intelligent multi-media interface technology. In *Intelligent User Interfaces*. Addison-Wesley.

Nualláin, S. Ó. (1994). An investigation into the common semantics of language and vision. *Artificial Intelligence Review*, *8*(2-3).

Sullivan, J. W., & Tyler, S. W. (1991). Introduction. In *Intelligent User Interfaces*. Addison-Wesley.

Zancanaro, M., Stock, O., & Strapparava, C. (1992). Dialogue cohesion sharing and adjusting in an enhanced multimodal environment. Tech. rep. 9212-05, Istituto per la Ricerca Scientifica e Tecnologica.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

Reconstruction of the neuronal network underlying feeding behaviour in the pond snail *Lymnaea stagnalis*

Stephen Dunn
stephend@cogs.susx.ac.uk

Sussex Centre for Neuroscience
University of Sussex
Falmer
East Sussex
UK

COGS
University of Sussex
Falmer
East Sussex
UK

1 Introduction

The pattern-generating neural circuitry underlying rhythmic feeding behaviour in the pond snail *Lymnaea stagnalis* is an ideal candidate for the neuroethological study of an entire biological neural network. I am using computer modelling to learn more about the mechansims underlying the generation of this rhythmic behaviour. This work is taking place in collaboration with another approach in which the circuit is being reconstructed in cell culture, enabling the examination of synaptic and cellular properties in controlled conditions[1]. The work on both projects is based on previous intensive anatomical, electrophysiological and behavioural studies (for a review, see Benjamin and Elliot (1989)). With this combination of collaborating projects we will come closer to understanding this circuit; how it functions as a whole, and how its activity is modulated to produce observable behaviour in the animal. This short paper explains the approach.

2 Feeding Behaviour

Lymnaea is a browsing herbivore that feeds on submerged algal films or floating vegetation. During feeding, the buccal mass performs a stereotyped series of rhythmic movements as depicted in the cartoon shown in figure 1. This is repeated for hundreds of cycles as the animal moves over the food substrate, typically moving its head from side to side. The cyclic movements, which result in food being scooped into the mouth and swallowed, may be divided up into four main phases: three active, and one during which the musculature is at rest. *In vivo*, each cycle is typically 3–5 seconds in duration. The three active phases of the rhythm can be summarised as follows: An initial protraction accompanied by mouth opening results in the protrusion of the radula (tongue) from the mouth (Protraction phase). The radula is then rasped across the food substrate (Rasp phase) and the gathered food is forced into the oesophagus with a strong backward motion of the radula (Swallow Phase). Following retraction a new cycle is initiated by the repeated protraction of the radula. It is the circuitry underlying the production and control of these three active phases of behaviour that is the subject of our study.

[1]The cell culture approach is being pursued by Volko Straub at Sussex Centre for Neuroscience.

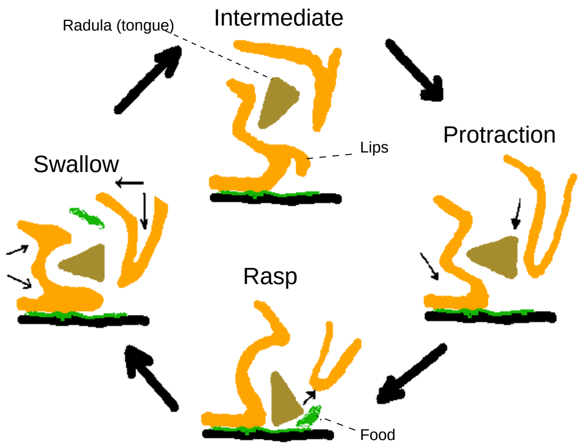

Figure 1: Cartoon cross-section depicting the buccal mass movements in the three active behavioural phases, plus the intermediate rest phase. **KEY:** Large arrows indicate the sequence of the behavioural cycle - small arrows indicate movements of the buccal mass. [This cartoon is only meant as a simple representation, and as such belies extensive behavioural work underlying the project.]

3 Electrophysiology

Cell bodies in *Lymnaea* are typically large (20-200μm) and readily identifiable from one animal to the next, due to regular colouring and position in the ganglia. In addition, the number of cells involved in this circuit is relatively small. These facts have made it possible to categorise nearly all the cells involved, their synaptic connectivity and firing characteristics. Underlying the feeding behaviour is an oscillatory circuit of interneurons which generates the rhythmic pattern driving the cyclic activation of the motoneurons. The rhythmic output appears to be a collective property of the network rather than due to the action of a single pacemaker cell. In addition, it is maintained in the absence of sensory feedback, and as such is a Central Pattern Generator (CPG). (for a discussion of this concept see Selverston (1980).) A summary of the activity, during 4 feeding cycles, of the 6 main central pattern generating interneurons and one modulatory neuron (which drives the pattern at rates observed *in vivo*) is shown in figure 2. The six CPG interneurons are identified as such due to their abilities to drive and reset the feeding rhythm. The interneurons N1-N2 -N3 are labelled according to which active motor phase they typically fire during (1, 2 or 3) and then according to subtype (e.g. N1L (lateral), N1m (medial)). There is no evidence to suggest that motoneurons play a direct role in pattern generation (Elliott & Benjamin, 1985).

4 Modelling

Our eventual aim is to have a working computer model which includes all of the cells involved in pattern generation and modulation. Computer modelling is just one of a variety of tools for understanding this system. The advantages of performing experiments on a computer model can be summarised as follows:

- The inherent modularity enables simple isolation of neurons in a network and alteration of connectivity, enabling the testing of hypotheses and confirmation that a circuit is sufficient to produce a given pattern.
- Simultaneous recording of all modelled cells is possible along with the monitoring of other properties that would be impossible to measure physiologically.

The feeding CPG is being modelled on a Sun SPARCstation 20 using the SNNAP simulator (Ziv, Baxter, & Byrne, 1994) which allows the modelling of many Hodgkin and Huxley type neurons; each with

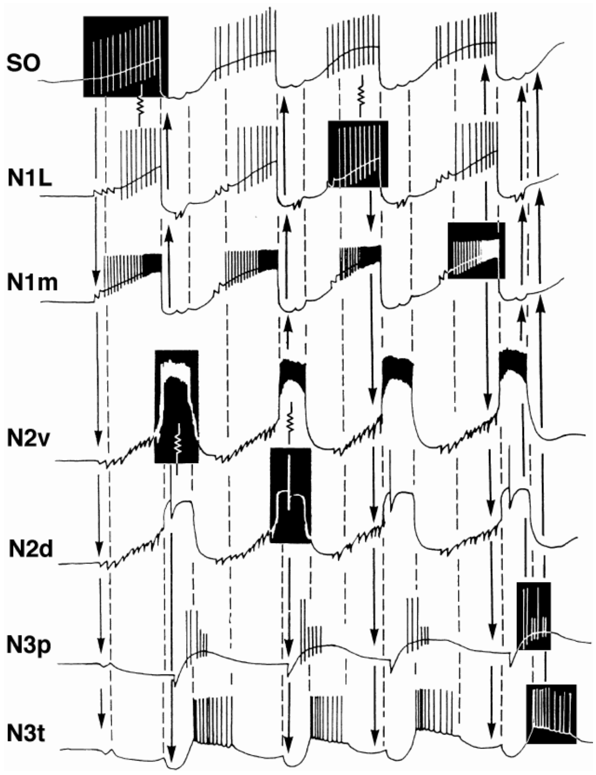

Figure 2: CPG activity summary diagram

This figure constructed from many paired intracellular recordings illustrates the diversity of cellular properties in the CPG network. **KEY:** On each feeding cycle, a different cell's connectivity is indicated. The presynaptic cell is highlighted and its synaptic connections shown by arrows. For example; the first trace to be highlighted is that of the slow oscillator modulatory cell (SO). The arrows indicate chemical synapses with the N1m, N2v, N2d, N3p, N3t cells, and an electrical synapse with the N1L cell. (edited from a summary diagram supplied by Paul Benjamin.)

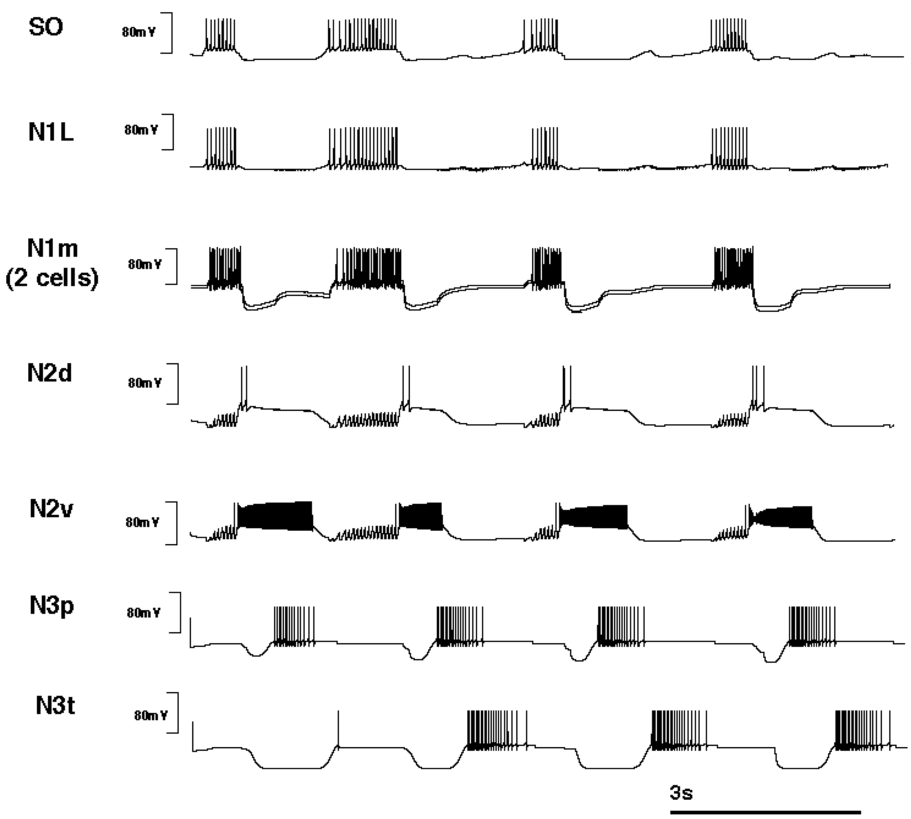

Figure 3: Modelled pattern activity driven by the Slow Oscillator (SO) cell. 2.5nA of simulated current injection to the SO cell (throughout this trace) drives a pattern similar to that as seen in isolated preparations.

an arbitrary number of specifiable voltage-dependent conductances. Neurons are linked with modelled chemical and electrical synapses. Post-synaptic potentials (PSPs) are modelled by the solution of a second order differential equation and are dependent on the duration of the presynaptic spike. The approach has been to start with a minimal model of the patterning network, and add complexity, one cell at a time. Where specific biophysical properties are unknown, the overall firing characteristics of the cell have been reproduced using a combination of appropriate voltage-dependent conductances. As with the firing characteristics, PSPs produced from the chemical synapses have been modelled to match data from intracellular recordings. The current model reproduces the observed physiological data on a network level (output from the model shown in figure 3), including experimental observations such as the pattern-driving ability of the N1m cells, and pattern-resetting capability of N2 cells when stimulated. However there are many free parameters yet to be eliminated.

Detailed modelling without exact biophysical data has been criticised in the past for leading researchers to incorrect assumptions about the circuits under study (Selverston, 1980). This may be true, but does not invalidate its value in focusing ideas and research. The model has already proved useful in examining the termination of phases in the rhythm and the role of intrinsic cell properties in the dynamics of the pattern. Although much is known about the connectivity and cells involved in this network, we do not know which cell properties and connections are important for maintaining and modulating the rhythm. The model serves to highlight which areas to investigate, due to the assumptions forced upon the modeller from a lack of specific data. These areas will then be investigated by the cell culture project. An example of one such area of investigation is the transition between the second and final phases of activity in the pattern:

The only mechanism for the activation of the N3p and N3t cells that will allow patterning to occur is post-inhibitory rebound. This can only occur if the N2v cell plateau is terminated by an intrinsic slow hyperpolarising current. Studies in cell culture will determine if the N2 plateau terminates intrinsically, or whether another, as yet undiscovered cell, is involved.

5 Collaboration and Future Work

The collaboration with the cell culture approach will enable us to enrich the model with data characterising the intrinsic properties of the different neuron types, as well as the connections between them. The computer model will be extended to neurons that are not "main" CPG cells such as the cerebral giant cell which has a dual pattern generating/modulatory effect (Yeoman et al., 1994b), (Yeoman et al., 1994a). This will enable interesting experiments on gating and control of the pattern. Using a high-level, more simplifying model it is also planned to investigate the issue of dynamics in the system. Currently using SNNAP, the fast conductances underlying action potential generation are modelled- this requires much processing power, and may not be necessary. A parallel approach is planned in which spikes are generated automatically when a dynamic threshold is reached (for example Getting (1989)), this will allow more effort to be devoted to accurate modelling slow conductances and PSPs.

References

Benjamin, P., & Elliot, C. (1989). Snail feeding oscillator: The central pattern generator and its control by modulatory interneurons. In Jacklet, J. (Ed.), *Neuronal and Cellular Oscillators*, chap. 7, pp. 173–214. Dekker.

Elliott, C., & Benjamin, P. (1985). Interactions of pattern generating interneurons controlling feeding in lymnaea stagnalis. *Journal of Neurophysiology*, *54*(6), 1396–1411.

Getting, P. A. (1989). Reconstruction of small neural networks. In Koch, C., & Segev, I. (Eds.), *Methods in Neuronal Modelling*, chap. 6. Bradford/MIT.

Selverston, A. I. (1980). Are central pattern generators understandable?. *The Behavioural and Brain Sciences*, *3*, 535–540. (inc. open peer commentry pp 541-571).

Yeoman, M., Kemenes, G., Benjamin, P., & Elliott, C. (1994a). Modulatory role for the seretonergic cerebral giant cells in the feeding system of the snail, lymnaea: II. Photoinactivation. *Journal of Neurophysiology*, *72*(3), 1372–1382.

Yeoman, M., Pieneman, A., Ter-Maat, A., & Benjamin, P. (1994b). Modulatory role for the serotonergic cerebral giant cells in the feeding system of the snail, lymnaea: I. Fine wire recording in the intact animal and pharmacology. *Journal of Neurophysiology*, *72*(3), 1357–1371.

Ziv, I., Baxter, D. A., & Byrne, J. H. (1994). Simulator for neural networks and action potentials: Description and application. *Journal of Neurophysiology*, *71*(1), 294–308.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

The Role of Neural Activity in the Development of the Cat Visual System

Stephen Eglen
stephene@cogs.susx.ac.uk

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

Abstract Mammalian visual systems are very complicated structures, and their development is by no means trivial. This paper discusses the role of neural activity in the development of the visual system. It is shown that visual experience (which generates neural activity) plays an important role in postnatal development, but also that neural activity plays an important part of the developmental process prenatally, before the eye's photoreceptors are functioning. These principles of prenatal neural activity have been captured in a computer model of the development of the cat Lateral Geniculate Nucleus (LGN). This model shows that hebbian like activity driven learning rules can replicate the topography and ocular segregation found in the newborn cat LGN, but only in conjunction with activity independent mechanisms.

Introduction

Visual systems are highly complicated structures, revealing many forms of organisation. A previous paper (Eglen, 1994) outlined the organisation of mammalian visual systems – the emphasis of this paper is on its development. One common feature found in visual systems (both vertebrate and mammalian) is that when an input set of cells projects to another area of cells, they do so in such a way that nearby input cells are mapped onto neighbouring output cells. For example, in the goldfish visual system, the retinal cells project into the optic tectum so that neighbouring retinal cells innervate neighbouring parts of the tectum. This is shown in Figure 1. An important question that has concerned researchers for many years is how these orderly projections develop.

Chemical Markers

Early experiments by Sperry and colleagues through the 1940–1960s examined the regrowth of retinal cells into the optic tectum in amphibians when the optic nerve had been severed. Their results showed that the retinal ganglion cells always seemed to know which part of the tectum to innervate. On the basis of this and similar experiments, Sperry proposed his chemoaffinity hypothesis (Sperry, 1963). Applied to the retinotectal pathway, it claims that each retinal cell has a unique marker, and the tectal cell to which the retinal cell should be wired to has the same marker. During development, each retinal cell searches for the tectal cell with the same (or closest) marker and then connects to it.

The chemoaffinity hypothesis assumes that the connectivity of the cells is rigidly determined by the distribution of the markers in the retinal axons and in the tectum. However, several mismatch experiments showed that abnormal retinotectal mappings could develop during regrowth. For example, Gaze and Sharma (1970) showed that a whole retina could regrow onto half a tectum in a topographic fashion, as shown in Figure 2. These, and other regrowth experiments showed that there is not a fixed affinity between retinal cells and tectal cells.

To account for some of these mismatch experiments, the notion of chemical gradients (or marker gradients) was developed. Rather than having a set of independent markers for each retinal or tectal cell, a chemical gradient could be applied across a set of cells, so that neighbouring cells have similar concentrations of chemical markers, whereas distant cells have different concentrations of chemical markers. A computer model, using a scheme of chemical gradients was developed by Willshaw and Von der Malsburg (1979), which accounted for the mismatch experiments. This model however relied on a process of adaptation of synaptic weight strengths as well as a matching process for the retinal cells to find tectal cells with similar chemical markers. Hence, chemical markers (and chemical gradients) by themselves do not sufficiently describe the full development of the pathway been the retina and the tectum. So, what other processes are crucial in development?

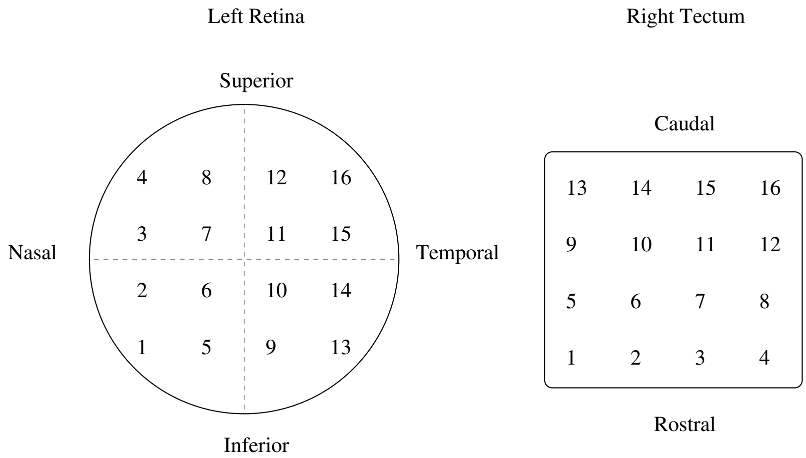

Figure 1: Mapping of the left retina onto the right optic tectum. Numbered retinal cells normally innervate the appropriately numbered area of the tectum. From this it can be seen that the topography of the retina is preserved in the tectum, so that neighbouring retinal cells innervate neighbouring areas of the tectum. Diagram adapted from (Gaze & Sharma, 1970).

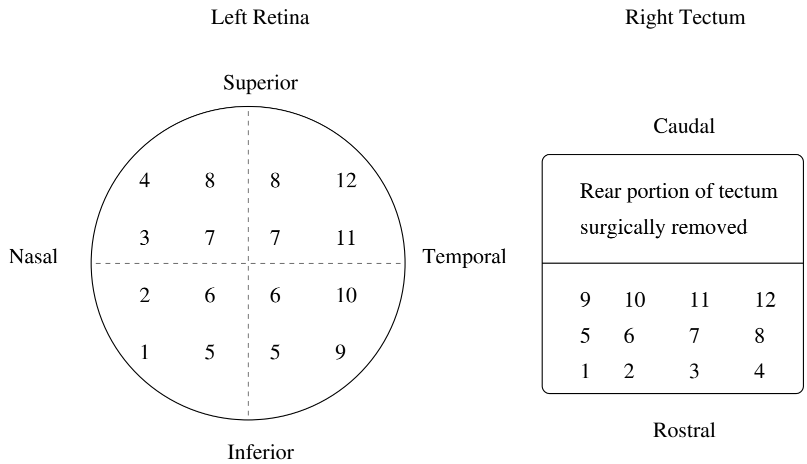

Figure 2: Retinotectal mapping after the caudal (back) half of the tectum has been removed. Although the map in the tectum is still orderly, some tectal cells (marked 5-8) can be stimulated by more than one region of the retina. Figure adapted from (Gaze & Sharma, 1970).

Experience Dependent Postnatal Development

A series of experiments by (Wiesel & Hubel, 1963b, 1963a; Hubel & Wiesel, 1963) demonstrated that visual experience plays an important role in development, by looking at the pattern of ocular dominance of cortical cells in kittens. In a newborn kitten, most cortical cells in the primary visual cortex (area 17) respond to stimulation of either eye, although there is a bias favouring stimulation of one of the eyes. After around 6 weeks of normal visual experience, most cells have adapted to respond to stimulation of only one eye, ignoring stimuli in the other eye. Overall, both eyes normally innervate an equal number of cortical cells. However, if one of the eyes is covered from birth for six weeks so that it does not receive visual input, most (if not all) of the cortical cells do not respond to stimulation of the eye that was deprived of vision.

Wiesel and Hubel (1963b) repeated these experiments with kittens of different ages. They found that the older the kitten was before deprivation began, the weaker the effects of deprivation were. Kittens aged 6 weeks before deprivation showed less severe disruptions to the normal patterns of ocular dominance, and adult cats were not affected at all. These results indicated a *critical period* of development, during which visual deprivation radically alters development.

Other deprivation experiments examined the effects of visual deprivation on the orientation selectivity of cortical cells. In normal cats, cortical cells have a preference for stimuli of a particular orientation. The range of orientation preferences in normal kittens covers the full range (0-180 degrees) of possible orientations. If a newborn kitten is raised in a restricted visual environment consisting of either horizontal or vertical lines, the effects on the distribution of preferred orientations is very severe: most of the cells have a preferred orientation the same as the orientation that was presented during deprivation (Hirsch & Spinelli, 1970). In related experiments, no cortical cells could be found to respond to stimuli oriented greater than 70 degrees away from the orientation presented during deprivation (Blakemore & Cooper, 1970).

A different, and fairly recent, approach to studying development has been to re-route visual inputs into non visual thalamic areas (Sur, Garraghty, & Roe, 1988). Sur and his colleagues re-routed retinal signals of newborn ferrets into the auditory thalamus, as shown in Figure 3. The ferrets were then raised to maturity. Cells in the auditory thalamus were examined, and found to be responsive to visual stimulation. Furthermore, cells in the auditory cortex were also found to be responsive to visual stimuli – with orientation and directionally selective receptive field properties. These properties are similar to those found in normal ferret visual cortex, although the receptive fields tended to be broader than normal. Cell responses also tended to be more sluggish than normal.

Sur et al. (1988) conclude by suggesting that "the modality of a sensory thalamic nucleus or cortical area may be specified by its inputs during development." In other words, it could be that there is nothing intrinsically special about the visual cortex: it becomes the main area for visual processing just because it receives visual input.

Similar experiments have since been performed in other animals. Métin and Frost (1989) re-routed newborn hamster visual inputs into somatosensory cortex. The hamsters were then reared to adulthood, and the cells in somatosensory cortex were examined. They found that the somatosensory cells developed visual receptive field properties similar to those found in normal hamsters. A slightly different approach was taken by Schlaggar and O'Leary (1991). This time, a section of embryonic rat occipital cortex (normally the area which develops to be the visual cortex) was transplanted to somatosensory cortex. Normal somatosensory cortex develops "barrels", which are not found elsewhere in the cortex. In the transplanted occipital cortex however, barrels similar in shape and distribution to normal somatosensory barrels were found. Hence the ability to generate barrels is not due to some intrinsic property of somatosensory cortex, but instead could be driven by the nature of the thalamic inputs.

Taken together, there is a wide range of evidence showing that the development of visual pathways and receptive field properties does not just follow a strict genetic blueprint, encoded in chemical markers; visual experience also plays an important role.

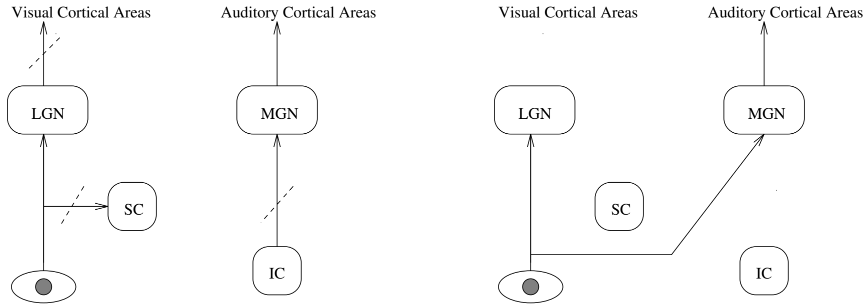

Figure 3: Experimental Procedure used to re-route retinal inputs in the ferret (Sur et al., 1988). The left hand figure shows the surgery performed: dotted lines indicates a pathway that was cut. Retinal inputs were re-routed from the Superior Colliculus (SC) to the Medial Geniculate Nucleus (MGN). Cells in the MGN and auditory cortex were found to be visually driven (IC = Inferior Colliculus).

A popular principle that is normally proposed on the basis of the experimental results just mentioned is that the cortex makes uses of activity driven processes — processes that are driven by the neural activity in the visual pathway (Goodman & Shatz, 1993). Quite often, these processes are self organising, in that the neurons adapt according to the activity of the system, rather than adapting according to some predefined program (Von der Malsburg & Singer, 1988). For example, Blakemore and Cooper (1970) hypothesised that:

> the visual cortex may adjust itself during maturation to the nature of its visual experience. Cells may even change their preferred orientation towards that of the commonest type of stimulus; so perhaps the nervous system adapts to match the probability of occurrence of features in its visual input. (Blakemore & Cooper, 1970, p478)

The deprivation experiments therefore alter the response properties of cortical cells by altering the visual environment, and therefore the neural activity in the system.

The results of the re-routing and transplant experiments can also be explained using an activity dependent process: each part of the cortex adapts (via general learning rules) to its inputs, regardless of where the inputs actually come from. In this view, the development of oriented receptive fields in the visual cortex is not due to intrinsic properties that are special to visual cortex, but rather they are due to the nature of the visual inputs.

Prenatal Development

The previous section has shown that the postnatal development of the visual system relies on activity dependent processes – the neural activity being caused by visual stimulation of the retina. This section will consider the possibility of activity driven processes in prenatal development.

Prenatally, a lot of development takes place in the cat visual system. In the Lateral Geniculate Nucleus, by the time the cat is born, it has already developed two features of organisation, as shown in Figure 4. Firstly, the LGN has formed into eye specific laminae: X and Y ganglion cells from the contralateral retina (the retina in the opposite hemisphere to the LGN) innervate layer A of the LGN, whereas layer A1 receives X and Y ganglion cells from the ipsilateral retina (the retina in the same hemisphere as the LGN). There is also a composite third layer, layer C, which mainly receives input from W and Y ganglion cells. Secondly, there is a smooth topographic mapping of the retina onto the LGN, so that neighbouring retinal cells innervate neighbouring geniculate cells.

When the retinal axons first innervate the LGN, there is an extensive intermixing of retinal axons throughout the LGN, and hence the eye specific laminae and topographic mappings are not yet present – instead, they develop during the period of prenatal development.

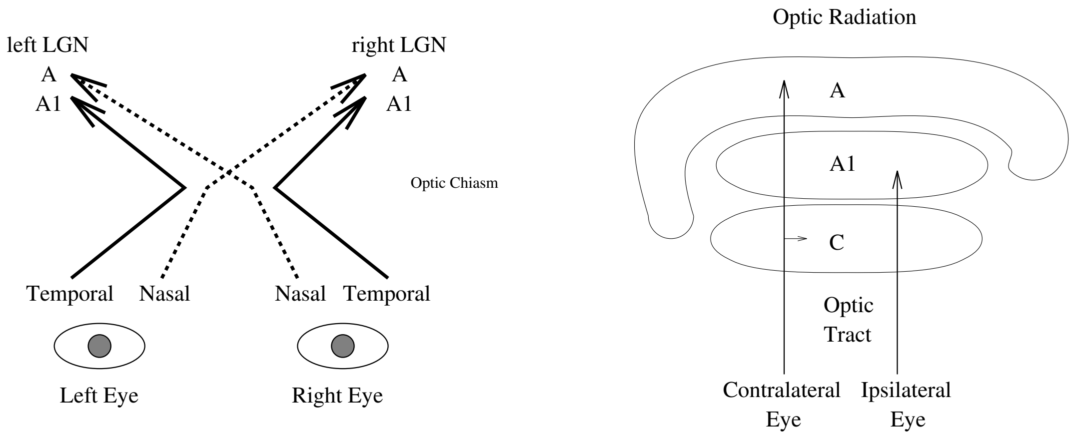

Figure 4: Organization of the cat retinogeniculate pathway. Each LGN receives input from only half of the visual field. The figure on the right shows a coronal section of the LGN, revealing its laminar structure.

At first, it seems impossible to hypothesise activity dependent hypotheses in prenatal development — the photoreceptors in the retina are not yet working, and furthermore the kitten's eyelids do not open until around postnatal day 10 (P10). If the retina cannot transfer any visual stimulation into neural activity, then it seems that there cannot be any neural activity to drive development.

However, recent experiments have shown that this is not the case: even though the retina is not converting visual images into neural activity, it is generating spontaneous neural activity within the retina which is then propagated through the visual pathway. This spontaneous activity of prenatal retinal ganglion cells was first reported by Galli and Maffei (1988) in rats aged between embryonic day 17 (E17) and 21. (The activity was termed spontaneous to indicate that the activity was not evoked by light, but rather by some other means.) Further studies in the rat by Maffei and Galli (1990), simultaneously recording from either two or three neighbouring cells, showed that the activity in retinal ganglion cells is locally correlated, since neighbouring retinal ganglion cells tended to fire at the same time (and also to be silent together). Since then, similar waves of retinal activity have been found in the ferret and cat retina using multielectrode arrays (Meister, Wong, Baylor, & Shatz, 1991; Wong, Meister, & Shatz, 1993).

A popular hypothesis for these waves of retinal activity is that they provide the neural activity to drive hebbian mechanisms of synapse modification to refine the connections between the retina and the LGN (Maffei & Galli, 1990; Meister et al., 1991; Wong et al., 1993).

This hypothesis is supported by evidence from Shatz and Stryker (1988) showing that if the spontaneous activity is blocked (using TTX – a drug that blocks the sodium channel in the ganglion cell axons), then the LGN does not segregate into the normal eye specific laminae. Furthermore, the retinal waves of activity are thought to begin at the right time during development. It is estimated that the correlated retinal activity begins somewhere between E30 and E40, whereas the segregation of retinal inputs into laminae is not thought to begin until around E47. (The prenatal timetable of events is listed in Table 1.) Of course this does not prove that the waves drive the refinement of the retinogeniculate pathway, but by beginning at the right time during development, they are quite likely to be a crucial factor in development.

Modelling Work

To investigate whether the retinal waves could drive the segregation of the LGN into eye specific laminae and produce a topographic mapping, a computer model was developed by (Keesing, Stork, & Shatz, 1992). This model used two activity driven mechanisms to alter the synaptic strengths in a feed-forward neural network model of the pathway between the retina and the LGN. Firstly, a correlational hebbian learning rule was used to implement a hebbian process of increasing weights when a retinal cell and

Day	Event
E23	Retinal axons are yet to arrive at the optic chiasm. (Sherman, 1985, p352)
E30–E40	Correlated bursting activity begins (Wong 1994, personal communication.)
E32	First retinal axons reach the LGN. During next two weeks, there is considerable overlap: most cells in A and A1 are binocular (Sherman, 1985, p351)
E38	Functional synaptic transmission between ganglion cell axon and LGN neuron (Shatz, 1994, p535)
E47	Segregation of retinal inputs into laminae begins.
E48	Loss of ganglion cells begins (Sherman, 1985, p353)
E63	Birth.
P10	Eye opening (Wong et al., 1993, p935)

Table 1: Timetable of cat retinogeniculate development. Days are either embryonic (E) or postnatal (P).

geniculate cell are positively correlated, and conversely to decrease weights when the two cells are negatively correlated. Secondly, a growth rule was used to ensure that the retinal cells grow into neighbouring parts of the LGN.

However, as shown in (Eglen, 1995), these two activity driven processes by themselves are not sufficient to produce the normal pattern of eye specific laminae and a topographic mapping. The model must also rely on two activity independent processes: one to ensure that the contralateral eye innervates the LGN before the ipsilateral eye, and one to provide a rough topographic mapping which can then be refined by the activity dependent mechanisms. Hence, the model demonstrates that the retinal waves can be used to drive activity dependent mechanisms to generate the ocular segregation and the fine topographic mapping, but only in conjunction with activity independent processes.

Conclusions

This paper has outlined the importance of activity dependent mechanisms in the development of the cat visual system. Many experiments have shown that postnatal development of the visual system relies on visual stimulation to provide neural activity, which in turn is used to drive activity dependent processes.

It has also been shown that the activity dependent hypothesis can be applied to prenatal development — before the eye's photoreceptors are functioning. This is because the retina is spontaneously active, producing local correlations in neighbouring retinal ganglion cells. These local correlations can then be exploited by hebbian type learning rules to refine synaptic strengths of retinal axons to produce the eye specific laminae and topographic mappings in the LGN.

It is important to emphasise the importance of both activity dependent processes (for example, the hebbian type learning rules) and activity independent processes (such as the use of chemical markers to attract axons to target cells) in development. In the model of LGN development described here, activity dependent processes alone are not sufficient to account for development: activity independent processes must also be used.

Acknowledgements

Thanks to Julian Budd for comments on a draft version of this paper.

References

Blakemore, C., & Cooper, G. (1970). Development of the brain depends on the visual environment. *Nature*, *228*, 477–478.

Eglen, S. (1994). An overview of motion processing in mammalian visual systems. In de Bourcier, P., Lemmen, R., & Thompson, A. (Eds.), *The Seventh White House Papers: Graduate Research in the Cognitive and Computing Sciences at Sussex*, CSRP 350, pp. 26–33.

Eglen, S. (1995). Modelling the development of the cat lateral geniculate nucleus with hebbian learning. Tech. rep. CSRP 383, School of Cognitive and Computing Sciences, Sussex University.

Galli, L., & Maffei, L. (1988). Spontaneous Impulse Activity of Rat Retinal Ganglion Cells in Prenatal Life. *Science*, *242*, 90–91.

Gaze, R., & Sharma, S. (1970). Axial differences in the reinnervation of the goldfish optic tectum by regenerating optic nerve fibres. *Experimental Brain Research*, *10*, 171–181.

Goodman, C. S., & Shatz, C. J. (1993). Developmental mechanisms that generate precise patterns of neuronal connectivity. *Cell*, *72*, 77–98. Joint Supplement with Neuron, Vol 10.

Hirsch, H. V. B., & Spinelli, D. N. (1970). Visual experience modifies distribution of horizontally and vertically oriented receptive fields in cats. *Science*, *168*, 869–871.

Hubel, D. H., & Wiesel, T. N. (1963). Receptive fields of cells in striate cortex of very young, visually inexperienced kittens. *Journal of Neurophysiology*, *26*, 994–1001.

Keesing, R., Stork, D. G., & Shatz, C. J. (1992). Retinogeniculate development: the role of competition and correlated retinal activity. Tech. rep. CRC-TR-92-02, Ricoh California Research Centre (Also in Neural Information Processing Systems 4, ed. J. E. Moody et al. 1992).

Maffei, L., & Galli, L. (1990). Correlation in the discharges of neighbouring rat retinal ganglion cells during prenatal life. *Proceedings of the National Academy Of Sciences of the United States of America*, *87*, 2861–2864.

Meister, M., Wong, R. O. L., Baylor, D. A., & Shatz, C. J. (1991). Synchronous bursts of action potentials in ganglion cells of the developing mammalian retina. *Science*, *252*, 939–943.

Métin, C., & Frost, D. O. (1989). Visual responses of neurons in somatosensory cortex of hamsters with experimentally induced retinal projections to somatosensory thalamus. *Proceedings of the National Academy of Sciences of the United States of America*, *86*, 357–361.

Schlaggar, B. L., & O'Leary, D. D. M. (1991). Potential of visual-cortex to develop an array of functional units unique to somatosensory cortex. *Science*, *252*, 1556–1560.

Shatz, C. J. (1994). Role for spontaneous neural activity in the patterning of connections between retina and LGN during visual system development. *International Journal of Developmental Neuroscience*, *12*(6), 531–546.

Shatz, C. J., & Stryker, M. P. (1988). Prenatal tetrodotoxin infusion blocks segregation of retinogeniculate afferents. *Science*, *242*, 87–89.

Sherman, S. M. (1985). Development of retinal projections to the cat's lateral geniculate nucleus. *Trends in Neuroscience*, *8*(8), 350–355.

Sperry, R. W. (1963). Chemoaffinity in the orderly growth of nerve fibre patterns and connections. *Proceedings of the National Academy Of Sciences of the United States of America*, *50*, 703–710.

Sur, M., Garraghty, P. E., & Roe, A. W. (1988). Experimentally induced visual projections into auditory thalamus and cortex. *Science*, *242*, 1437–1441.

Von der Malsburg, C., & Singer, W. (1988). Principles of cortical network organization. In Rakic, P., & Singer, W. (Eds.), *Neurobiology of Neocortex*, pp. 69–99. John Wiley & Sons Ltd.

Wiesel, T. N., & Hubel, D. H. (1963a). Effects of visual deprivation on morphology and physiology of cells in the cats lateral geniculate body. *Journal of Neurophysiology*, *26*, 978–993.

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

Willshaw, D. J., & Von der Malsburg, C. (1979). A marker induction mechanism for the establishment of ordered neural mappings: Its application to the retinotectal problem. *Philosophical Transansctions of the Royal Society, London B*, *287*(1021), 203–243.

Wong, R. O. L., Meister, M., & Shatz, C. J. (1993). Transient period of correlated bursting activity during development of the mammalian retina. *Neuron*, *11*(5), 923–938.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

A Scaleable Approach to Face Identification

A. Jonathan Howell and Hilary Buxton
{jonh,hilaryb}@cogs.susx.ac.uk

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

Abstract This paper describes a novel approach to solving two problems: 1) Our 'face unit' network system avoids the unmanagability of neural networks above a certain size by using small, individual networks for each class; and 2) allows the addition of new data to the database without complete re-training of the system. The use of 'face unit' Radial Basis Function networks is shown to give both increased classification performance and more flexible training than conventional RBF methods.

Introduction

Recognising objects and, in particular, the difficult subproblem of recognising human faces is the subject of a great deal of research in computer vision. However, it is only recently that work on biologically-motivated, statistical approaches to face recognition has begun to deliver real solutions. One of the main problems that these approaches tackle is dimensionality reduction to remove much of the redundant information in the original images. There are many possibilities for such representations of the data, including principal component analysis, Gabor filters and various isodensity map or feature extraction schemes. A well known example is the work of (Turk & Pentland, 1991), on the 'eigenface' approach, which is widely acknowledged to be useful for practical application. However, the need for representations at a range of scales and orientations causes extra complexity and updating the average eigenface (used for localisation) when new faces are added to the dataset are problems for this scheme. These difficulties have been overcome to some extent in later work by various researchers (Pentland, Moghaddam, & Starner, 1994; Petkov, Kruizinga, & Lourens, 1993; Rao & Ballard, 1995). In particular, it seems that problems of lighting variation and multiple scales can be overcome by choosing an appropriate representation scheme. (Rao & Ballard, 1995) claim some tolerance to variations in facial features and expressions using natural basis functions. Their representation also addresses the problems of rotation normalisation and scale invariance. However, it seems that greater variations in face orientation, expression, occlusion etc. may still be difficult to overcome in any scheme which does not employ an adaptive learning component.

In this paper we are concentrating on the issues of learning to overcome variability in different views of the same face and the ability of a processing scheme to scale up to larger datasets without compromising discrimination performance. We want our face recognition scheme to generalise over a wide range of conditions to capture the essential similarities for a given face. The Radial Basis Function (RBF) network is a very good candidate given our requirements (Moody & Darken, 1989; Poggio & Girosi, 1990; Girosi, 1992; Ahmad & Tresp, 1993). Its main characteristics are computational simplicity (allowing fast convergence in training) and its description by well-developed mathematical theory (resulting in statistical robustness). An RBF interpolating classifier (Edelman, Reisfeld, & Yeshurun, 1992), was effective and gave performance error of only 5-9% on generalisation under changes of orientation, scale and lighting. This compares favourably with other state of the art systems such as the Turk and Pentland scheme. In an earlier study of our own (Howell & Buxton, 1995a), we found that appropriately trained

RBF networks could perform without error over a range of view orientations for small datasets and that performance was invariant to large ranges of offsets and scales. However, for large datasets performance was much lower and the training was much slower as the network had to cope with many more hidden units. In this study, we address the issue of scaling up by reorganising our RBF networks into smaller 'face recognition units'.

We are adopting the idea of 'face units' for recognising familiar faces from the work of (Bruce & Young, 1986; Bruce, 1988) as they seem a useful way of developing a modular, scaleable architecture. The reorganisation is to allow fast small networks trained with examples of views of the person to be recognised. These face units should give high performance and also alleviate the problem of adding new data to an existing trained network. We are use the various views of the person to be recognised together with selected confusable views of other people as the negative evidence for the network. Our face units have just 2 outputs corresponding to 'yes' or 'no' decisions for the individual. This is in contrast with (Edelman et al., 1992) who did not use such negative evidence in their study. The rest of the paper outlines our approach and presents results to show that this system organisation allows flexible scaling up which could be exploited in real-life applications.

The RBF Network Model

The RBF network is a two-layer, hybrid learning network (Moody & Darken, 1989), with a supervised layer from the hidden to the output nodes, and an unsupervised layer, from the input to the hidden, where individual radial Gaussian functions for each hidden unit simulate the effect of overlapping and locally tuned receptive fields. Unlike a back-propagation network, this means that the RBF activation is related to the relative proximity of test data to the training data. Thus we have a direct measure of confidence in the output of the network for a particular pattern.

The 'Face Unit' Concept

We use the concept of *face recognition units* from the perceptual frameworks for human face processing proposed by (Hay & Young, 1982; Bruce & Young, 1986). Each unit here produces a positive signal only for the particular person it is trained to recognise. For each individual an RBF network is trained to discriminate between that person and others selected from the data set. Rather than using all the data available to train the network against an individual, the strategy adopted was to use only negative data that was most similar (ie confusable) to the positive data. It was anticipated that this data would cause the most 'trouble' when learning to discriminate 'for' and 'against' the individual. Unlike earlier tests which had only positive output signals (one per class), here two outputs are used for each 'face unit' network: 'yes' for the current class and 'no' for all other classes.

The reduction in the size of the network plus the use of negative knowledge, ie images of other people, allows a more efficient coding of the information with greatly reduced training times. Furthermore, people can be added to the data set of a trained set of networks by the creation of a new 'face unit' network for each new individual to be added without retraining the original database.

Form of Test Data

Lighting and location for the training and test face images in these initial studies has been kept fairly constant to simplify the problem. For each individual to be classified, ten images of the head and shoulders were used in ten different positions in 10° steps from face-on to profile of the left side, 90° in all.

Two datasets were used: Type I with two faces (20 images) for quick processing to give a general view of the networks' properties, and Type II with ten faces (100 images). The images were gathered from video as 8-bit grey-scale images. In each image, a 100×100-pixel 'window' was located manually, centred on the tip of the person's nose, so that visible features on profiles should be in similar locations to face-on. This 'window' region was sub-sampled to a variety of resolutions for testing. Full details are given in (Howell & Buxton, 1995a).

Figure 1: Example of 'pro' (top line) and 'anti' (middle and bottom lines) evidence used for a 5+10 "face unit" network

The resolution of the images is represented as '$n \times n$', eg '10×10' for 10 by 10 pixel data. The ratio of training and test images used is represented as 'train/test', eg '2/18', where 20 images were in the data set and 2 were used for training and 18 for test. Several resolutions were used for testing (10×10, 21×21, 44×44, and 90×90) to see how small an image could be used so that the data dimensions and computational effort are kept to a minimum.

Two extra datasets were created to test the RBF network's generalisation abilities: A scale-invariance dataset was produced with five copies of each image re-scaled at 0%, $\pm$12.5% and $\pm$25% of its surface area. An offset-invariance dataset was also produced with five copies of each image: one at the standard sampling 'window' position, and four others at the corners of a box where all *x,y* positions were $\pm$10 pixels from the centre. The random selection of data from this set effectively doubles the variation in data, eg the scale of a test scale-invariance image could be up to $\pm$50% that of a training image.

Types of 'Face Unit' Networks

For the training of 'face unit' networks, the term 'pro' is used to denote hidden units or evidence *for* the class, whilst 'anti' denotes that *against* the class. This evidence was selected according to Euclidean vector distance comparisons with images of the same pose angle of face with 'anti' evidence taken from the class that was the closest (most confusable) to the 'pro' class.

Two types of network layout were used: one where equal numbers of 'pro' and 'anti' hidden units were used, and one where two 'anti' were used for every 'pro'. The latter was used to show whether it would give better negative descrimination, which is important where there are large number of potential classes in large datasets. The 'face unit' network size is denoted by '$p+a$', where p is the number of 'pro' hidden units, and a is the number of 'anti' hidden units. Tests were made on a range of network sizes from 1+1 to 6+12.

Two strategies were investigated for the selection of 'anti' evidence: *Multiple best negative* networks used whichever 'anti' image was closest for each pose angle, so that several 'anti' person-classes could be used. *Single best negative* networks used an average of all vector distances over all pose angles to select one 'anti' person-class to represent all negative evidence. It was anticipated that the latter method would be superior, as a more coherant 3-D class boundary would be given by a single negative person-class for all pose angles.

Fig. 1 shows how the images used for training were selected in an actual test for a 5+10 multiple best negative 'face unit' network. This shows how the same person is not necessarily used for all 'anti' views.

Full details of "face unit" networks and discard strategy are given in (Howell & Buxton, 1995b).

Type I and II Data Tests

With Type I (2 person) data, a conventionally-arranged RBF network achieved 100% generalisation with 10/10 training of the invariant data at all four resolutions, and for 6/14 training at 90×90 and 44×44. The slight difference in performance between the resolutions led to the 21×21 data set being used for later tests. The network attained a peak performance of 97% with 40/60 training with the offset variance data, and 93% for the size variance data. This showed that the RBF network maintained its performance with high variation in the image data.

With Type II (10 person) data, the network attained a peak performance of 78% for 50/50 training with no variation in the data, 37% with 100/400 training with the offset variance data, and 38% for the size variance data. As with the Type I tests, the variant data gave similar performance to the non-variant where the train/test ratios were the same. However, tests were limited by long training times for large networks (eg 2-3 weeks for a 100/400 network).

'Face Unit' Tests

With Type II (10 person) data, a *multiple best negative* 'face unit' network gave average performances of 83% for 5+5 and 6+6 networks. This increased to 89% for 5+5 and 88% for 6+6 networks after discarding low-confidence results. With one 'pro' to two 'anti' training, the average was 83% for 5+10 networks and 89% for 6+12. Discarding low-confidence output gave results of 89% and 96% respectively. Tests with *single best negative* 'face units' gave slightly lower performance on all tests (around 5-8%). Training times were much shorter than for the original Type II data due to the smaller network size (eg about minute for each 6+12 network).

Conclusion/Future Work

In summary, the RBF network 'face unit' organisation has proved to give a flexible, scaleable architecture which can perform at a high level in terms of both classification, generalisation over varying views, and speed of training. It is also a highly modular architecture that allows us to add more data and create as many new face units as are required. In particular, these studies showed that negative evidence plays a crucial role in shaping the discrimination between individuals and that this showed up particularly in the correct 'no' responses of trained units. Multiple views of different people were more effective in improving performance than taking the same number of views of just one confusable person even though we might have expected a clearer decision boundary for the latter. Discard based on confidence gave some improvement and, it is anticipated, will be effective in studies of face recognition from image sequences. This extension of the work will exploit motion segmentation and look at a range of representations of the face data. We are interested in tracking faces and gathering enough information to classify them accurately with good generalisation to other image sequences containing familiar people.

One disadvantage of our current scheme is the need to try all candidate face units during recognition of test data. This could be improved by parallel implementation or an indexing scheme to find the right face unit or set of face units in a hierarchical organisation of the units themselves. The work of (Rao & Ballard, 1995) is particularly interesting in this respect as they claim real-time indexing is possible using convolutions for distance computations to identify likely candidates. Another promising approach uses Gabor wavelet representations (Daugman, 1988) which can be used for segmentation and tracking of faces using transforms of the data and may allow indexing in a similar way. Although such processing schemes are capable of multiscale face recognition and are robust to some changes in expression and orientation, we feel that a better strategy is to characterise the degrees of freedom in the input data required for the application.

References

Ahmad, S., & Tresp, V. (1993). Some solutions to the missing feature problem in vision. In Hanson, S. J., Cowan, J. D., & Giles, C. L. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 5. Morgan Kaufmann.

Bruce, V. (1988). *Recognising Faces*. Lawrence Erlbaum Associates.

Bruce, V., & Young, A. (1986). Understanding face recognition. *British Journal of Psychology*, *77*, 305–327.

Daugman, J. G. (1988). Complete discrete 2-D gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *36*(7), 1169–1179.

Edelman, S., Reisfeld, D., & Yeshurun, Y. (1992). Learning to recognize faces from examples. In *2nd European Conference on Computer Vision*, pp. 787–791 Genoa, Italy.

Girosi, F. (1992). Some extensions of radial basis functions and their applications in artifical intelligence. *Computers Math. Applic.*, *24*(12), 61–80.

Hay, D. C., & Young, A. (1982). The human face. In Ellis, H. D. (Ed.), *Normality and Pathology in Cognitive Functions*. Academic Press.

Howell, A. J., & Buxton, H. (1995a). Invariance in radial basis function neural networks in human face classification. *Neural Processing Letters*, *2*(3), 26–30.

Howell, A. J., & Buxton, H. (1995b). A scaleable approach to face identification. In *Proc. Int. Conference on Artificial Neural Networks (ICANN'95)*, pp. 67–73 Paris.

Moody, J., & Darken, C. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation*, *1*, 281–294.

Pentland, A., Moghaddam, B., & Starner, T. (1994). View-based and modular eigenspaces for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 84–91.

Petkov, N., Kruizinga, P., & Lourens, T. (1993). Biologically motivated approach to face recognition. In *Proceeding of International Workshop on Artificial Neural Networks*, pp. 68–77.

Poggio, T., & Girosi, F. (1990). Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, *247*, 978–982.

Rao, R. P. N., & Ballard, D. H. (1995). Natural basis functions and topographic memory for face recognition. (preprint).

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, *3*(1), 71–86.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

Automated Pipe Route Generation Using Genetic Algorithms

Dae Gyu Kim
dgkim@cogs.susx.ac.uk

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

Abstract This study is about pipe route design using Genetic Algorithms (GAs). Pipe route design work, which is usually done manually in industry, was able to be solved by a computer program.

The pipe route generation process is defined as a kind of optimisation problem, and four different search techniques were used for comparison. GAs showed a better performance than Simulated Annealing (SA) (van Laarhoven & Aarts, 1987), Stochastic Hill Climbing (SHC), and Random Search (RS).

GAs worked better with a large number of loosely interacting population sets. Using SA was not so simple because of the difficulty of parameter tuning, but even the best parameter set in this study did not show as good a result as GAs. Interestingly, on some problems, the performance of SHC was worse than that of RS.

1 Introduction

The pipe routing problem is one of the most important steps in designing systems like ships, power plants, chemical plants, etc. The pipe route decision process has mainly relied on human experts and their experiences because of the highly complex constraints involved in the problem(Kim, 1994).

In the pipe route design problem, equipment must be connected by pipes in a specified way. In most industrial design practice, the location of most of the equipment is determined in advance, and a Piping and Instrumentation Diagram (P&ID) set is prepared, which specifies the connections which have to be made. The pipe route designer then has the job of designing the pipe routes in accordance with the P&ID specification, while meeting several other constraints (avoiding obstacles, keeping pipe lines as simple as possible, etc), and optimising according to objectives such as pipe length.

If the pipe route design which is one of the most time consuming manual processes can be constructed automatically, the design loop from the equipment selection to pipe routing process can be made more than once toward an optimal point. The result from pipe route design can be used in hydrodynamic analysis, which may require some changes in the pipe and equipment sizing, and structure design.

Automatic pipe route design may therefore reduce the time for designing a good overall system. An optimal pipe route for the given system can reduce the cost for building such a system. Reduction of the pipe material is one of the advantages. The down-sizing of equipment followed by the reduction of fluid friction loss has far more effect on the cost reduction than the material itself. These reasons can be a good basis for considering automatic pipe route design.

2 Outline of the work

The pipe route generation process is defined as a kind of optimisation problem(Hazelrigg, 1993), based on viewing it as a variant of the Minimal Rectilinear Steiner Tree problem (Gibbons, 1984), (Hesser, Maenner, & Stucky, 1989), (Julstrom, 1993), (Julstrom, 1994), and four different search techniques were used for comparison.

Some algorithms adopted from (Ganley & Cohoon, 1993) for producing pipe routes in the existence of obstacles were devised and implemented. It was used as a kind of evaluation function for Genetic Algorithms (Grefenstette, 1987), (Davis & Steenstrup, 1987), also it was used with other programs using different search methods as a function for evaluating given pipe configurations.

Five different problem sets were devided for test of the programs. Proformance of each seach method was compared and analysed to find some parameter values giving better result concerning with the problem size, number of pipe connections. Figure 1(left) is the global minimum case configuration for a 6-connection problem.

3 Conclusion

GAs worked fine because it could find the best solutions in a reasonable number of trials, and also parameter selection was not so much important as in SA. GAs worked better with a large number of population sets, rather than large individual population sizes, if the same number of individuals are kept in total.

Using SA was not so simple because of the difficulty of parameter selection: Initial temperature should have been kept in a certain range, temperatures above and below which produced worse result.

SHC and RS worked fine, even though they often had difficulty in finding the global minimum because of problem characteristics. Considering their simplicity in applying to a problem, they can be used for simple design problems, to find reasonable solutions.

Many interesting extra topics were found while doing this study; multiple objective optimisation problem when allowing multiple pipe diameter in the system, extension of this work to three dimension, and combining the pipe route generation program with existing CAD packages.

Number of evaluation steps in reaching the solution increased polynomially even though the problem becomes more complex exponentially, Figure 1(right). This is a very promising aspect of the performance of GAs, which shows the possibility of using GAs in much larger similar kinds of engineering optimisation problems.

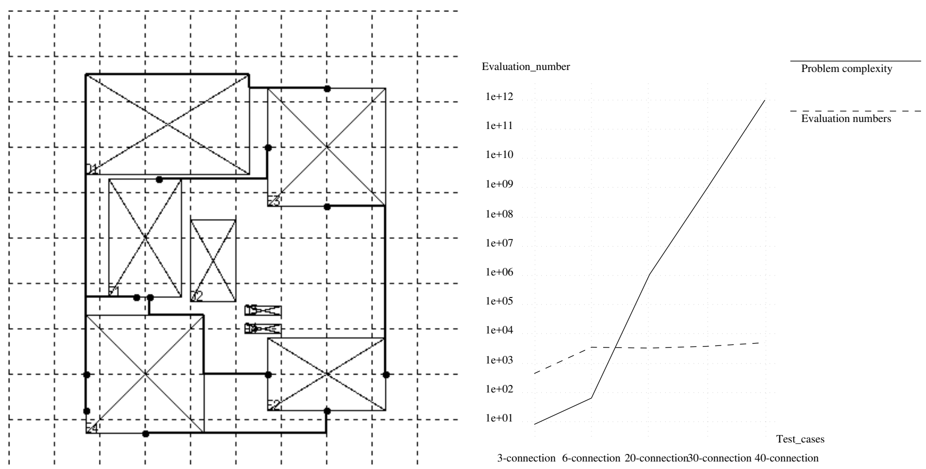

Figure 1: Global minimum case configuration for a 6-connection problem using GAs(left), Comparison of problem size and number of evaluations finding global minimum with GAs(right)

References

Davis, L., & Steenstrup, M. (1987). Incorporating problem specific knowledge into genetic algorithms: An overview. In Davis, L., & Steenstrup, M. (Eds.), *Genetic Algorithms and Simulated Annealing*. Pitman.

Ganley, J. L., & Cohoon, J. P. (1993). Optimal rectilinear steiner tree routing in the presence of obstacles..

Gibbons, A. (1984). *Algorithmic Graph Theory*. Cambridge University Press.

Grefenstette, J. J. (1987). Incorporating problem specific knowledge into genetic algorithms. In Davis, L., & Steenstrup, M. (Eds.), *Genetic Algorithms and Simulated Annealing*. Pitman.

Hazelrigg, G. A. (1993). *Systems Engineering* (Prepublication edition). Institute for Advanced Engineering, Seoul, Korea.

Hesser, J., Maenner, R., & Stucky, O. (1989). Optimization of Steiner trees using Genetic Algorithms. In Schapper, D. (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 231–236. International Conference on Genetic Algorithms.

Julstrom, B. A. (1993). A Genetic Algorithm for the Rectilinear Steiner Problem. In Forrest, S. (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 474–479. International Conference on Genetic Algorithms.

Julstrom, B. A. (1994). Seeding the Population: Improved Performance in a Genetic Algorithm for the Rectileaner Steiner Problem. In *Proceedings of the 1994 ACM Symposium on Applied Computing*, pp. 222–226.

Kim, K. S. (1994). Personal communication with Kang Soo Kim, Executive director in the DAEWOO Heavy Industry Co., Seoul, Korea..

van Laarhoven, P. J. M., & Aarts, E. H. L. (1987). *Simulated Annealing: Theory and Applications*. D. Reidel Publishing Company.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

Whole Cognizers, Phenomenology, and Artificial Life*

Ronald Lemmen
ronaldl@cogs.susx.ac.uk

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

Abstract The view that cognition must be representational is a consequence of the Cartesian assumption that the mental and the material are fundamentally opposed to each other. If we take mind and body to form a unity, rather than a union, cognitive science becomes quite different from what it is usually taken to be. Computational theory, linguistics, and neuroscience are replaced as core cognitive sciences by phenomenology, artificial life, and ethology.

1 Introduction

Cognitive science has been shaped by analytic philosophy and the computer metaphor, which has led to the dominance of such themes and notions as rationality, language, representation, functionalism, and computation. I would like to make a case for a cognitive science which lets itself be shaped by a phenomenological philosophy and biology. Central notions then become: experience, embodiment, ethology, evolution, and dynamics. And the artificial contributions are to be expected from ALife, not from AI.

1.1 Representationalism is Cartesian

Cartesianism starts by opposing the mind to the world, the subjective to the objective, the mental to the physical. Such an opposition does not necessarily lead to dualism, it is also possible to take one of the two realms of existence as fundamental, and arrive at either idealism or materialism. Cognitivists, having chosen the materialist option, think they can reduce the mind-body problem to the mind-brain problem, because they see the body as merely material. The mind becomes some kind of "inner entity", one that is not in direct contact with the world, but is always one step removed from it, having to use its rational powers and empirical cues in order to create a model of the world. Knowledge and understanding are expressed entirely in the representations the mind has of the world and the way it uses them. Representationalism is a form of Cartesianism.

1.2 Organismal Content

There are those who happily embrace the conclusion that we are never in contact with our world, but always only with a model of it. However, as McDowell (McDowell, 1994) shows, on a cognitivist account we cannot even say that much, because such an account leaves out the cognizer altogether. If we introduce the notions of *organismal* and *organal* (sub-organismal) levels, we can say that cognitivist explanations are pitched entirely at the organal level. They are about the informational transactions between one part of an organism and another part, e.g., about "what the frog's eye tells the frog's brain".

*This paper was originally presented as a poster at the fourth conference of the European Society for Philosophy and Psychology in Oxford.

We cannot say that the frog itself can be at the end of such organal tellings. The frog does not have access to its interior, but to the environment. So, although it may seem as if the organal story warrants content ascription, it does not allow us to speak of the organism's contentful involvement with its environment. In fact, McDowell concludes, it is only at the organismal level that it is literally true that there is content involved. The organal story of informational transactions is a metaphorical one. "We have inside us something that is not intelligent at all (it knows nothing and understands nothing)" (McDowell, 1994).

1.3 Towards a Phenomenological Cognitive Science

This view has several consequences that go against the tradition. The notion of mental representation loses its philosophical centrality, becoming a useful scientific metaphor at best. Organismal content cannot be equated with the cognizer's internal structures or processes, but has to be understood in terms of the way the whole cognizer deals with its environment. The only way to make proper sense of such dealings is in terms of the environment *as it is for the cognizer*. It is the *phenomenal*, or *lived*, world that provides the reasons for the cognizer's actions. In fact, when we talk about a world this way, we already start to see that the structure of the world and the structure of a cognizer's content (i.e., the way it experiences the world) are nothing but respectively the object-side and the subject-side of a single phenomenon. We start to see the sense in a phenomenological approach to cognition:

"The world is inseparable from the subject, but from a subject which is nothing but a project of the world, and the subject is inseparable from the world, but from a world which the subject itself projects. The subject is a being-in-the-world and the world remains 'subjective' since its texture and articulations are traced out by the subject's movement of transcendence" (Merleau-Ponty, 1962)

From a Cartesian (e.g., representational) perspective, it may look as if phenomenology will inevitably slip into idealism. Not so if the opposition of mind and world is not taken as a starting point. We should not see cognizers as a problematic union of mind and body, but as mind-body *unities*, which are not just passive receptacles of experiences, but which are concretely and actively involved in the world. I want to argue that if we take such a conception of cognizers, as cognitive scientists we are best served by dropping computationalist, functionalist, or representationalist theories and by taking up a mixture of insights and methods as found in phenomenology (especially Merleau-Ponty's), ethology, and Artificial Life. These fields have in common that they are concerned with *whole* cognizers, their environments and abilities. Thus we can, at last, study *organismal* level phenomena, such as behaviour, understanding, skills, significances, experiences, beliefs, desires, etc.

2 No Need for Representations

A general characterisation of a representation is that it is a "vehicle of content". In thinking about the needs and uses for representations, most cognitive scientists focus on the vehicle-side. But it may be more useful to approach the issue from the content-side, especially since content is *the* phenomenon that sets cognizers apart. In cognitive science we are not only trying to explain behaviour but also the content had by cognizers. But this is a notion of content that attaches to the *whole cognizer* (it is "organismal content"), whereas the representational vehicle is supposed to be a part of the cognizer (it is "organal"). Representationalism is committed to the idea that organismal content is to be explained in terms of organal content. If that were true, it would follow immediately that there could be no cognition without underlying representations. That is exactly what is being challenged here.

The problem with representationalism is twofold. First, it is hard to see on what grounds we could claim that "content has entered the world" when we are on an organal level, whereas the answer is quite obvious in the case of organismal level. It is simply that there now is question of an organism who is a subject of experiences, so that now things *matter*, have significance. Organismal content is experiential content.

Second, it is relatively easy to explain representational content in terms of organismal content (e.g., something is a representation to a cognizer if the cognizer experiences it as representing something else).

But it is impossible to explain experiential content in terms of representations. Representationalism is a form of functionalism and therefore epiphenomenalist with respect to experiences.

2.1 No Experience, No Meaning, No Cognition

Content is always *someone's* content, it is rooted in a cognizer's needs and concerns. Without there being a "mattering" there is no content. Hence, the cognizer has to be the kind of entity to which things *can* matter: an experiencer. A theory of content has to presuppose and build on a theory of experience. But cognitivism has it exactly the other way around. It wants to understand the cognitive properties of the whole cognizer on the basis of properties of its parts, and attempts to show how content enters the picture at an organal level (it wants to naturalise content). But at this level there is no question of a "mattering" which can explain the presence of content, since the whole idea is to explain experiential content on the basis of representations.

Since content is a matter of experience and since representationalism can't take experience seriously, it can't take content seriously either. If consciousness is epiphenomenal, content is as well.

2.2 The Representation Metaphor

A representational view can emerge in several ways. But there is no way of establishing a form of representational content which is independent of and prior to experiential content. Here is how the most common approaches fail to establish anything more than metaphorical content.

(1) **Homunculi**

We tend to decompose complex machinery into separate bits the working of which we find easier to understand. This has led us to think of cognizers as societies of homunculi. Even if such functional decompositions are warranted, it does not follow that the homunculi deal in representations, for *they* aren't capable of understanding anything as representational, and, what is more, it doesn't matter to them. Homunculi do not have experiential content and therefore cannot endow anything with meaning.

(2) **Natural Signs, Interpretability, Explanatory Power**

It is sometimes argued that certain entities are representations simply because it is possible, and maybe even useful, for us to see them that way. But we are also capable of seeing natural signs as "indicating" something else. Natural signs do not count as representations because they weren't "meant" to be representations. The same applies to patterns in the brain.

With the latter point we have touched on the matter of normativity. Experiential content supplies its own normativity, but the representationalist has to get it from a more 'objective' source.

(3) **Truth**

It is often claimed that some entity is capable of representing because it also is capable of misrepresenting. What is 'forgotten' is the implicit reference to the representation-consumer and her *interests* (note that the observers may even be the only representation-consumers, as in (2)).

(4) **Evolution and Adaptive Function**

When there *is* a concern with the representation-consumer's interests, it is usually claimed that these are gifts of evolution. But Darwinism teaches us that evolution has no goals, despite the fact that it can be handy to treat it *as if* it has. But all such function-talk, hence all representationtalk, is metaphorical. Genuine content is experiential.

Experiential content is not built up out of smaller content-components, but arises out of the dynamic relations between the cognizer as a whole and its environment. Content should be studied at its own level, the organismal level. The notion of representation has no interesting role to play here. It is just that once there *is* a cognizer, it will necessarily exhibit internal patterns which are correlated with entities in the external world, and it is very natural for an interpreter to refer to these in terms of the metaphor of "representation".

3 Merleau-Ponty, Embodiment, and Experience

3.1 The Mind-Body Unity

There is no mind-body problem. Mind and body are not two mutually exclusive entities which have to be brought together (Cartesianism), but are two aspects of a single unity. Nothing is either purely subjective or purely objective, except as an ideal abstraction. The unity (not union) of body and mind implies the presence of the mental in the bodily and that of the bodily in the mental. The mind is not an entity somewhere *in* the body, but it *is* the body. Merleau-Ponty (Mallin, 1979) speaks of the "body-subject", a beingin-the-world which–"anonymously", or pre-personally–traces out the texture and articulations of its world, so that the conscious subject *finds* itself in an already meaningful world. Nevertheless, there is no question here of objective meanings for they are and remain tied to this subject, with her way of carving up the world (phenomenology does give a lot of attention to intersubjectivity, but that is beyond the scope of this poster).

Is it not still true that there is a world out there which simply, and objectively, *has* the properties it has? Such a question, however, already implies an allegiance to a Cartesian split between mind and world. In contrast, Merleau-Ponty tells us that the primordial constituents of the lived world are not objective properties, but *situations*. Situations are as much part of the subject as they are of the world, they always have both a subject-side and an object-side which are inextricably linked to each other (Mallin, 1979).

Situations only emerge thanks to (either innate or acquired) structures *of the cognizer*. Situations are modes of being-in-the-world. The most fundamental situations are also the most general ones; in them we are at one with the world, we do not experience ourselves as distinct from the world, and we do not carve up the world in objects, properties, and events. (This is meant to be reminiscent of Cussins' treatment of non-conceptual content (Cussins, 1990).)

The unity of mind and body implies that many traditional representationalist conceptions are mistaken. For example, since the mind is not an inner entity, we should not think of the mind as infusing some gesture or expression with meaning. Instead, we should say that the meaning is *in* the gesture and that the expressed is not separable from the expression. The notion of intentionality also changes considerably. The Cartesian view of the mind as an inner entity leads naturally to the question how it can be that entities locked up in it can actually be *about* the world. For us, however, the question of intentionality becomes one about the *contact* a body-subject has with its world. It is about the complementarity of a cognizer and its life-world, about what it is to be in a situation.

But the focus of this poster is on the fact that the unity of mind and body forces us to take both embodiment and experience seriously, something which traditional representationalism is utterly incapable of.

3.2 Taking Embodiment Seriously

The structure of the mental is rooted in and dependent on the structure and capacities of the body. This is demonstrated by Lakoff and Johnson (Lakoff & Johnson, 1980), but it is Merleau-Ponty who investigated the deeper reasons for it. He showed that worlds are lived, not conceived, and that, although we "find" ourselves in an already meaningful world, we *also* "bring forth" our worlds ourselves. For example, we *find* ourselves, not in objective space, but in oriented space. There already is near and far, up and down. Those primordial meanings are a result of our *situatedness* and of our bodily skills of manipulating things and moving around. To put it simplistically, 'near' is what is within reach, and therefore dependent on the body. It is only because we live this primordial, oriented space that we can arrive at a notion of objective space.

The same kind of argument can be applied more generally. Whatever properties I perceive, they are *my* way of carving up the world, and of situating myself. For example, it is only because I have a body that tires and has joints that bend in certain ways, that my world is one with chairs. At this level, I have much in common with other humans, so we can have a common world. But because we also have different concerns (e.g., cultural), we end up with a plurality of worlds. There is no unique, objective picture of reality, every world is from a certain point of view. There is intersubjectivity, though, which is based on an ability of taking up someone else's situation.

3.3 Taking Experience Seriously

Embodiment is indeed taken seriously more and more these days, certainly within ALife. However, it is not generally appreciated that the taking seriously of embodiment *requires* the taking seriously of experience. The fact that mind and body are one does not only mean that we need to recognize the body in the mind, but the mind in the body as well. The body can no longer be treated as a mere physical thing, as it is in Cartesianism. The body *is* the mind.

It follows that part of doing cognitive science is doing phenomenology, that is, finding out about other creature's life worlds. It will be objected that there is no way we can get to know about someone else's subjective experiences. However, and fortunately, much of the power of that objection derives from the Cartesian myth of the mind as an "inner", unworldly entity (and from confused ideas about "qualia"). The Cartesian already has doubts about the contact of mind with world, let alone the possibility of contact between one mind, through the world, with another. But we no longer accept this idea of the mind as being shielded off from the world, and so we have taken the first hurdle in understanding how cognizers can understand each other, namely through the body (Sheets-Johnstone, 1990).

4 Studying Cognizers

The mind is nothing like a desituated, disembodied computer program running on the brain. Cognition is not the outcome of representational processes, but representations are the product of cognition. And it is not the brain that cognizes, but the *whole, situated* cognizer. This implies that many of the conceptions and methodologies of cognitive science need overhauling. But before we can start doing that, it is important to get rid of the idea that brains might be minds. Also, with some basic phenomenological insights in place, we can see a few other interrelated questions for cognitive science. These are: what do phenomenal worlds look like and how do we find out, and: what are the structures or processes that introduce a subject into the world?

4.1 Brains in Vats are not Minds

In Cartesian vein, it is often argued that brains in vats could be minds. But we are not "in the predicament of our nervous systems, blocked off from the environment by transducers rather than inhabiting it" (McDowell, 1994) and so we should not identify ourselves with our brains. Brains do not think. The structure of the experiential (i.e., that of the world) is to be explained, not in terms of structures in (or programs running on) the brain but in terms of bodily structures and abilities (Sheets-Johnstone, 1994). It is the whole embodied cognizer that is the source of significances, that interacts with the world and thus comes to understand it (first by way of know-how, then propositionally). There is no such thing as pure perception, it is all bound up with mobility and action. The world we perceive is the same as the one in which we act, and is brought about in communication with "otherness". Brains, on the other hand, are not agents, they just sit there.

But is it not true that the body can be simulated, so that from the perspective of the brain it *does* (appear to) have a body? No. The mistake is to think that I have been talking about the objective body. But our starting point has to be the phenomenal body, as Israel Rosenfield lucidly shows with the use of neuropsychological case histories (Rosenfield, 1993). It is "the phenomenal body, that is the true version of the body that we live by . . . , so that the [mind-body problem] has nothing to do with the objective body, which exists only conceptually, but with the phenomenal body" (Merleau-Ponty, 1962). We are thus not forced to say cognition happens within the boundaries of the skin. We are where our project is, where we run up against "otherness". So visually, I can inhabit another room than the one I am sitting in. Dennett can be where his body is, and it does not matter where his brain is (Dennett, 1981). We are not shut up within the boundaries of the skin. We are open to and even intertwined with the world, which is why it can happen that an instrument (a hammer, or even a car) becomes a true extension of the (phenomenal) body. The mechanisms of the mental are to be found not only inside the cognizer's objective body, let alone her brain.

4.2 The Physical Depends on The Mental

In order to treat a cognizer *as* a cognizer we have to start "from within", for insofar as there is meaning, it is a matter of the way in which the cognizer experiences (or "has") a world. But this is a phenomenological project, whereas cognitive science is about the *mechanisms* of cognition. Although we will not *identify* mental happenings with bodily or physical happenings, there is nothing to stop us from investigating the physical side of cognitive processes. We just need to remember that those very same processes also have mental aspects with which the physical ones are intertwined. The purely mental and the purely physical are abstractions; the mental and the physical do not exist independently of each other.

So what does this all imply with respect to the way cognitive science should go about?

4.3 Enabling Explanations

It is still important that the cognizer's innards have the right kind of organisation, and it makes perfect sense to research that organisation and the way it enables the cognizer to create and maintain a world. But we should resist reductionist tendencies, which means we should not infer that the structures and processes in the cognizer's head which *enable* cognition are also the structures and processes which *constitute* cognition. There are no meaning-entities in the head, nor are there sub-agents with somewhat less intelligence or intentionality than the whole agent.

But if we do take a "mechanistical attitude", it is natural to talk about the cognizer's "control mechanisms", and as soon as we do *that*, especially in complex cases, we all too soon try and functionally decompose the system and talk about it in information processing terms. Leaving aside the question if natural cognizers *allow* for such decompositions, we have to take notice of the fact that what we are doing is try and "make sense of an animal's internal control mechanism, and connect it conceptually to the competence it is supposed to explain, . . . by describing it *as if* it were, what we know it is not really, a semantic engine" (McDowell, 1994). McDowell claims that nobody knows another way of making sense of the mechanism.

4.4 Dynamical Systems Approach

That, however, may not be true anymore. The seductive power of the representational metaphor derives to a large extent from the fact that the control mechanism has been modelled in *computational* terms. Van Gelder (van Gelder, 1992) lists various features of computationalism (Turing machines are digital and deterministic, their operation is essentially atemporal, their state transitions are local, and there is a low degree of interdependencey among state variables) that encourage one to think of the described systems in syntactic terms, i.e. in terms of the manipulation of representational constituents. Van Gelder points out that there *is* a genuine alternative to computationalism, namely the Dynamical Systems Approach (DSA) which simply claims that "the most appropriate framework for the study of cognition is dynamical modeling and dynamical systems theory".

From the present perspective DSA is interesting, first, because of itself it is absolutely neutral with respect to the question, Representations, Yes or No? Also, the approach does not presuppose that there is a 'true' decomposition of the system. It is conducive to the idea that the mechanical underpinnings of cognition do not constitute cognition, and that the "mechanics" are nothing but an aspect of a unitary phenomenon which also has experiential aspects. A dynamical explanation does not need to be in terms of (organal) meanings at all. Finally, DSA firmly subscribes to the importance of situatedness and embodiment. Dynamical systems are always seen as being part of more inclusive systems, and it is understood that what constitutes a dynamical system may depend on the situation.

4.5 ALife, Ethology and a Fifth Aim

DSA, then, is sensitive to the importance of the interaction between a system and its environment, and allows us to talk of cognition as a property of the *whole* cognizer, while also providing a means of studying the cognizer's innards. It is thus not surprising that DSA is especially popular within ALife, where there is a general dissatisfaction with the representational, disembodied, non-situated approaches of traditional cognitive science (e.g., Brooks, 1991). ALife recognizes that the study of adaptive behaviour requires more than the study of control mechanisms. Hence the emergence of *synthetic ethology*, the study of artificial worlds and their inhabiting "animats". Tinbergen has laid down the aims of ethology as being the study of causation (i.e., mechanistic control), ontogeny, evolution, and survival value. He also insisted that these aims should not be pursued in isolation of each other. In a full blown and coherent study of behaviour the four aims are integrated and mutually supporting (they are only "heuristically separated").

Clearly, as ethology stands, it completely ignores the experiential aspect of a cognizer's life, thus both leaving out one crucial kind of answer to the question "why does it behave the way it does?" and essentially treating animals as mere objects. Interestingly, in a recent paper Burghardt calls for the introduction of a fifth aim in ethology, namely the study of experience. He argues that experience is interwoven with the other themes of ethology in the same way as they are among themselves, which implies that experience can be studied *through* the other four aims. Also, the explicit introduction of a fifth aim will help prevent the pooling together of mechanism and experience, just as the aims of causation and function led to recognition of confusions between proximate and distal processes. With the fifth aim, ethology returns to its original goal of attempting to "understand the perceptual and inner worlds of organisms, both human and nonhuman" (Burghardt, forthcoming). The early ethologist Von Uexküll used the concept of an *Umwelt*, which is simply the environment *as perceived* and responded to by an organism, or, in other words, the organism's phenomenal world. If such a concept makes sense at all, Burghardt is right in claiming that "experiential descriptions are eventually necessary for the most complete understanding", even in ethology.

5 Summary/Conclusion

We are animals. This means both that we are thoroughly bodily (there are no purely mental phenomena) and that we *live* our worlds before we conceive them, before we set ourselves apart from them. We perceive and act in the same, phenomenal, world. Hence, if we want to study cognition, we'd better investigate phenomenal worlds, their structure, the way they are complementary to and inextricably linked with the organisms that bring them forth, and their evolution. These issues have long been studied in ethology, and so that field should be considered one of the core cognitive sciences. However, ethology does not count 'the mental' as one of its research aims. Yet, recently, Burghardt (Burghardt, forthcoming) has called for the introduction of a fifth aim for ethology: experience. I suggest that this be done by takingwant to up the phenomenological work of Merleau-Ponty and followers, from which we can learn what it is to have a world.

The importance of embodiment is starting to become a widely accepted issue, but usually merely for pragmatic (engineering) reasons. It is not generally appreciated that embodiment can only really be said to be absolutely central on the basis of the claim that mind and body form a unity. Undeniably, another consequence of that same claim is that experience has to be taken seriously, which it is in phenomenology. As a bonus, we can simply say that "content is what is available in experience", which means that we can recognize the affective component of meaning, instead of having to settle for the cold, formalistic meaning of cognitivism. It also means the denial of a principled distinction between cognition and emotion, between the rational and the irrational. We are animals.

The taking serious of embodiment and experience demands the abandonment of representationalism. First, because it is a form of epiphenomenalism. Second, because representationalism is Cartesian: it posits that all it takes to be a mind is to instantiate some representational structure. On the one hand, this denies the importance of embodiment, on the other, it does not do justice to the actual *experiences* we have. We have experiences of being-in-the-world in which we do not carve up the world into objects, properties, and events (e.g, the well-known Heideggerian hammering-experience). Merleau-Ponty makes sense of such experiences with his notion of *situations*.

The only philosophical justifications for representationalism are either the idea that the intentionality of a whole cognizer is to be explained away gradually into the lesser intentionalities of its parts, or the idea that organismal content has to be explained in terms of organal contents. Both these ideas have been argued against. Our underlying machinery is just that, machinery. It understands nothing, because it does not understand at all. Any meaning that we ascribe to it is "as if" meaning, the consequence of an attempt to make sense of it by describing it in informational/computational terms.

Cognition is a matter of the relations between cognizer and world, it is not constituted by processes in the head. Insofar as we investigate processes inside the cognizer, they should be understood as merely *enabling* the cognizer to take its place in a world. Brain processes are not mental processes. In fact, we should not try to objectively localise cognition or meaning at all. Meaning only exists in/as experience. It is to be found in the phenomenal, not the objective, world. (The objective is an abstraction based on the phenomenal. It is a product, not a starting point.) Thus, there is no need to seek the underlying mechanisms of enabling explanations only inside the skin. The insight that the physical system which 'underlies' cognition may change with the situation has led us to embrace a Dynamical Systems Approach.

Finally, there is the question if phenomenology provides ways of studying experience which are different from the ones that are already used in studying the other four aims of ethology. This is not the place to lay down principles for doing phenomenology (but see (Sheets-Johnstone, 1990)). It was pointed out that phenomenology has freed the mind from its Cartesian prison and thrown it out in the world. The mind is flesh; meaning is *in* behaviour, not behind it. A second relevant insight is that living (and understanding) comes *before* conceiving. The conceptual is linguistic, and language is social. We are social, we live *with* others, before we can even start to wonder about other minds. We can try to figure out exactly how this empathic understanding of others comes about (it is thoroughly bodily, (Merleau-Ponty, 1962; Sheets-Johnstone, 1990)). In the meantime there is no use in denying that we have this *skill*. We should use it just like we use other skills in doing science, and realise that it does not give third person knowledge, but rather a "second person understanding".

References

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, *47*, 139–59.

Burghardt, G. (forthcoming). Amending Tinbergen: A fifth aim for ethology. In Mitchell, R., Thompson, N., & Miles, H. L. (Eds.), *Anthropomorphism, Anecdotes, and Animals: The Emperor's New Clothes*. University of Nebraska Press, Omaha.

Cussins, A. (1990). The connectionist construction of concepts. In Boden, M. (Ed.), *The Philosophy of Artificial Intelligence*, Oxford Readings in Philosophy, pp. 368–440. Oxford University Press, Oxford.

Dennett, D. C. (1981). Where Am I?. In Hofstadter, D. R., & Dennett, D. C. (Eds.),*The Mind's I; Fantasies and Reflections on Self and Soul*. Basic Books, New York.

Lakoff, G., & Johnson, M. (1980). *Metaphors We Live By*. University of Chicago Press, Chicago.

Mallin, S. B. (1979). *Merleau-Ponty's Philosophy*. Yale University Press, New Haven and London.

McDowell, J. (1994). The content of perceptual experience. *The Philosophical Quarterly*, *44*, 190–205.

Merleau-Ponty, M. (1962). *Phenomenology of Perception*. Humanities Press, New York.

Rosenfield, I. (1993). *The Strange, Familiar and Forgotten.* Vintage Books, New York.

Sheets-Johnstone, M. (1990). *The Roots of Thinking*. Temple University Press, Philadelphia.

Sheets-Johnstone, M. (1994). What is it like to be a mind? or why a brain is not a body. Manuscript.

van Gelder, T. (1992). What might cognition be if not computation?. CSRP 75, Indiana University.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

Creating a computerized ZPD

Rosemary Luckin
rosel@cogs.susx.ac.uk

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

Abstract Vygotsky's theory on how children learn and how they should be taught has been a catalyst for investigations by those involved in designing Intelligent Tutoring Systems (ITSs). My research aims to continue within this ITS tradition and address the following questions: Can a Zone of Proximal Development (ZPD) be created in an educational computer system through interaction between computer and learner? If implementation is possible, is it useful and can it help to clarify the ZPD concept? This paper discusses practical as well as theoretical considerations with regard to how a ZPD might be created within a an Intelligent Tutoring System.

1 Introduction

If educational computer systems are to play a useful instructional role they need to be integrated into a broader teaching strategy and based upon an efficient theory of instruction (Laurillard, 1993). Intelligent Tutoring Systems; such as those of Tennyson, Christensen and Park (1984) or Chan; Chee and Lim (1992), have used a cognitive learning theory as the basis for the instructional theory which has been implemented. This promotes a symbiotic relationship in which the learning theory provides the design principles for the ITS and the ITS a testbed for the cognitive theory (Anderson, 1984). The theory of instruction which is the focus of this research is that presented by Vygotsky (1978, 1986).

Intelligent Tutoring Systems research has considered Vygotsky's work in a diverse variety of contexts. Derry (1991) in her investigation of metacognitive issues gives considerable attention to Vygotsky. She proposes a design model for a Cognitive Mentorship system called TAPS (Training Arithmetic Problem-solving Skills) and concludes that whilst difficult to implement within an ITS, this cognitive apprenticeship model, attributable to Vygotsky, is potentially powerful. Cognitive apprenticeship is also promoted by Brown, Collins and Duguid in their 1989: "Situated Cognition and the Culture of Learning" and Brown (1990). They suggest that the separation between 'knowing how' and 'knowing what', prevalent in educational practices, is misguided because learning and cognition are "fundamentally situated". Clancey (1992) suggests that cognitive apprenticeship is a "contextualised way of teaching abstractions". However, only Derry (1984) amongst these 'apprenticeshipists' actually references Vygotsky. In the main the importance of his particular theory, and specifically the ZPD has been somewhat superficial.

In a few cases Vygotsky's work has been of greater significance. The EXPLAIN system of Wood, Shadbolt, Reichgelt, Wood and Paskiewitz (1992) extends the scaffolding metaphor (Wood, Bruner and Ross 1976) in the development of their contingent teaching strategy. The system of Gegg-Harrison (1992) offers the student guided problem-solving sessions in which they are given assistance in solving a difficult problem. He uses the concept of the "knowledge zone" to distinguish ITS's that model the student's potential from those that model her actual ability. The system uses schema templates which are seen as "Prolog microworlds" they limit the possible program implementations. The lesson plan for the student is then adapted in line with the student's "knowledge zone." In this manner he suggests that his intelligent tutoring system provides “instruction that is truly adapted to the capability level of its student.” (Gegg-Harrison, 1992)

2 Creating a Computerized ZPD

The current research project involves the design and construction of an ITS which aims to instruct children within a Vygotskian paradigm, and specifically within their individual ZPD, but what does this mean? Vygotsky defined the ZPD as

“The discrepancy between a child’s actual mental age and the level he reaches in solving problems with assistance”

(Vygotsky, 1986 page 187)

It was introduced as part of Vygotsky’s clarification of the relationship between school instruction and a child’s mental development. The work of Shif (Vygotsky, 1986) had highlighted to Vygotsky that the child’s scientific concepts developed ahead of their spontaneous concepts. He noted that the scientific concepts evolve under the conditions of “systematic cooperation between the child and the teacher” (Vygotsky, 1986). It is this systematicity within cooperative instruction which Vygotsky cites as the reason for the earlier development of the scientific concepts. The contact point between the everyday and the scientific is collaboration within the ZPD. Indeed, he states that the creation of the ZPD is an “essential feature of learning” (Vygotsky, 1978). Only if a ZPD is created will the instruction of which it is a part influence the developmental course of the learner.

3 Representing this Knowledge structure in an ITS

The separation of Scientific and Everyday concepts within Vygotsky’s theory is fundamental. He also specifies what he considers to be the structure of the concepts within the learner’s mind and the means by which this structure is built. The concepts of which the child has direct experience, the everyday concepts, are already within her mind, but they have no systematicity. The scientific concepts, of which the child does not have direct experience, are introduced to him during his education at school, he learns about them in a systematic manner. The scientific concepts within the child’s mind reflect this systematicity and represent a hierarchy based upon their level of generalisation. The everyday concepts mediate in the process of the acquisition of the scientific concepts and are gradually integrated into this hierarchical structure. The process is evolutionary and the structure of the concepts within the child’s mind reflects the history of their experiences. The lower levels are still accessible, but the higher levels of generalisation add a ‘depth’ to the concept’s meaning. The words used by an individual to describe a concept may identify the same concept but their semantic weight is different, the more advanced concepts represent a “reconceptualisation of existing knowledge” (Van der Veer and Valsiner, 1991). This notion of concept structure forms the foundations to the ZPD theory. In order to create the ZPD the teacher must introduce concepts which represent the next step up from the everyday concepts already within the child’s mind. For this to be achieved within an ITS one of the system requirements will be a means of representing the domain knowledge and the child’s current understanding of that domain which acknowledges this hierarchical structure of concepts. Within this structure the everyday concepts need to form the foundations upon which the introduction of the scientific must be built. How might this be represented computationally?

Goldstein’s Genetic Graph (Goldstein, 1982) presents a representation for the evolution of a learner’s knowledge within the domain of a computerized adventure game - WUSOR II. Goldstein describes rules (or groups of rules with the same goal) as nodes in a Genetic Graph and declarative facts as explaining and justifying the behaviour/s of these rules. There are links between the nodes of the graph which represent possible transformations one of these links is a generalisation/specialisation link. Vygotsky suggests that a concept has a position in a ”Generalisation structure” in which the position of the concept within the structure defines the operations that are possible with it. In the Genetic Graph the student is modelled in terms of their "possession of the conceptual base underlying a rule set" and their movement to a new level is considered to be a movement to a "new conceptual base". The evolutionary nature of the Genetic Graph allows the student to be modelled as an overlay of the domain representation, rather than a subset of an 'expert's' knowledge. This means the learner is being modelled in terms of the evolution of knowledge and skills as represented in the structure of the Genetic Graph instead of upon the final set of skills and knowledge thought to be used by an expert. The earlier knowledge (everyday concepts) is not replaced by the latter more advanced scientific knowledge. The new higher level concepts evolve out of the lower. The Genetic Graph captures a means of representing the evolutionary nature of the learner's increased knowledge and understanding. Their previous knowledge is not over-ridden by their new knowledge but is still available. This is all in line with the Vygotskian notion of a generalisation structure and the evolution of this within the mind of the learner.

If the Genetic Graph is combined with an object oriented design methodology, objects, as opposed to rules, can be represented in a hierarchical manner in which the higher levels represent generalisations of the lower ones. The Combination of the two strategies produces a Genetic Graph type structure in which the nodes are the world objects of the Object Oriented design methodology and the links are those of generalisation/specialisation only. The frontier of the graph provides a means of representing the learner's current level of understanding, the level which she can achieve without assistance. A higher level in the hierarchy will represent that which the learner might be able to achieve with assistance. For some learners this higher level will be the closest possible next level of generalization within the Genetic Graph structure. For others it may lie several levels higher. In all cases the distance between the lower and higher levels within the graph provide an initial crude ZPD representation. The tasks which the system requires the learner to complete will involve the concepts which lie between the two levels of the Genetic Graph - within their ZPD. The amount of assistance which each learner requires to achieve the tasks set can be used to amend the system's ZPD representation and ensure that the dynamic nature of the ZPD is fully recognised.

4 Conclusion

This project will extend the area of ITS research which addresses the work of Vygotsky. It is particularly concerned with the clarification of the ZPD and the plausibility of its creation through interaction between computer and learner. A possible means of representing knowledge, both of system and learner, has been presented. This object oriented adaptation of Goldstein's Genetic Graph maintains the structure of knowledge as proposed by Vygotsky, and allows for the identification and maintenance of a dynamic ZPD for each learner.

References

Anderson J. R. (1984): *Cognitive psychology and intelligent tutoring* Proceedings of the Cognitive Science Society Conference, Boulder, Colorado. pp. 37–43.

Brown J.S (1990): *Toward a New Epistemology for Learning in Intelligent Tutoring Systems: at the Crossroads of Artificial Intelligence and Education.* Frasson C and Gauthier G (Eds.) Ablex Publishing Corp. Norwood. New Jersey

Brown J.S.; Collins A.; and Duguid P. (1989): *Situated Cognition and the Culture of Learning.* In Educational Researcher. Jan-Feb 1989

Chan T.; Chee Y. S. and Lim E.L. (1992): *COGNITIO: An Extended Computational Theory of Cognition.* In Frasson, Gauthier and McCalla (Eds.), 2nd International Conference of ITS 1992. Montreal, Canada. Springer-Verlag, Berlin.

Clancey W (1992): *Representations of Knowing: In Defense of Cognitive Apprenticeship.* In The Journal of Artificial Intelligence in Education 1992 Volume 3 Number 2

Derry S. J. (1991) *MetaCognitive Models of Learning and Instructional Systems Design*. In Foundations and Frontiers in Instructional computing Systems - Winne P.H. and Jones M. (Eds.).

Gegg-Harrison T. S. (1992): *Adapting Instruction to the Student's Capabilities*. In Journal of Artificial Education in Education. Vol 3, pp 169–181. Association for the Advancement of Computing in Education Press

Goldstein I.P (1982)*The Genetic Graph - a representation for the evolution of procedural knowledge*. In D. Sleeman and J.S. Brown (Eds.) Intelligent Tutoring Systems - New York: Academic Press

Laurillard D. (1993) *Rethinking University Teaching: a framework for the use of educational technology*. Routledge, London

Tennyson R.D., Christensen D.L. and Park S.I. (1984) *The Minnesota Adaptive Instructional System: An Intelligent CBI system.* In Journal of Computer Based Instruction 11, 1, pp 2–13

Vygotsky L.S. (1978) :*Mind in Society: The Development of Higher Psychological Processes*. Cole M.; John-Steiner V.; Scribner —S.; Souberman E. —(Eds.). Harvard University Press, Cambridge, Mass.

Vygotsky L.S. (1986) *Thought and Language*. Kozulin A. (Ed.) The M.I.T. press, Cambridge, Mass.

Wood D., Bruner J S and Ross G (1976): *The Role Of Tutoring in Problem Solving*. In Journal of Child Psychology and Psychiatry Vol 17, pp 89–100

Wood D, Shadbolt N, Reichgelt H, Wood H. and Paskiewitz T. (1993) : *EXPLAIN: Experiments in Planning and Instruction*. Dept of Psychology, University of Nottingham

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

Creativity in Writing

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

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

1 Introduction

The goal of this research is to establish a computational model of the creative process in arts in terms of Engaged and Reflective states, in particular in the field of writing. The Engaged state can be described as a state in which the artist is involved intensely in the production of material related to his/her task, e.g. to build a piece of music, to write a story, etc. Such production of material is guided by tacit constraints e.g., cultural background, or the use of a specific style or technique. Thus, although everybody can experience such states, in the case of artists some of these tacit constraints are the result of years of experience, the development of certain skills, and the training in specific areas which allow them to produce works of art which are difficult to imagine being produced by the layman. The Reflective state can be described as a state where the artist analyses, and/or evaluates his/her work; also the artist deliberately explores and transforms a conceptual space (Boden, 1991). During this state, therefore, artists are helped by explicit and tacit knowledge, such as music theory, writing techniques, etc. which they have learned during their training and experience as artists.

2 Hypothesis and Research Questions

The main hypothesis of this research is that both states participate actively during the creative process although during this process people can have an orientation towards one or other of them.

The production of material during the Engaged state is driven by intentions and relies on associations between previous experiences; thus, some of these previous experiences are used as a framework for the production of new material. In the Reflective state, the person evaluates the material produced and generates guidelines for the Engaged state.

Some of the questions I shall consider for my research are:-

- What is the relation between Engaged and Reflective states?
- What is the importance of the rhythm between Engaged and Reflective states?
- Are these two states enough to provide a basis for an in-depth analysis on creativity?
- Is it possible to develop a computational model of creativity which incorporates Engaged-Reflective states?
- How important is personal experience and cultural background for the Engaged-Reflective states?
- What is the role and importance of previous experiences as a framework for the production of material in the Reflective state?
- What is the importance and role of intentions in the creative process?

3 Antecedents

Although I have not found any approach which can be used as a starting point for the development of a computational model of creativity in terms of Engaged-Reflective states, different authors have talked about similar ideas for many years. Some of these authors have pointed out how creative people have orientation towards one or other of these states. For example, Aarond Copland mentions different kinds of composers;

> "The type that has fired public imagination most is that of the spontaneously inspired composer -the Franz Schubert type, in other words…this type [of composer] is more spontaneously inspired. Music simply wells out of him…" (Copland, 1955, p.22)

The second type is called by Copland "The constructive type"; this type starts with a musical idea or theme, and then s/he analyses and plans how to work with it

In Beethoven's case there is no doubt about it, for we have the notebooks in which he put the themes down. We can see from his notebooks how he worked over these themes, how he would not let them be until they were as perfect as he could make them. Beethoven was not a spontaneously inspired composer in the Schubert sense at all. He was the type that begins with a theme; make it a germinal idea; and upon that constructs a musical work, day after day, in painstaking fashion (Copland, 1955, p.22)

Spender (Spender, 1946, p.115) talks about two kinds of poets: one who can write immediately and his/her works scarcely need revision. The other type writes in stages, draft after draft, until finally, after many revisions s/he produces something which has little connection to the original sketch. Spender refers to these two kind of poets as Mozartians and Beethovenians:

The difference between two types of genius is that one type (the Mozartian) is able to plunge the greatest depths of his own experience by the tremendous effort of a moment, the other (the Beethovenian) must dig deeper and deeper into his consciousness, layer by layer. (Spender, 1946, p.116)

Wellek and Warren (Wellek & Warren, 1970, p.86) define two kinds of writers, the "possessed" and the "maker":

There is certainly a typological pair of the 'possessed', i.e. the automatic or obsessive or prophetic poet, and the 'maker', the writer who is primarily a trained, skilful, responsible craftsman. This distinction seems partly historical: the 'possessed' is the primitive poet, the shaman; the Romantic, the Expressionist, the Surrealist, we say. The professional poets, trained in the bardic schools of Ireland and Iceland, the poets of the Renaissance and neoclassicism, are 'makers'. But of course these types must be understood as not mutually exclusive but polar.… we have to think of the writers as both 'maker' and 'possessed', as combining an obsessively held vision of life with a conscious, precise care for the presentation of that vision. (Wellek & Warren, 1970, pp.84,85)

James Hartley and Alan Branthwaite (Hartley & Branthwaite, 1989) talk about two kind of writers: the 'Thinkers' who "spent more time thinking about what they were doing" and the 'Doers' who "seemed to think less about what they were doing". (Chandler, 1992) has defined two types of writers, The Planners and The Discoverers:-

Planners tend to think of writing primarily as a means of recording or communicating ideas which they already have clear in their minds;

Discoverers tend to experience writing primarily as a way of 'discovering' what they want to say.

The closest approach I found for a computational model of creativity in terms of Engaged and Reflective states was given by (Gelernter, 1994). He says that the way in which a person thinks depend on a characteristic called "mental focus". Focus can be high or low or medium. When one is at the high end the thought is analytic and penetrating. "As we set off down-spectrum, thinking becomes less penetrating and more diffuse, consciousness gradually 'spreads out' and…emotions starts gradually to replace logical problem-solving as the glue of thought" (Gelernter, 1994, p.5). However, when he describes his computational model, he just talks about the high focus state and in a very general and some times ambiguous way; he never makes any description of the medium or low focus state and the way they are related. This is strange since he complains that "Almost all attempts to simulate thought on a computer have dealt exclusively with this narrow, high-focus band at the top of the spectrum" (Gelernter, 1994, p.5).

Regarding to the relationship between Engaged and Reflective states, Wason has commented that "writing is difficult for some people because they try to do two incompatible things at the same time: say something, and say it in the most acceptable way." (Wason, 1980); the same idea that composition and transcription can interfere with each other has been expressed by (Smith, 1982, p.21). Wason also has talked about the way Reflective state can influence the production of material: "The process of criticism becomes creative. It does not merely refashion language, it enlarges thought. Scrutiny of language alters the thought expressed in language" (Op.Cit.)

With regard to intentions, (Smith, 1982) has noticed how they lead the production of material during the creative process. He explains it in the following phrase:

> "This word, this sentence that I am writing now was not a concern of mine when I began this book, or even the present chapter. My global intentions have brought me to the point where I am now, and my global intentions will, I hope, carry me to the end." (Smith, 1982, p.90)

Finally, different authors have pointed out the importance of experience during the creative process. Robert Weisberg (Weisberg, 1992) gives a very good description of the role and importance that previous experiences played in the work of very famous creative people like Darwin, Picasso, etc. and Larry Austin and Thomas Clark (Austin & Clark, 1989) talk about the same idea in composing music:

It is important to learn from composers of the past, to study their work, accepting the challenge of their best works to be masterful, original, and to strive for beauty in our own new works. New, artful compositions grow out of the past but are not obeisant to it. In this healthy relationship to the past, composers have traditionally been students of their art, not just during their novice period as intern composers but throughout their composing careers - always probing, always learning, always listening, intimidated only by the fear that they may not yet know enough. (p. 2)

4 Contributions

The contributions of my research are:-

- It attempts to be one of the first models which involves Engaged and Reflective states.
- The model attempts to highlight the importance of the interaction between Reflective and Engaged states.
- The model attempts to highlight the way previous experiences influence the creative process.
- The model attempts to highlight the way we produce material during the Engaged state.
- It attempts to be one of the first systems which evaluates its own output.

References

Austin, L., & Clark, T. (1989). *Learning to Compose.* Wm. C. Brown Publishers.

Boden, M. A. (1991). *The Creative Mind.* Basic Books, New York.

Chandler, D. (1992). The phenomenology of writing by hand. *Intelligent Tutoring Media*, *3*(2/3), 65–74.

Copland, A. (1955). *What to listen for in Music*. New American Library, New York.

Gelernter, D. (1994). *The Muse in the Machine*. Fourth Estate, London.

Hartley, J., & Branthwaite, A. (1989). The psychologist as wordsmith: a questionnaire study of the writing strategies of productive british psychologists. *Higher Education*, *18*, 423–452.

Smith, F. (1982). *Writing and the Writer*. LEA Publisher, London.

Spender, S. (1946). The making of a poem. In Ghiselin (Ed.), *The Creative Process*. University of California Press, Berkeley.

Wason, P. C. (1980). Specific thoughts on the writing process. In Gregg, L. E., & Steinberg, E. R. (Eds.), *Cognitive Processes in Writing*. LEA Publishers.

Weisberg, R. W. (1992). *Creativity: Beyond the myth of genius*. W. H. Freeman and Company.

Wellek, R., & Warren, A. (1970). *The Theory of Literature* (3rd edition). Harvest, New York.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

Explicit Context and Information Discovery Systems

Christopher D Staff
chrisst@cogs.susx.ac.uk and cstaff@unimt.mt

School of Cognitive and Computing Sciences
University of Sussex
Falmer, Brighton
England
Department of Computer Science and Artificial Intelligence
University of Malta
Msida MDS 06
Malta

1 Introduction

Information Discovery Systems (e.g., hypertext and information retrieval (IR) systems) are used to help their users discover the information they are looking for. However, more often than not, the user is not given any assistance with expressing the information need. The user is instead confronted with information that the system has calculated to be relevant to a query (in IR), or which the author of a hypertext has a priori decided is relevant to the information contained in the current node.

This paper is a speculative argument as to why this is not enough user support and what extensions to hypertext and information retrieval systems are required to support the information discovery process.

2 The Structure of an Information Discovery System

Three modules comprise an information discovery system (IDS). The system needs an information base, a representation of the information content and/or organisation, and an interface through which the user can interact with the system.

An IR system usually has a collection of information-containing documents, a representation of the information content (e.g., words or terms) of all documents, together with probabilistic weights of the terms (to attempt to capture the "aboutness" of the document, and an interface through which a user can submit a query and receive results. The information discovery process is led by the user by encapsulating the information need in a query. The IR system calculates which documents are probably "about" the query, and it is left to the user to select which documents are really relevant to the information need. If the information need has not been satisfied, then the user can refine the query and cycle through the process once again.

A hypertext system also has a collection of information-containing documents (called nodes), but this time the representation reflects the organisation of and relationships between the nodes. A link between nodes is a reflection of some association between the information in the nodes. The interface is used to enable users to browse through the collection of nodes, and to move from one to another by traversing links.

Although the information discovery process is different in the different systems (it is query-led in IR and link traversal-led in hypertext), there is an important underlying similarity between the two. A hypertext system can be regarded as a special case of IR system. In IR, the emphasis is on ad-hoc querying, but in hypertext a link from one node to another can be seen as the embodiment of a prior query (formulated by the author or designer of the hypertext) and its result (selected by the author/designer as the "best" node of all the nodes that could possibly have been relevant to the former node). This leads to a generalisation about hypertexts and information retrieval systems: that in order to determine how accessible a node or document should be, it is necessary to anticipate all the likely circumstances in which a user will want to or need to access the node or document. As relevance is measured in terms of a user's information need, this means that the location of a node in a hypertext or the indexing of a document in an IR system should anticipate users' information needs. This generalisation leads to the crux of the information discovery problem. It is not possible to anticipate all users' information needs. The best that can be achieved is a flexible approximation of the information needs.

3 The Information Need

A characteristic of an information need is that it is vague, imprecise and incomplete. We are not talking about experts who know precisely what information they are looking for, but have temporarily forgotten the title of the paper that contains it. The users that need most assistance from an IDS are those who use the IDS because they cannot describe the characteristics of the information for which they are searching. In an IR system such a user might try a tentative query with one or two terms in it to see what the response will be. If the response is overwhelming, the user might browse through a few documents to see if they contain anything which might help refine the query. In a hypertext system, the user will try to identify a reasonable starting node and will then follow links in the hope that they will lead to the information they are looking for.

4 Context

Something that is generally overlooked in IDSs is that although users might be incapable of expressing their information need, they usually know why they want the information. Additionally, information authors (of any media type) can anticipate some of the reasons why somebody will want to access their work. This is one of the important concepts in the definition of context. Information exists for a reason, and what the context tries to capture is the suitability, or fitness for purpose, of the information in relation to the context of the query need. The context does not replace the information-content representation of a document or node. Rather, it "wraps around" the representation, but an information discovery system should be robust enough to withstand the absence of context information. In an IR system this would be tantamount to alerting the user that some document exists which shares features with other information-in-context that meets the requirements of the query-context, but no information is available about the context of this particular document. It is then up to the user to determine the applicability of the document to the context of the information need.

A context is a particular perspective of a document, and a document can belong to several contexts. It is not necessarily the case that the relative importance of terms in a document will be the same, from one context to another. It also cannot be assumed that the context can be made explicit by the user, nor that the contextual ontology will be fine-grained enough to cater for all possible interpretations under all possible circumstances. What is more likely to be the case is that the IDS will need to infer the appropriate context according to user behaviour of link traversals, in hypertext systems, query refinement in IR systems, and combinations of query refinement and link traversals in combined Hypertext Information Systems (HIR).

A combined Hypertext Information Retrieval system is likely to be used as follows. A user may start a session with a query. The HIR will identify a node, or nodes, which can probably satisfy the query in as much as the user expressed it. The user will then select the apparently most appropriate node and, after assimilating the content of the node, will follow links to other nodes (where links exist). In the event that the user feels that the author-defined path is not likely to lead to a satisfaction of the information need (or if there are no out-going links from the current node), the user may use additional information to refine the original query in the hope that the HIR system can determine which node or document should next be presented to the user.

In a combined information retrieval and hypertext system, all of the documents should have an information-content representation. However, it is not necessarily the case that all the documents will be organised into a hypertext structure, or that there will be only one hypertext structure. In a massive hypertext system, it is unlikely that an author will have all-encompassing knowledge of the collection of documents, so that when the author creates a new node there is a guarantee that the node will be linked to all other associated nodes and vice versa. What is more likely to happen is that the author will identify the most appropriate place to locate the node, given the author's knowledge of some local topological area. However, if the author explicitly identifies as many possible contexts to which the information in the node is applicable, then, in a flexible information discovery system, the node will eventually be connected to other applicable nodes according to usage.

I mentioned earlier that only a flexible approximation of the information need is possible. This flexibility is usually missing from IDSs. Information is a shared resource, yet even though links are information, if a user decides to link two nodes, the link is private and unsharable even though the link could be useful to somebody else embarking on a similar information discovery process. In an IR system, two users who are looking for the same information for the same reasons have to separately design their own information discovery strategy. Some IR systems are flexible in that relevance judgements can be used to alter the probabilistic term weights, but there is usually no opportunity for the terms which describe the information content to be changed according to perspective, or context. Explicit contexts are intended to be flexible and dynamic, to reflect actual usage.

5 The Dynamicity of Contexts

Contexts are dynamic features of information. Unlike term features, which are generally static and must be identified in advance of a document being retrievable, an author will not have to specify all the contexts to which a document is applicable. Documents can still be retrieved by a user according to their information-content representation. The user can use relevance judgements to determine if the current context should be extended to retrieved documents which are not explicitly applicable to the context. If the user determines that it should be applicable, then the current context is captured by the system and recorded with the context information of the document.

As relevance judgements (given by authors and users) are subjective, contexts are probabilistic, rather than binary in value. When an author applies a context to a document, a weighting-factor is used to capture the likelihood that the document is applicable to each specified context. When a user causes a new context to be applied to a document, initially the weighting-factor will be low, and it will increase or decrease with subsequent usage.

The IDS is instrumental in determining the context in which a user is viewing a document or node. The objects that are used to determine the context are the previous nodes visited and links traversed, in hypertext. A link connecting two nodes is typed according to context and probability of relevance. If two nodes are relevant to each other in several contexts, then there will be a separate link for each context. In IR, where a list of relevant documents is returned to a user in response to a query, the system should not only take into account the documents selected by the user as being relevant, but also the context and features of those documents that the user considered irrelevant, when calculating which documents to return to the user in response to a refined query. This task is extended over all queries submitted in the current context session.

6 Contextual Proximity

Another feature of contexts is that they are not necessarily mutually exclusive. In all likelihood, contexts can be related to each other to some degree or another. This is called contextual proximity. If contexts can be structured as a network, then node distance between contexts is an indicator of contextual proximity. It is not sufficient to determine if the context of two nodes is identical. It could be that given two nodes which are linked, the second node's context could be a generalisation or specialisation of the first node's context. Together with weighting-factors on the applicability of information to a context, this means that determining the current context is a complex task.

7 Dynamic Links

In a combined hypertext and information retrieval system, the context can spread over the browsing and querying activities. However, there is an opportunity for the system to automatically modify the hypertext structure if a user submits a query after browsing through part of a hypertext. This behaviour could very well indicate that the result of the query should be accessible via a link from some node in the path that the user had followed. The system may, with appropriate support from the user, create a dynamic link from the "best" document or node in the query result to one of the nodes that the user has visited in the current context session. A dynamic link can have other properties. It may be that the query does not yield satisfactory results. In this case a dummy representation can be created for the "ideal" document and if such a document matching the representation is created in the future, then the user can be notified.

In current hypertextual systems, the number of out-going links from a node is limited to prevent "information overload". If a user is presented with a node which has links to several other nodes, the user might be tempted to visit all the other nodes, just in case they contain relevant information. The use of dynamic links could lead to a large increase in the number of out-going links from a node, as different users search for different information, via queries, the results of which are potentially relevant to the node. The context of links and the nodes to which they are connected can be used to reduce the effect. Once the context in which a user is visiting a node has been determined, all out-going links which do not correspond to the context can be suppressed, giving the user the impression that there are only a few links. This could also reduce the "lost in space" syndrome. Users will have less opportunity to inadvertently move outside the context in which they wish to seek information.

8 Context Switching

It is important that a user can override the system, and perform a "context switch" - an action which causes an abrupt change to the context in which they wish to look for information. This is equivalent to starting a new context session, but from the current node (in hypertext) or from the current document set (in IR). The user must be able to instruct the system to abandon the context profile that it has been constructing. In hypertext, this may be equivalent to the user instructing the system to show all links from the current node, regardless of context, and then following one of the links. In information retrieval, some other method will need to be identified with which to flush the context.

9 Further Issues

There are several issues which will need to be tackled if contexts can be used to assist in the information discovery process. An ontology for, and structured representation of, contexts will be needed in order to select contexts from a common representation and to determine contextual proximity. There needs to be a user-friendly method of allowing an author or user to express the degree of applicability of a document or node to a context. The context representation needs to be portable, so that it can be used to wrap around any information representation. A formal definition of context is needed to differentiate the context of information from the syntax and semantics of information. A formal framework and operational semantics is required to enable contexts to be manipulated and reasoned about.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

Pattern Recognition Analysis of Tumour *in vivo* MR Spectra

A. Rosemary Tate[1] **and John R. Griffiths**[2] **and Des Watson**[1]
rosemary@cogs.susx.ac.uk

[1] **School of Cognitive & Computing Sciences**
University of Sussex, Falmer
, Brighton, BN1 9QH

[2] **CRC Biomedical MR Research Group,**
St George's Hospital Medical School
London SW17 0RE

Abstract The ability to classify spectra of tumours according to their stage and type will be essential if magnetic resonance spectroscopy (MRS) is to be used as an aid in the diagnosis of cancer. MRS data are normally classified on the basis of selected peak measurements but these may be difficult to extract automatically. We present two alternative methods of feature extraction which we used to discriminate between spectra from tumours and normal tissues. Discrimination could be achieved either using features from the whole spectrum, or from a selected region containing the peaks from the phospholipid precursors in the phosphomonoester region.

1 Introduction

Magnetic Resonance Spectroscopy (MRS) provides a non-invasive method for observing the biochemical processes *in vivo* and thus has great potential for the diagnosis and study of cancer and for monitoring cancer therapy.

MRS, already used extensively to study cancer biochemistry (Negendank, 1992) , also has great clinical potential, not least for the non-invasive diagnosis or grading of tumours located in regions that make biopsy difficult. Reliable methods for analysing and classifying the data will be needed if MRS is to become a practical clinical tool. Most current methods for analysing *in vivo* MRS data are based on eliciting the relative quantities of specific biochemical compounds of interest from measurement of selected peaks in the spectra. This quantification may be essential for ascertaining the biochemical structure of the tissue but may not be appropriate for clinical applications when we need more abstract information about the tissue being examined, such as its disease type, or degree of malignancy. Furthermore, it presupposes that we know which compounds are of interest.

Here we discuss the development of automated methods for eliciting such general information using pattern recognition analysis. This is an approach which encompasses a wide range of techniques for finding patterns in groups of data, and which can be used to distinguish between different subgroups. Automated pattern recognition analysis methods can greatly enhance the information available from MR spectra of tumours, whether obtained *in vivo* or *in vitro*. At present, ^{31}P spectroscopy is the only form of MRS that is routinely used to study tumours in sites other than the brain or prostate. Surprisingly, although they contain peaks from only a few metabolites even ^{31}P MR spectra obtained *in vivo* from tumours or normal tissues of rats contain enough information to provide a considerable degree of classification. Examination of the data sets revealed that the phosphomonoester (PME) region is important for classification, and that some groups could be classified successfully on the basis of this region alone. It is not normally possible to quantify the separate components of this region in spectra obtained *in vivo*. However, our programs have been able to distinguish their contributions from changes in the shapes of the combined peak. We also found that certain minor peaks in the PME and Pi regions, that are occasionally seen in ^{31}P MRS spectra of living tissues but are not normally assigned, were important in the classification.

2 Subjects and Methods

Three tumour types were studied: Morris hepatoma 7777 (fast-growing, poorly differentiated) and Morris hepatoma 9618a (slow-growing, well differentiated) were grown in female Buffalo rats; Walker 256 carcinosarcomas were grown in female Wistar rats; and GH3 prolactinomas ware grown in female Wistar-Furth rats. In all cases, tumours were implanted subcutaneously into the flank, and generally grew to a size suitable for spectroscopy within 2-3 weeks. The animals used for studies on normal tissues were male Wistar rats.

Spectra from two classes of normal tissue (10 livers and 10 brains) and four tumours (10 h9618a hepatomas, 13 Walker carcinomas, 4 h7777 hepatomas and 8 GH3 prolactinomas) were obtained from anaesthetised rats (i.p. Sagatal) in a SISCO 4.7T spectrometer using ISIS localisation (volumes 0.22–1.0cm^3, chosen after examination of ^{1}H images) with a 25mm surface coil. The spectra were filtered with a line broadening of 10Hz.

Pre-processing necessary to make the spectra compatible for pattern recognition analysis was kept to a minimum. Each spectrum was automatically phased and then normalised by summing the squares of all the values in the spectrum and dividing each value by the square root of this sum. Co-registration was carried out by aligning the β peak of NTP in each study, simultaneously reducing the number of datapoints to 512, which covered the region containing all the main peaks. The initial stage of the statistical analysis of the spectra was carried out using these 512 values. The second stage, in which we investigate the lipid metabolite peaks, involved further processing the data by extracting the 32 datapoints indexed from ppm 7.6–5.73 from each spectrum, normalising these 32 points as above, and transforming them using a wavelet transform.

2.1 Pattern recognition methods

2.1.1 Statistical Techniques

The aim of this phase of the study was to see whether pairs of tumour and tissue types could be discriminated and to identify the important discriminating features in the spectra.

Linear discriminant analysis is a statistical 'supervised learning' technique which produces linear functions from the variables of known cases which can be used to predict the class of cases (i.e. subjects) whose class membership is unknown. The discriminant functions are calculated so as to make the separation between the populations as large as possible by minimizing differences within a group and maximizing them between groups. A 'training' set of individuals of known class is used to develop the discriminant function and a 'test' set of individuals of unknown class can be used to evaluate how well the functions perform. We used the 'leave one out' method for assessing the success of the discriminant functions, which entails using all the cases, except one, as the training set, and then using the excluded case as the test set, repeating this process until the whole set has been tested. A similar method can also be used for determining which variables give the best classifications, but in this case the discriminating variables are dropped and added in turn to the analysis (Everitt & Dunn, 1991).

The number of variables that can be used for discriminant analysis should ideally be no more than the number of cases in the smallest group divided by three, so a major task of the analysis was to reduce the number of variables (512 datapoints and 32 wavelet coefficients). to two or three, depending on which groups were to be discriminated. These variables were selected on the basis of their correlation coefficient with the tissue type of the spectra.

Correlation coefficients take values between -1 and $+1$, with -1 or $+1$ representing a perfect linear association (negative or positive respectively) and 0 none, and give a measure of the strength of the relationship between pairs of variables.

The package SPSS was used for the statistical analysis (SPSS Inc., 1987).

2.1.2 Wavelet transform

This transform provides a way of modelling the shapes of the peaks and can be used to classify spectra in which the spectral shapes differ (Tate, Watson, & Eglen, 1994).

The Discrete Wavelet Transform (DWT) transforms a data vector of length *n* into another vector of *n* wavelet coefficients using a set of basis functions called wavelets. Each wavelet is a dilation and translation of a single function called the mother wavelet. The DWT is similar to the Fast Fourier Transform (FFT) in some respects but, unlike the sine and cosine basis functions of the FFT, wavelets are localised in space as well as scale. This localisation property allows the relevant information of MR spectra to be compactly represented using a small number of wavelet coefficients. In this study we used the wavelet transform to extract features for classification from the phosphomonoester region which is composed of a number of overlapping peaks which are difficult to identify separately, even *in vitro*.

3 Results

3.1 Datapoints

Each pair of tissue types was analysed in turn, and correlation coefficients were calculated between tissue type and each of the 512 spectrum values.

For each pair, some spectrum values had a highly significant correlation with spectrum class ($p<0.01$). These were typically in groups of five or six adjacent values, and coincided with the positions of the peaks in the spectra, as would be expected. The value with the highest correlation (with tissue type) from each of the groups of highly correlated values was selected and entered into the discrimination program.

For some groups of correlated values, i.e. peak regions in the spectra, the correlations themselves had two or more 'peaks' and in this case the value with the highest correlation from each subgroup was chosen. This reduced the number of variables from 512 to between two and seven depending on the tissue types.

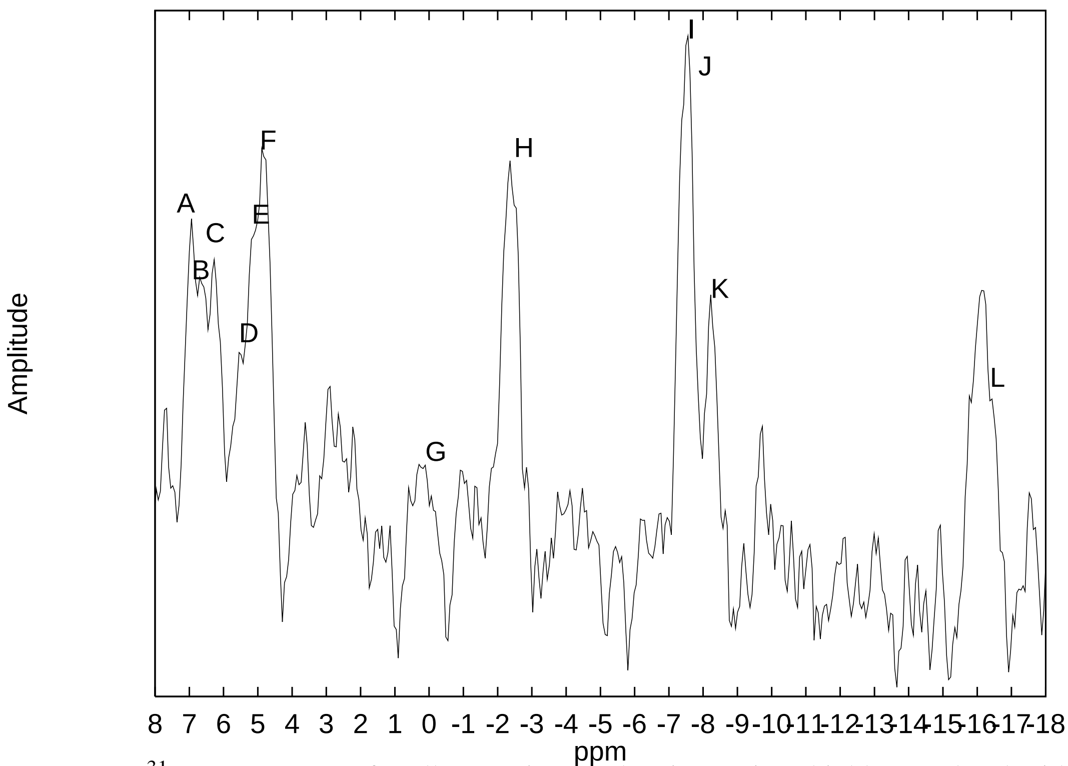

Figure 1: *In vivo* ^{31}P rat spectrum of Walker carcinoma showing regions highly correlated with class, assigned as follows: A, B, C (PME), D, E (unassigned), F(P_i), G(PCr), H(γATP), I,J(αNTP), K(NAD) and L(βNTP).

Figure 1 shows the location of these values on a typical spectrum and shows that most of the signifi-

Class	No. correct	Peaks used
h9618a & GH3	20/20 (100%)	D H I
h9618a & Walker	22/23 (96%)	D I L
h9618a & h7777	13/14 (93%)	D G L
GH3 & Walker	18/21 (86%)	H K
liver & tumours	41/45 (91%)	B E J
liver & hepatomas	24/24 (100%)	E A
brain & tumours	45/45 (100%)	B C G

Table II: Classification results when highly correlated datapoint values (from the labelled peak regions) were used in the discriminant program

Class	No. correct
h9618a & GH3	19/20 (95%)
h9618a & Walker	21/23 (91%)
h9618a & h7777	13/14 (93%)
GH3 & Walker	-
liver & tumours	43/45 (96%)
liver & hepatomas	20/24 (83%)
brain & tumours	40/45 (89%)

Table III: Classification results when wavelet coefficients from PME region of the spectra were used in the discriminant program

cantly correlated values correspond to peaks in the spectrum.

Table I summarises the results from discriminant analysis when the leave-one-out method was used to create the test set. Despite the small number of values that could be used in the program, at least 86% of the spectra were assigned correctly for each pair of tissue types. The method used for selecting variables was a form of peak selection. However, unlike the usual method of peak identification and selection, our method used the differences between groups of spectra to identify the important datapoints. We have included the results of the pair h9618a & h7777 hepatomas for completeness although we are aware that their discrimination could be by chance, due to the small number of h7777 hepatomas.

3.2 Wavelet Coefficients

For this analysis, 32 points from the PME region (peaks A–C, ppm 7.6–5.73) were selected and wavelet transformed. Correlation coefficients were calculated for each wavelet coefficient value with spectrum class. For most pairs of tissue types a few wavelet coefficients were significantly correlated with class. These were selected for use in the discrimination program. Table II shows the results. Apart from GH3 & Walker, very good results were obtained (at least 83% correct) for each pair of tissue types, showing that they can be discriminated using only the PME region on the basis of peak shape. Examination of this region by eye showed at least three clear peaks in many of the spectra, but their positions shifted quite considerably from spectrum to spectrum (by as much as 0.2 ppm), perhaps because of differences in pH or in Mg2+ content. An advantage of wavelet transformation is that the wavelet coefficients may not be so dependent on these positions as the original datapoints.

4 Discussion

In this study the raw data were interpreted statistically without making any prior judgements about the data sets, with the exception of class origin. Only then were the results interpreted with biochemical knowledge. The results shown above, together with previous work , (Tate et al., 1994), (Howells, Maxwell, Peet, & Griffiths, 1992) indicate that pattern recognition is a very promising approach to the interpretation and classification of MRS spectra. We could discriminate between the pairs of tissue types with a success rate of at least 86% and regions of importance, consisting of a few raw data points, could be clearly identified – a fact we found surprising.

The wavelet transform has the advantage that it encodes information concerned with the shape of spectra. Highly correlated wavelet coefficients from the PME region from some pairs of tissue types produced better results than the original datapoints from the same region. This suggests that the ability of the wavelet transformation to discriminate on the basis of peak shape has allowed it to distinguish groups of spectra in which the components of the combined PME peak differ, even though the spectroscopic method used was incapable of resolving them. These results suggest that lipid precursor signals contain important information for automated cancer diagnosis, with only minimal pre-processing. Furthermore they pinpointed regions of the spectra known to be of biochemical interest.

If the discrimination achieved in this study could be repeated on patients, it could have a clinical application for non-invasive diagnosis or grading of tumours. At present, diagnosis and grading are decided upon mainly by subjective (and labour-intensive) histological examinations of biopsies. If the same information could be obtained from MRS it would obviate the necessity of a biopsy, an important consideration if the tumour is in an inaccessible location, e.g. the brain. It would be even more useful, in the long run, if we could use MRS as an aid in the prediction of the response of a tumour to therapy. For instance, only about 25–40% of cancers respond positively to any single chemotherapeutic agent, so 60–75% of patients are given large doses of one or more drugs to no good purpose, leading to unnecessary morbidity and even mortality, and great financial expense. Several studies on ^{31}P MRS of tumours in patients (reviewed in (Negendank, 1992)) have demonstrated that the PME peak(s) are promising indicators for monitoring the success or failure of chemotherapy, and a multi-centre trial of this is now in progress. The fact that we could discriminate between one tumour type and another was very encouraging, not only because this would be useful in its own right, but also because it indicates that it may be feasible to extend these methods to discriminate between more than two groups. For this to be possible, however, we would need much larger data sets.

5 Acknowledgements

JRG thanks the Cancer Research Campaign for financial support under Programme Grant SP1971 0402.

References

Everitt, B. S., & Dunn, G. (1991). *Applied Multivariate Data Analysis*. Edward Arnold.

Howells, S. L., Maxwell, R. J., Peet, A. C., & Griffiths, J. R. (1992). An investigation of tumor ^{1}H nuclear magnetic resonance spectra by the application of chemometric techniques. *Magnetic Resonance in Medicine*, *28*, 214–236.

Negendank, W. (1992). Studies of human tumours by MRS: A review. *NMR in Biomedicine*, *5*, 303–324.

SPSS Inc. (1987). *SPSSX User's Guide* (3rd edition). SPSS Inc., Chicago.

Tate, R., Watson, D., & Eglen, S. (1994). Using wavelets for classifying human *in vivo* magnetic resonance spectra. In *Proceedings of the XVth Franco-Belgian Meeting of Statisticians: Wavelets and Statistics* Grenoble, France.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

An Evolved Dynamical Electronic Robot Control System

Adrian Thompson
adrianth@cogs.susx.ac.uk

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

Abstract It is possible to use artificial evolution as an automatic design process for electronic circuits. The nature of the evolutionary process is very different from that of human design, so a fundamental re-thinking of what electronic circuits can be is called for. This paper provides an easy introduction to some of the ideas by describing a simple, yet important, experiment in non-technical terms. The experiment is the evolution of the first ever evolved hardware robot controller, and the result is remarkable.

Introduction

This paper provides a brief, non-technical, demonstration of ground-breaking (and world-leading) COGS research into the use of artificial evolution as an automatic design process for electronic circuits. The interested reader is referred to (Thompson, 1995a, 1995b) for full details: these papers are also available as COGS technical reports CSRP368 and CSRP385.

The strategy is simply to describe an experiment, and then to identify its most interesting features.

The Task

The aim was to evolve an electronic control system for the robot known as “Mr Chips.” For this experiment, Mr Chips’ only sensors were a pair of time-of-flight sonars, one pointing left and the other right. The actuators were a pair of d.c. motors driving the left and right wheels, which could only turn forwards. The target behaviour of the robot was simply to move about while avoiding the walls in an uncluttered rectangular enclosure.

The Evolvable Hardware Architecture

The evolvable electronic architecture is shown in Figure 1. It is comprised of a pair of binary logic nodes, each able to perform any Boolean function of four inputs. For each node, two of the inputs come directly from the two sonars: these signals consist of five digital pulses per second, with the length of the pulse being equal to the time it took for a burst of ultrasonic sound to travel away from the robot, bounce off an object, and return to the sonar transducer. The other two inputs of each node come from the outputs of the nodes themselves: it is a fully connected recurrent logic network. The two motors are directly turned on or off according to the outputs of the nodes, as shown.

The strange rectangular symbols in the figure represent “Genetic Latches.” These act either just like a normal wire (passing the signal through unchanged), or they can allow the input to change the output only on the regular beating of a clock.

This architecture was implemented directly in hardware, using an off-the-shelf RAM memory chip to implement the logic nodes. In fact, if it were not for the Genetic Latches, we would just have a standard implementation of a finite-state machine. However, the Genetic Latches change the nature of the machine quite radically, allowing much richer dynamical behaviour: I call it a “Dynamic State Machine” (DSM).

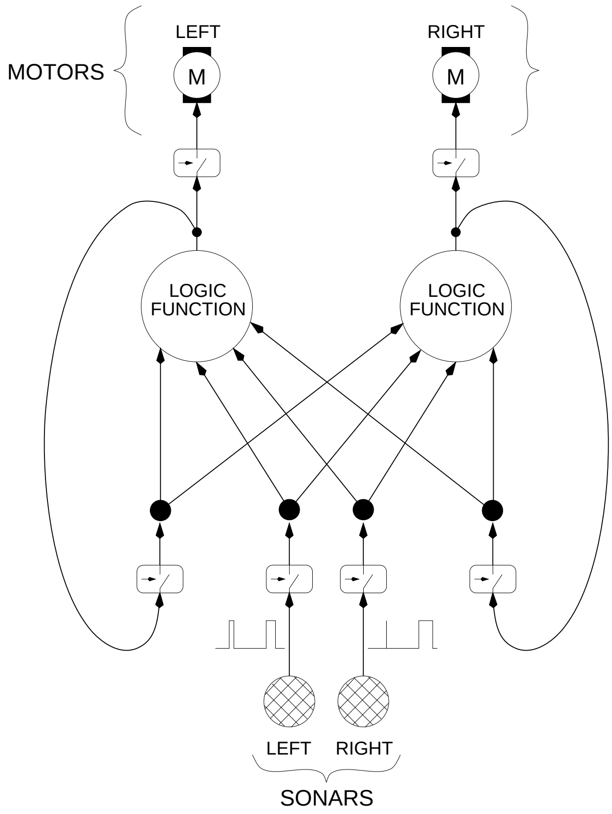

Figure 1: The evolvable electronic architecture.

The Evolution

The Dynamic State Machine, although a real physical piece of electronics, is able effectively to be a whole range of electronic circuits. By changing the contents of the RAM chip, the operation performed by each of the logic nodes can be changed to any one of the 65536 possible binary functions of four inputs. By changing whether the Genetic Latches act like wires or whether they synchronise signal changes to the clock, the fundamental dynamics of the system can be changed. Even the rate of beating of the clock can be changed.

Artificial evolution works with a population of individuals: in this case each individual is a setting of the RAM contents, the Genetic Latches, and the clock rate. Initially, these are all set completely at random. There then follows an iterative process of trying out the individuals to see how good they are at the task (causing the robot to avoid walls), and "breeding" the better ones preferentially by a process inspired by Darwinian evolution. Some of the random individuals will be slightly less terrible at causing the robot to avoid walls than others, and over the course of tens of breeding cycles (with a population of 30 individuals) small improvements build up to give competent wall-avoiders.

To try out each individual, the settings of the RAM, the genetic latches and the clock are used to configure a single piece of DSM hardware, one individual at a time. One way of evaluating the individual's performance would then be to allow the DSM to control the robot as it moves in the real enclosure, and giving it a score according to how well it avoided the walls. It is inconvenient to have an initially incompetent robot on the loose for long periods of time, so instead a virtual-reality simulator was built for it. Just as a human playing a virtual-reality arcade game uses her real body but receives simulated sensory input, so the robot uses its real wheels (but they're jacked up off the floor so it doesn't move) and receives simulated sonar signals. These are generated by a computer simulation of the way sound bounces around the enclosure and the way that wheel rotations cause the robot to move. The arrangement is shown in Figure 2.

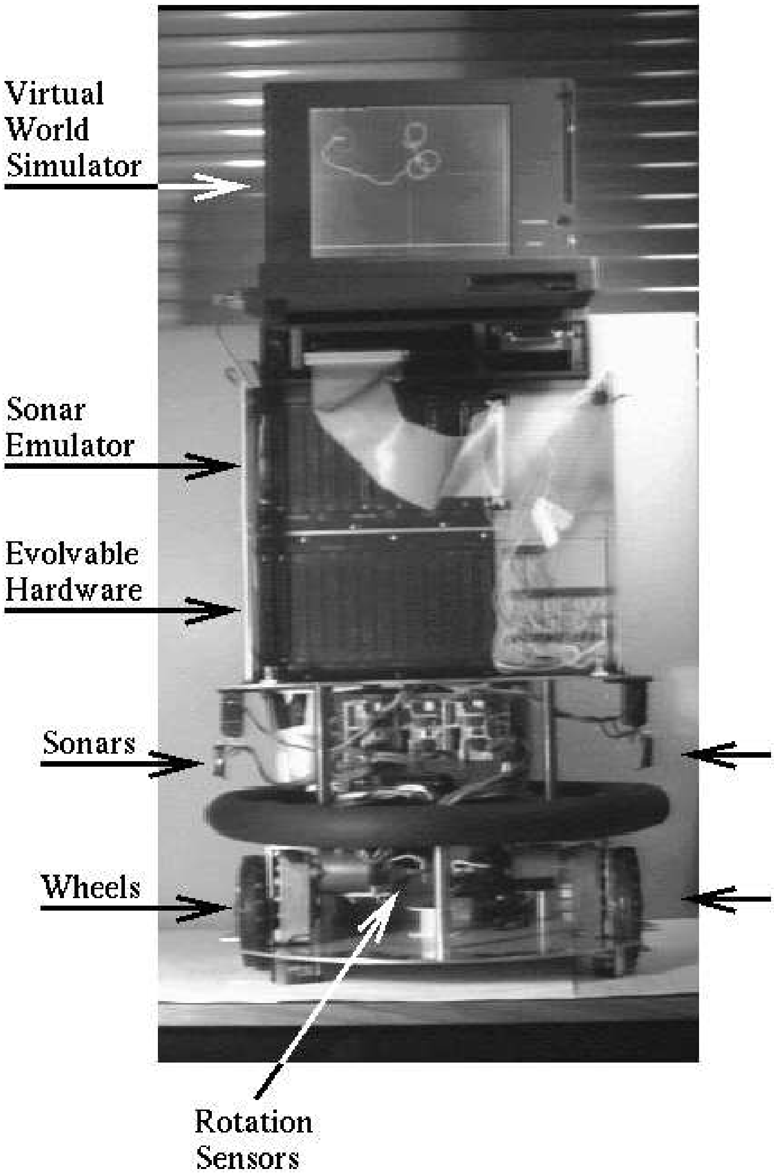

Figure 2: Mr Chips evolving.

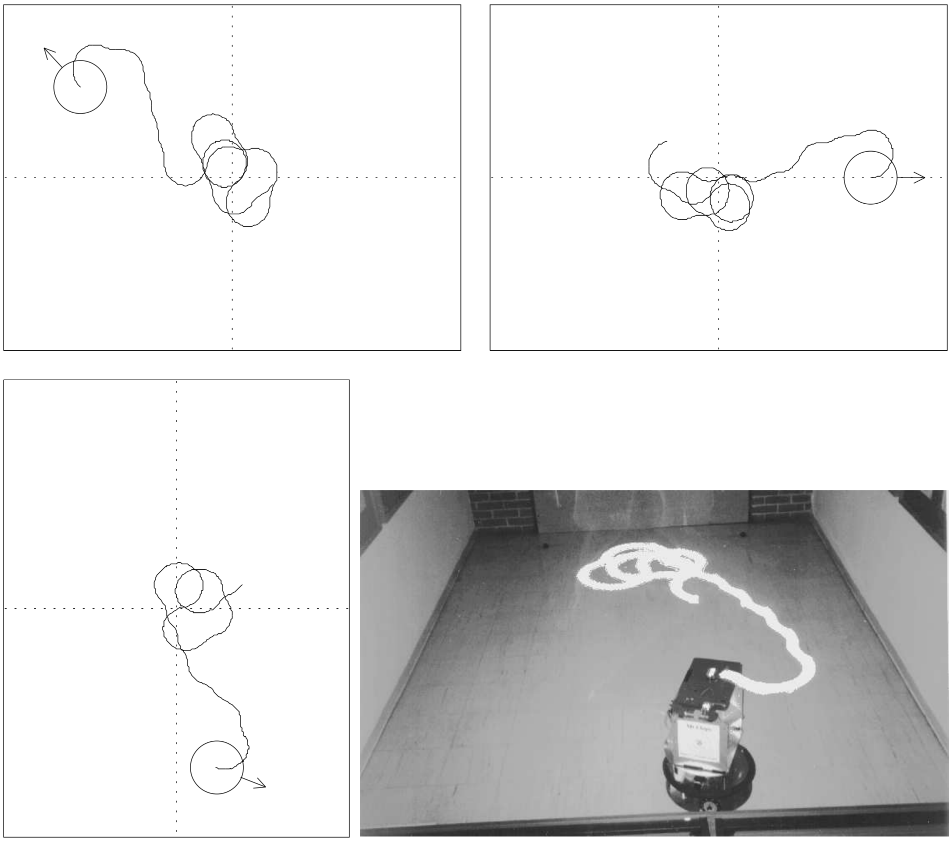

Figure 3: Wall avoidance in virtual reality and (bottom right) in the real world, after 35 generations. The top pictures are of 90 seconds of behaviour, the bottom ones of 60.

The Results

After 35 generations of evolution (35 breeding cycles), the robot was displaying the excellent behaviour seen in Figure 3. The bottom right image in this figure was made by photographing the robot at its initial position, then turning the lights out and leaving the shutter open while the robot moved around with a light fixed on top. In the other images, showing the behaviour in virtual reality, the robot is drawn to scale at its starting position with an arrow indicating the forwards direction; thereafter only the path taken by its centre is plotted.

Why is This Important?

1. The amount of hardware used is *extremely* small. In the final solution, the effective circuitry in-between the sonars and the motors was nothing more than 32 bits of RAM and three flip-flops. This accepted the echo pulses directly from the sonars, and directly controlled the motors; there was no pre- or post-processing.

2. This controller could not have been designed by humans. The Genetic Latches allow the hardware a great deal of dynamical freedom that is normally strenuously avoided in order to make a human's design task tractable. Furthermore, the evolved system can exploit detailed physical properties of the hardware (such as time delays) that a human designer would find hard to measure, never mind design with. The rich dynamical potential of the hardware allows evolution to exploit a tight coupling with the dynamics of the sensors, actuators and the environment which is largely beyond the scope of human design methods.

3. The effectiveness of this approach strengthens earlier proposals that evolvable hardware justifies a radical re-thinking of the nature of electronic systems: many constraints on the dynamics and physical organisation of the circuit can be stripped away, revealing the true power of the underlying hardware.

4. The evolved controller exhibits a certain amount of tolerance to faults in the RAM chip. Part of this arises from the dynamics of evolving populations under certain circumstances, and is the subject of ongoing research. Evolved systems can be as much as 10% less sensitive to faults than equivalent systems arrived at by non-evolutionary means.

Closing

This paper has aimed to give an easy overview of some of the ideas of evolvable hardware. However, there are deep issues to be pondered and hard problems to be tackled at all stages of this research: it is uncharted territory. (The control system described above was the first ever evolved hardware robot controller.) Nevertheless, there is the real prospect of being able to produce powerful new kinds of electronic systems by artificial evolution that cannot be produced by human design.

Acknowledgements

This research is funded by a scholarship from the School of Cognitive and Computing Sciences, for which I am very grateful. My thanks also to Phil Husbands, Dave Cliff and Inman Harvey for being good in so many respects.

References

Thompson, A. (1995a). Evolving electronic robot controllers that exploit hardware resources. In Morán, F., et al. (Eds.), *Advances in Artificial Life: Proceedings of the 3rd European Conference on Artificial Life (ECAL95)*, pp. 640–656. Springer.

Thompson, A. (1995b). Evolving fault tolerant systems. In *Proc. of the 1st IEE/IEEE Int. Conf. on Genetic Algorithms in Engineering Systems (GALESIA'95), IEE Conference Publication No. 414*, pp. 524–529.

In: A. Jonathan Howell & Joseph A. Wood eds., 1995 *The Eighth 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 390.

Showtree, the Next Generation

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

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

Abstract Showtree is provided by Poplog for displaying trees — but it has certain implicit limitations. This paper proposes an alternative, Dotty, which is more general than Showtree and permits interactive graphical manipulation via X-windows. Having shown that Dotty has fewer limitations than Showtree we examine the practical problems of using Dotty from Poplog.

1 Introduction

Mathematically a graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ is defined as a set of nodes ($\mathcal{N}$) and a set of edges ($\mathcal{E}$) joining nodes together. Nodes and edges may have additional properties, and such a graph is called a labeled (or coloured) graph. It follows from this that a graph is just a set of relationships which may be manipulated mathematically. However, graphs also tend to occur frequently in science. An example might have nodes of people, with edges representing communications. A further example would be the graph of a neural network.

Frequently we wish to display graphs and increasingly with interactive computing we wish to manipulate a displayed graph.

2 The Problem

Most programming languages make no special provisions for processing graphs let alone displaying them. Pop-11 (Barrett, Ramsey, & Sloman, 1985) is unusual in supporting the display of trees — a special form of graph. The usual solution to displaying graphs or interacting with them, is to use something like the X-windowing System. X-windows was, however, designed as a general purpose windowing system. Hence there is a large learning curve (and toolkit) associated with using X-windows.

Showtree is a package in Pop-11 for displaying trees, see for example Figure 1.

However, Showtree has several limitations:-

- Wide trees cannot be displayed on paper;
- There is no support for interactive manipulations of displayed trees;
- It is limited to trees, and hence no line crossings can occur.

Dotty (Koutsofios & North, 1994; Koutsofios, 1994) is a system for displaying (directed) graphs in X-windows, such that the user does not need much knowledge of X-windows, but instead thinks in terms of graphs.

*Supported by a CASE award from the Engineering and Physical Sciences Research Council in association with British Telecom Laboratories (ML 464531).

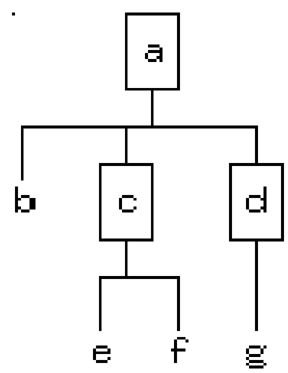

Figure 1: Showtree, sample tree

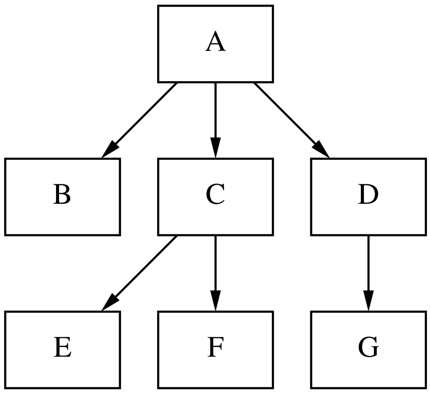

Figure 2: Dotty, sample tree

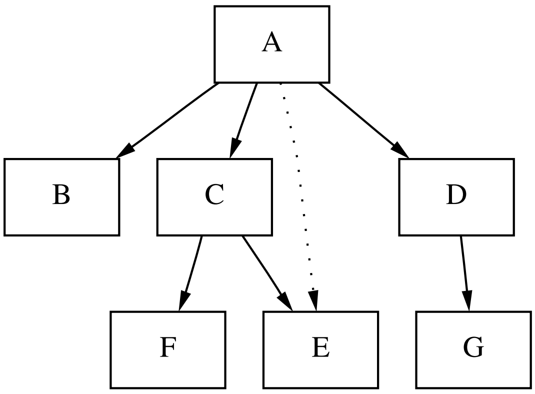

Figure 3: Dotty, illegal tree

Figure 2 shows the same tree as Figure 1 in Dotty. Notice that adding the (dotted) edge *AE*, (see Figure 3), results in a graph Dotty can still display, but it is no longer a tree and cannot be displayed by Showtree.

Before discussing Dotty in more detail, it is worth pointing out that the limitations identified in Showtree do not apply, since:-

- Dotty can produce output in PostScript[1], which is easily displayed on many different devices, see for example (Thomas, 1988);
- Dotty is programmable and can be made to respond to mouse clicks;
- Dotty is not limited to trees, as Figure 2 shows;
- An arbitrary graph cannot be represented on a piece of paper without line crossings. Dotty tries to reduce line crossings.

As supplied by AT&T, Dotty is in fact a graph visualisation system and can be used either to display graphs or via its programmable interface to trigger its sub-process to do additional work. Hence the intended model of usage is as shown in Figure 4.

[1]PostScript is a trademark of Adobe Systems Incorporated

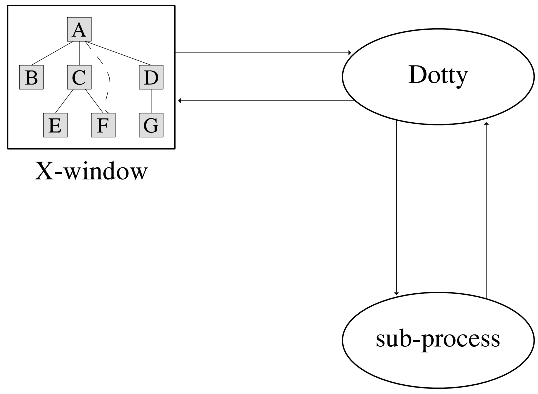

Figure 4: Dotty usual configuration

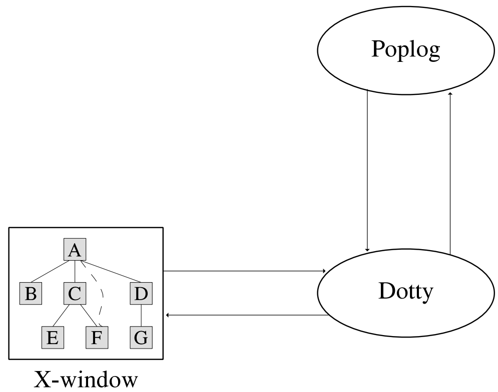

Figure 5: Preferred Poplog configuration

3 Dotty and Poplog

Our implementations of both Poplog[2] (Barrett et al., 1985) and Dotty run under the UNIX[3] operating system. For a description of UNIX see (Leffler, McKusick, Karels, & Quarterman, 1989)

We wanted to use Pop-11 with Dotty and since we wanted all the facilities of the Poplog system, we required Dotty to be a sub-process of Poplog, see Figure 5. In point of fact, Dotty forks its own sub-process (Dot) for calculating graph layouts.

This raised a number of communications problems:-

- How to send commands to Dotty;
- How to send graphs to Dotty;
- How to get data and error reports from Dotty.

Since Dotty is to be a sub-process of Poplog the obvious solution is to use pipes, and this indeed was the chosen option. However, Dotty's behaviour is partly governed by human input and is thus asynchronous.

Fortunately, Dotty has a start-up option to read and execute an initial set of commands from a file. Therefore, we can get 'inside' Dotty and plant our own code and start-up procedures.

We shall now explore each of the previously identified communication problems in turn.

[2]Poplog is a registered trademark of the University of Sussex
[3]UNIX is registered trademark of AT&T Bell Laboratories

3.1 Commands to Dotty

Let us consider the problem of commands to Dotty. This is easy, Dotty, has an inbuilt primitive, 'run' which executes any supplied string. Hence all we need to do is get Dotty to execute a loop of the form

repeat getinput () $->$ s ; run (s) ; endrepeat ;

and apart from minor complications with getting Dotty to read asynchronous input from a pipe that completes the passing of commands to Dotty.

3.2 Graphs to Dotty

Sending a graph to Dotty, should now be easy, but UNIX does not support an end-of-file character so there is no way to indicate the end of the graph. The only solution is to write the graph data out to a temporary file and pass the file's name into Dotty as a 'read graph from file' style command.

Since we want to supplement Showtree with Dotty, it is necessary to demonstrate how to convert Showtree's input format to that of Dotty. This is sketched in Figure 6

3.3 Data from Dotty

Before discussing data from Dotty, we should point out that the following is hypothetical — it should work — the authors of Dotty agree. But the version of Dotty we currently have in COGS is not robust, a new version will be installed shortly.

Data from Dotty is in principle simple, Dotty has a command to output a line of text to a file descriptor (hence a pipe); akin to `fprintf`. So we just sent the data down a pipe, created when the process is `exec`ed, in ASCII format. Currently, this fails because Dotty closes the pipe on start-up.

The other part of this puzzle is getting Poplog to recognize it has data available. This is handled by an interrupt routine which regularly polls the pipe looking for data, when found a flag is set which the main program can examine at its own convenience. The nature of pipes means that no data is lost or overwritten by this simple approach.

4 Conclusion

Once the bug mentioned above is fixed, we can send data to and from Dotty. As we have already seen Dotty provides all the functionality of Showtree and permits interactive graph manipulation. Hence Dotty could serve as a sophisticated replacement for Showtree.

References

Barrett, R., Ramsey, A., & Sloman, A. (1985). *POP11: A Practical Language for Artifical Intelligence*. Ellis Horwood.

Koutsofios, E. (1994). *Editing Pictures with* Lefty. AT&T Bell Laboratories, Murray Hill, NJ.

Koutsofios, E., & North, S. C. (1994). *Editing graphs with* dotty. AT&T Bell Laboratories, Murray Hill, NJ.

Leffler, S. J., McKusick, M. K., Karels, M. J., & Quarterman, J. S. (1989). *The Design and Implementation of the 4.3 BSD Operating System.* Addison-Wesley Publishing Company, Reading, MA.

Thomas, B. (1988). *A PostScript Cookbook.* Macmillan Education Ltd., Basingstoke, England.

```
define showtree_to_dotty ( list ) -> name ;
;;; This procedure takes input in the form of
;;; showtree, and converts it to a series of
;;; output statements, that model the input to
;;; dotty. Preamble and postamble are ignored.

;;; list is the input list
;;; name is named head of the list
    lvars list, name ;

;;; declare head and tail of list
;;; and loop iterator
    lvars _hd, _tl, item ;

;;; if input is just an element, use this as the name
    if atom ( list ) then
        list -> name ;
    else

;;; split the list into head and tail
        dest ( list ) -> _tl -> _hd ;

;;;     if head is an element, then
        if atom ( _hd ) then
;;;         name the head as given
            _hd -> name ;
        else
;;;         generate a new node for the unnamed element
            gensym ( "void" ) -> name ;
;;;         and use the list as the list's tail
            list -> _tl ;
        endif ;

        for item in _tl do
;;;         find the name of the element, and print it
            lvars name2 = showtree_to_dotty ( item ) ;
            printf ( '%P -> %P\n', [% name, name2 %] ) ;
        endfor ;

    endif ;

enddefine ;
```

Figure 6: Basic Algoritm Showtree input to Dotty's input