

Research and Development in Intelligent Systems XXI

Max Bramer, Frans Coenen and
Tony Allen (Eds)

Research and Development in Intelligent Systems XXI

**Proceedings of AI-2004, the Twenty-fourth SGAI
International Conference on Innovative Techniques
and Applications of Artificial Intelligence**

Professor Max Bramer, BSc, PhD, CEng, FBCS, FIEE, FRSA
Faculty of Technology, University of Portsmouth, Portsmouth, UK

Dr Frans Coenen
Department of Computer Science, University of Liverpool, Liverpool, UK

Dr Tony Allen
Nottingham Trent University

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

ISBN 1-85233-907-1
Springer is part of Springer Science+Business Media
springeronline.com

© Springer-Verlag London Limited 2005
Printed in Great Britain

The use of registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Typesetting: Camera-ready by editors
Printed and bound at the Athenæum Press Ltd., Gateshead, Tyne & Wear
34/3830-543210 Printed on acid-free paper SPIN 11006770

TECHNICAL PROGRAMME CHAIR'S INTRODUCTION

M.A.BRAMER

University of Portsmouth, UK

This volume comprises the refereed technical papers presented at AI-2004, the Twenty-fourth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, held in Cambridge in December 2004. The conference was organised by SGAI, the British Computer Society Specialist Group on Artificial Intelligence.

The papers in this volume present new and innovative developments in the field, divided into sections on AI Techniques I and II, CBR and Recommender Systems, Ontologies, Intelligent Agents and Scheduling Systems, Knowledge Discovery in Data and Spatial Reasoning and Image Recognition.

This year's prize for the best refereed technical paper was won by a paper entitled *Extracting Finite Structure from Infinite Language* by T. McQueen, A. A. Hopgood, T. J. Allen and J. A. Tepper (School of Computing & Informatics, Nottingham Trent University, UK). SGAI gratefully acknowledges the long-term sponsorship of Hewlett-Packard Laboratories (Bristol) for this prize, which goes back to the 1980s.

This is the twenty-first volume in the *Research and Development* series. The Application Stream papers are published as a companion volume under the title *Applications and Innovations in Intelligent Systems XII*.

On behalf of the conference organising committee I should like to thank all those who contributed to the organisation of this year's technical programme, in particular the programme committee members, the executive programme committee and our administrators Linsay Turbert and Collette Jackson.

Max Bramer

Technical Programme Chair, AI-2004

ACKNOWLEDGEMENTS

AI-2004 CONFERENCE COMMITTEE

Dr. Tony Allen, Nottingham Trent University	(Conference Chair)
Dr Robert Milne, Sermatech Intelligent Applications Ltd	(Deputy Conference Chair, Finance and Publicity)
Dr. Alun Preece, University of Aberdeen	(Deputy Conference Chair, Electronic Services)
Dr Nirmalie Wiratunga, Robert Gordon University, Aberdeen	(Deputy Conference Chair, Poster Session)
Prof. Adrian Hopgood Nottingham Trent University	(Tutorial Organiser)
Prof. Ann Macintosh Napier University	(Application Programme Chair)
Richard Ellis Stratum Management Ltd	(Deputy Application Programme Chair)
Professor Max Bramer University of Portsmouth	(Technical Programme Chair)
Dr Frans Coenen, University of Liverpool	(Deputy Technical Programme Chair)
Dr. Bob Howlett, University of Brighton	(Exhibition Organiser)
Rosemary Gilligan	(Research Student Liaison)

TECHNICAL EXECUTIVE PROGRAMME COMMITTEE

Prof. Max Bramer, University of Portsmouth (Chair)
Dr. Frans Coenen, University of Liverpool (Vice-Chair)
Dr. Tony Allen, Nottingham Trent University
Prof. Adrian Hopgood, Nottingham Trent University
Mr. John Kingston, University of Edinburgh
Dr. Peter Lucas, University of Nijmegen, The Netherlands
Dr. Alun Preece, University of Aberdeen

TECHNICAL PROGRAMME COMMITTEE

Alia Abdelmoty (Cardiff University)

Andreas A Albrecht (University of Hertfordshire)

Tony Allen (Nottingham Trent University)

Somaya A. S. Almaadeed (Qatar University)

Yaxin Bi (Queen's University Belfast)

Arkady Borisov (Riga Technical University)

Max Bramer (University of Portsmouth)

Ken Brown (University College Cork)

Frans Coenen (University of Liverpool)

Bruno Cremilleux (University of Caen)

Juan A. Fdez. del Pozo (Technical University of Madrid)

Marina De Vos (University of Bath)

John Debenham (University of Technology, Sydney)

Stefan Diaconescu (Softwin)

Nicolas Durand (University of Caen)

Anneli Edman (University of Upsala)

Mark Elshaw (University of Sunderland)

Max Garagnani (The Open University)

Adriana Giret (Universidad Politecnica de Valencia)

Mercedes Gomez Albarran (Univ. Complutense de Madrid)

Martin Grabmüller (Technische Universität Berlin)

Anne Håkansson (Uppsala University, Sweden)

Mark Hall (University of Waikato, New Zealand)

Eveline M. Helsper (Utrecht University)

Ray Hickey (University of Ulster)

Adrian Hopgood (The Nottingham Trent University)

Chihli Hung (De Lin Institute of Technology, Taiwan)

Piotr Jedrzejowicz (Gdynia Maritime University, Poland)

John Kingston (University of Edinburgh)

T. K. Satish Kumar (Stanford University)

Alvin C. M. Kwan (University of Hong Kong)

Brian Lees (University of Paisley)

Peter Lucas (University of Nijmegen)

Angeles Manjarrés (Universidad Nacional de Educación a Distancia, Spain)

Daniel Manrique Gamo

Raphaël Marée (University of Liège, Belgium)

David McSherry (University of Ulster)

Alfonsas Misevicius (Kaunas University of Technology)

Ernest Muthomi Mugambi (Sunderland University, UK)

Lars Nolle (Nottingham Trent University)

Tomas Eric Nordlander (University of Aberdeen)

Tim Norman (University of Aberdeen)

Dan O'Leary (University of Southern California)

Barry O'Sullivan (University College Cork)

Alun Preece (University of Aberdeen)

Gerrit Renker (Robert Gordon University)

María Dolores Rodríguez-Moreno (Universidad de Alcalá)

Fernando Sáenz Pérez (Universidad Complutense de Madrid)

Miguel A. Salido (Universidad de Alicante)

Barry Smyth (University College Dublin)

Jon Timmis (University of Kent)

Kai Ming Ting (Monash University)

Andrew Tuson (City University)

M.R.C. van Dongen (University College Cork)

Ian Watson (University of Auckland)

Graham Winstanley (University of Brighton)

Nirmalie Wiratunga (Robert Gordon University)

Shengxiang Yang (University of Leicester)

CONTENTS

BEST TECHNICAL PAPER

Extracting Finite Structure from Infinite Language (x) <i>T. McQueen, A. A. Hopgood, T. J. Allen and J. A. Tepper, School of Computing & Informatics, Nottingham Trent University, UK</i>	3
--	---

SESSION 1a: AI TECHNIQUES I

Modelling Shared Extended Mind and Collective Representational Content <i>Tibor Bosse, Catholijn M. Jonke and Martijn C. Schut, Department of Artificial Intelligence, Vrije Universiteit Amsterdam; Jan Treur, Department of Artificial Intelligence, Vrije Universiteit Amsterdam and Department of Philosophy, Universiteit, Utrecht</i>	19
Overfitting in Wrapper-Based Feature Subset Selection: The Harder You Try the Worse it Gets <i>John Loughrey and Pádraig Cunningham, Trinity College Dublin, Ireland</i>	33
Managing Ontology Versions with a Distributed Blackboard Architecture <i>Ernesto Compatangelo, Wamberto Vasconcelos and Bruce Scharlau, Department of Computing Science, University of Aberdeen</i>	44
OntoSearch: An Ontology Search Engine <i>Yi Zhang, Wamberto Vasconcelos and Derek Sleeman, Department of Computing Science, University of Aberdeen, Aberdeen, UK</i>	58

SESSION 1b: CBR AND RECOMMENDER SYSTEMS

Case Based Adaptation Using Interpolation over Nominal Values <i>Brian Knight, University of Greenwich, UK and Fei Ling Woon, Tunku Abdul Rahman College, Kuala Lumpur, Malaysia</i>	73
Automating the Discovery of Recommendation Rules <i>David McSherry, School of Computing and Information Engineering, University of Ulster, Northern Ireland</i>	87
Incremental Critiquing (x) <i>James Reilly, Kevin McCarthy, Lorraine McGinty and Barry Smyth, Department of Computer Science, University College Dublin, Ireland</i>	101

Note: x indicates SGAI recognition award

SESSION 2: AI TECHNIQUES II

A Treebank-Based Case Role Annotation Using An Attributed String Matching <i>Samuel W.K.Chan, Department of Decision Sciences, The Chinese University of Hong Kong, Hong Kong, China</i>	117
A Combinatorial Approach to Conceptual Graph Projection Checking <i>Madalina Croitoru and Ernesto Compatangelo, Department of Computing Science, University of Aberdeen</i>	130
Implementing Policy Management Through BDI <i>Simon Miles, Juri Papay, Michael Luck and Luc Moreau, University of Southampton, UK</i>	144
Exploiting Causal Independence in Large Bayesian Networks (x) <i>Rasa Jurgelenaite and Peter Lucas, Radboud University Nijmegen, The Netherlands</i>	157

SESSION 3: INTELLIGENT AGENTS AND SCHEDULING SYSTEMS

A Bargaining Agent Aims to 'Play Fair' <i>John Debenham, Faculty of Information Technology, University of Technology, Sydney, NSW, Australia</i>	173
Resource Allocation in Communication Networks Using Market-Based Agents (x) <i>Nadim Haque, Nicholas R. Jennings and Luc Moreau, School of Electronics and Computer Science, University of Southampton, Southampton, UK</i>	187
Are Ordinal Representations Effective? <i>Andrew Tuson, Department of Computing, City University, UK</i>	201
A Framework for Planning with Hybrid Models <i>Max Garagnani, Department of Computing, The Open University, UK</i>	214

SESSION 4: KNOWLEDGE DISCOVERY IN DATA

Towards Symbolic Data Mining in Numerical Time Series <i>Agustín Santamaría, Technical University of Madrid, Spain; África López-Illescas, High Council for Sports, Madrid, Spain; Aurora Perez-Perez and Juan P. Caraña-Valente, Technical University of Madrid, Spain</i>	231
Support Vector Machines of Interval-based Features for Time Series Classification (x) <i>Juan Jose Rodriguez, Universidad de Burgos, Spain and Carlos J. Alonso, Departamento de Informatica, Universidad de Valladolid, Spain</i>	244

Neighbourhood Exploitation in Hypertext Categorization <i>Houda Benbrahim and Max Bramer, Department of Computer Science and Software Engineering, University of Portsmouth, UK</i>	258
--	-----

Using Background Knowledge to Construct Bayesian Classifiers for Data- Poor Domains <i>Marcel van Gerven and Peter Lucas, Institute for Computing and Information Sciences, University of Nijmegen, The Netherlands</i>	269
---	-----

SESSION 5: SPATIAL REASONING, IMAGE RECOGNITION AND HYPERCUBES

Interactive Selection of Visual Features through Reinforcement Learning <i>Sebastien Jodogne and Justus H. Piater, Montefiore Institute, University of Liege, Belgium.....</i>	285
---	-----

Imprecise Qualitative Spatial Reasoning <i>Baher El-Geresy, Department of Computer Studies, University of Glamorgan, UK and Alia Abdelmoty, Department of Computer Science, Cardiff University, UK.....</i>	299
--	-----

Reasoning with Geometric Information in Digital Space (x) <i>Passent El-Kafrawy and Robert McCartney, Department of Computer Science and Engineering, University of Connecticut, USA</i>	313
--	-----

On Disjunctive Representations of Distributions and Randomization <i>T. K. Satish Kumar, Knowledge Systems Laboratory, Stanford University</i>	327
---	-----

AUTHOR INDEX	341
---------------------------	------------

BEST TECHNICAL PAPER

Extracting Finite Structure from Infinite Language

T. McQueen, A. A. Hopgood, T. J. Allen, and J. A. Tepper
School of Computing & Informatics, Nottingham Trent University,
Burton Street, Nottingham, NG1 4BU, UK
thomas.mcqueen{adrian.hopgood , tony.allen, jonathan.tepper}@ntu.ac.uk
www.ntu.ac.uk

Abstract

This paper presents a novel connectionist memory-rule based model capable of learning the finite-state properties of an input language from a set of positive examples. The model is based upon an unsupervised recurrent self-organizing map [1] with laterally interconnected neurons. A derivation of functional-equivalence theory [2] is used that allows the model to exploit similarities between the future context of previously memorized sequences and the future context of the current input sequence. This bottom-up learning algorithm binds functionally-related neurons together to form states. Results show that the model is able to learn the Reber grammar [3] perfectly from a randomly generated training set and to generalize to sequences beyond the length of those found in the training set.

1. Introduction

Since its inception, language acquisition has been one of the core problems in artificial intelligence. The ability to communicate through spoken or written language is considered by many philosophers to be the hallmark of human intelligence. Researchers have endeavoured to explain this human propensity for language in order both to develop a deeper understanding of cognition and also to produce a model of language itself. The quest for an automated language acquisition model is thus the ultimate aim for many researchers [4]. Currently, the abilities of many natural language processing systems, such as parsers and information extraction systems, are limited by a prerequisite need for an incalculable amount of manually derived language and domain-specific knowledge. The development of a model that could automatically acquire and represent language would revolutionize the field of artificial intelligence, impacting on almost every area of computing from Internet search engines to speech-recognition systems.

Language acquisition is considered by many to be a paradox. Researchers such as Chomsky argue that the input to which children are exposed is insufficient for them to determine the grammatical rules of the language. This argument for the poverty of stimulus [5] is based on Gold's theorem [6], which proves that most classes of

languages cannot be learnt using only positive evidence, because of the effect of overgeneralization. Gold's analysis and proof regarding the unfeasibility of language acquisition thus forms a central conceptual pillar of modern linguistics. However, less formal approaches have questioned the treatment of language identification as a deterministic problem in which any solution must involve a guarantee of no future errors. Such approaches to the problem of language acquisition [7] show that certain classes of language can be learnt using only positive examples if language identification involves a stochastic probability of success.

Language acquisition, as with all aspects of natural language processing, traditionally involves hard-coded symbolic approaches. Such top-down approaches to cognition attempt to work backwards from formal linguistic structure towards human processing mechanisms. However, recent advances in cognitive modelling have led to the birth of connectionism, a discipline that uses biologically inspired models that are capable of learning by example. In contrast to traditional symbolic approaches, connectionism uses a bottom-up approach to cognition that attempts to solve human-like problems using biologically inspired networks of interconnected neurons. Connectionist models learn by exploiting statistical relationships in their input data, potentially allowing them to discover the underlying rules for a problem. This ability to learn the rules, as opposed to learning via rote memorization, allows connectionist models to generalize their learnt behaviour to unseen exemplars. Connectionist models of language acquisition pose a direct challenge to traditional nativist perspectives based on Gold's theorem [6] because they attempt to learn language using only positive examples.

2. Connectionism and Determinacy

Since the early nineties, connectionist models such as the simple recurrent network (SRN) [8] have been applied to the language acquisition problem in the form of grammar induction. This involves learning simple approximations of natural language, such as regular and context-free grammars. These experiments have met with some success [6, 7], suggesting that dynamic recurrent networks (DRNs) can learn to emulate finite-state automata. However, detailed analysis of models trained on these tasks show that a number of fundamental problems exist that may derive from using a model with a continuous state-space to approximate a discrete problem.

While DRNs are capable of learning simple formal languages, they are renowned for their instability when processing long sequences that were not part of their training set [8, 9]. As detailed by Kolen [10], a DRN is capable of partitioning its state space into regions approximating the states in a grammar. However, sensitivity to initial conditions means that each transition between regions of state space will result in a slightly different trajectory. This causes instability when traversing state trajectories

that were not seen during training. This is because slight discrepancies in the trajectories will be compounded with each transition until they exceed the locus of the original attractor, resulting in a transition to an erroneous region of state space. Such behavior is characteristic of continuous state-space DRNs and can be seen as both a power and a weakness of this class of model. While this representational power enables the model to surpass deterministic finite automata and emulate non-deterministic systems, it proves to be a significant disadvantage when attempting to emulate the deterministic behavior fundamental to deterministic finite state automata (DFA).

Attempts have been made to produce discrete state-space DRNs by using a step-function for the hidden layer neurons [9]. However, while this technique eliminates the instability problem, the use of a non-differentiable function means that the weight-update algorithm's sigmoid function can only approximate the error signal. This weakens the power of the learning algorithm, which increases training times and may cause the model to learn an incorrect representation of the DFA.

The instability of DRNs when generalizing to long sequences that are beyond their training sets is a limitation that is probably endemic to most continuous state-space connectionist models. However, when finite-state extraction techniques [9] are applied to the weight space of a trained DRN, it has been shown that once extracted into symbolic form, the representations learnt by the DRN can perfectly emulate the original DFA, even beyond the training set. Thus, while discrete symbolic models may be unable to adequately model the learning process itself, they are better suited to representing the learnt DFA than the original continuous state-space connectionist model.

While supervised DRNs such as the SRN dominate the literature on connectionist temporal sequence processing, they are not the only class of recurrent network. Unsupervised models, typically based on the self-organizing map (SOM) [11], have also been used in certain areas of temporal sequence processing [12]. Due to their localist nature, many unsupervised models operate using a discrete state-space and are therefore not subject to the same kind of instabilities characteristic of supervised continuous state-space DRNs. The aim of this research is therefore to develop an unsupervised discrete state-space recurrent connectionist model that can induce the finite-state properties of language from a set of positive examples.

3. A Memory-Rule Based Theory of Linguistics

Many leading linguists, such as Pinker [13] and Marcus [14], have theorized that language acquisition, as well as other aspects of cognition, can be explained using a memory-rule based model. This theory proposes that cognition uses two separate mechanisms that work together to form memory. Such a dual-mechanism approach is supported by neuro-biological research, which suggests that human memory

operates using a declarative fact-based system and a procedural skill-based system [15]. In this theory, rote memorization is used to learn individual exemplars, while a rule-based mechanism operates to override the original memorizations in order to produce behaviour specific to a category. This memory-rule theory of cognition is commonly explained in the context of the acquisition of the English past tense [13]. Accounting for children's over-regularizations during the process of learning regular and irregular verbs constitutes a well-known battlefield for competing linguistic theories. Both Pinker [13] and Marcus [14] propose that irregular verbs are learnt via rote-memorization, while regular verbs are produced by a rule. The evidence for this rule-based behaviour is cited as the over-regularization errors produced when children incorrectly apply the past tense rule to irregular verbs (e.g. *runned* instead of *ran*).

The model presented in this paper is a connectionist implementation of a memory-rule based system that extracts the finite-state properties of an input language from a set of positive example sequences. The model's bottom-up learning algorithm uses functional-equivalence theory [2] to construct discrete-symbolic representations of grammatical states (Figure 1).

4. STORM (Spatio Temporal Self-Organizing Recurrent Map)

STORM is a recurrent SOM [1] that acts as a temporal associative memory, initially producing a localist-based memorization of input sequences. The model's rule-based mechanism then exploits similarities between the future context of memorized sequences and the future context of input sequences. These similarities are used to construct functional-relationships, which are equivalent to states in the grammar. The next two sections will detail the model's memorization and rule-based mechanisms separately.

4.1 STORM's Memorization Mechanism

STORM maintains much of the functionality of the original SOM [11], including the winning-neuron selection algorithm (Equation 1), weight-update algorithm (Equation 2) and neighbourhood function (Equation 3). The model's localist architecture is used to represent each element of the input sequence using a separate neuron. In this respect, STORM exploits the SOM's abilities as a vector quantization system rather than as a topological map. Equation 1 shows that for every input to the model (X), the neuron whose weight vector has the lowest distance measure from the input vector is selected as the winning neuron (Y). The symbol d denotes the distance between the winning neuron and the neuron in

question. As shown in fig 1, each input vector consists of the current input symbol and a context vector, representing the location of the previous winning neuron.

$$y_i = \arg \min_i (d(x, w_i)) \quad (1)$$

The weight update algorithm (equation 2) is then applied to bring the winning neuron's weight vector (W), along with the weight vectors of neighbouring neurons, closer to the input vector (X) (equation 2). The rate of weight change is controlled by the learning rate α , which is linearly decreased through training.

$$w_{ij}(t+1) = w_{ij}(t) + \alpha h_{ij}(x(t) - w_{ij}(t)) \quad (2)$$

The symbol h in equation 2 denotes the neighbourhood function (equation 3). This standard Gaussian function is used to update the weights of neighbouring neurons in proportion to their distance from the winning neuron. This weight update function, in conjunction with the neighbourhood function, has the effect of mapping similar inputs to similar locations on the map and also minimizing weight sharing between similar inputs. The width of the kernel σ is linearly decreased through training.

$$h_{ij} = \exp\left(\frac{-d_{ij}^2}{2\sigma^2}\right) \quad (3)$$

The model uses an orthogonal input vector to represent the grammar's terminal symbols. Each of the seven terminal symbols are represented by setting the respective binary value to 1 and setting all the other values to 0 (table 1).

Grammatical symbol	Orthogonal vector
B	1 0 0 0 0 0 0
T	0 1 0 0 0 0 0
P	0 0 1 0 0 0 0
S	0 0 0 1 0 0 0
X	0 0 0 0 1 0 0
V	0 0 0 0 0 1 0
E	0 0 0 0 0 0 1

Table 1 – Orthogonal vector representations for input symbols

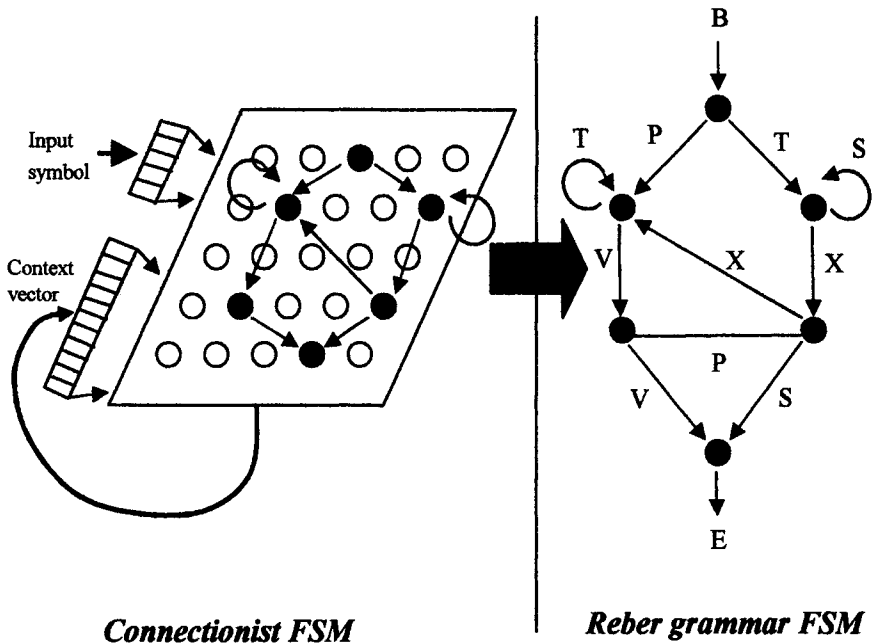


Fig. 1 – Diagram showing conceptual overview of model. The left side shows STORM's representation of a FSM, while the right side of the diagram shows the FSM for the Reber grammar.

As shown in Figures 1 and 2, STORM extends Kohonen's SOM [11] into the temporal domain by using recurrent connections. The recurrency mechanism feeds back a representation of the previous winning neuron's location on the map using a 10-bit Gray-code vector. By separately representing the column and row of the previous winning neuron in the context vector, the recurrency mechanism creates a 2D representation of the neuron's location. Further details of the recurrency mechanism, along with its advantages, are provided in [1]. This method of explicitly representing the previous winner's location as part of the input vector has the effect of selecting the winning neuron based not just on the current input, but also indirectly on all previous inputs in the sequence. The advantage of this method of recurrency is that it is more efficient than alternative methods (e.g. [16]), because only information pertaining to the previous winning neuron's location is fed back. Secondly, the amount of information fed back isn't directly related to the size of the map (i.e. recursive SOM [16] feeds back a representation of each neuron's activation). This allows the model to scale up to larger problems without exponentially increasing computational complexity.

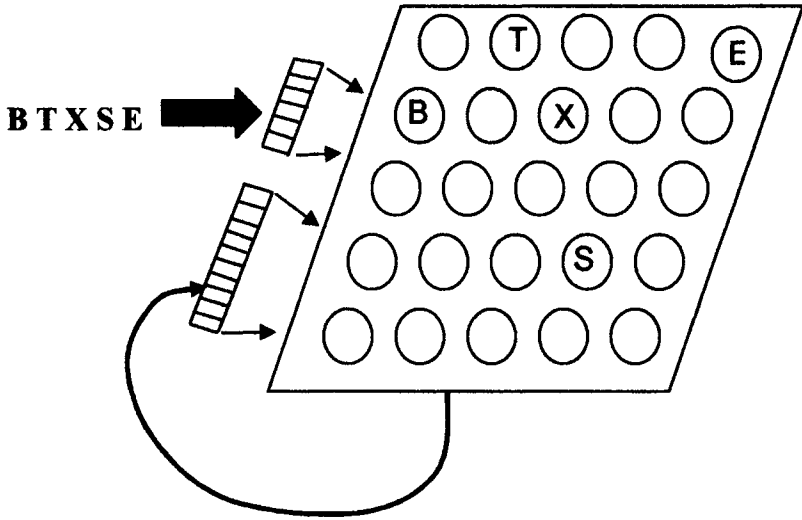


Fig. 2 – Diagram showing STORM's input representation. The model's weight vector consists of a 7-bit orthogonal symbol vector representing the terminal symbol in the grammar, along with a 10-bit Gray code context vector, representing the column and row of the previous winning neuron.

4.2 STORM's Rule-Based Construction Mechanism

The model's location-based recurrency representation and localist architecture provide it with a very important ability. Unlike using conventional artificial neural networks, the sequences learnt by STORM can be extracted in reverse order. This makes it possible to start with the last element in an input sequence and work backwards to find the winning neurons corresponding to the previous inputs in the sequence. STORM uses this ability, while processing input sequences, to find any existing pre-learned sequences that end with the same elements as the current input sequence. For example, Figure 3 shows that the winning neuron for the symbol 'T' in sequence 1 has the same future context ('XSE') as the winning neuron for the first symbol 'S' in sequence 2.

Functional-equivalent theory [2] asserts that two states are said to be equivalent if, for all future inputs, their outputs are identical. STORM uses the inverse of this theory to construct states in a bottom-up approach to grammar acquisition. By identifying neurons with consistently identical future inputs, the model's temporal Hebbian learning mechanism (THL) mechanism binds together potential states via lateral connections. By strengthening the lateral connections between neurons that

have the same future context, this THL mechanism constructs functional-relationships between the winning neuron for the current input and the winning neuron for a memorized input (referred to as the alternative winner) whose future-context matches that of the current input sequence (Figure 4). In order to prevent lateral weight values from becoming too high, a negative THL value is applied every time a winning neuron is selected. This has the effect of controlling lateral weight growth and also breaking down old functional relationships that are no longer used.

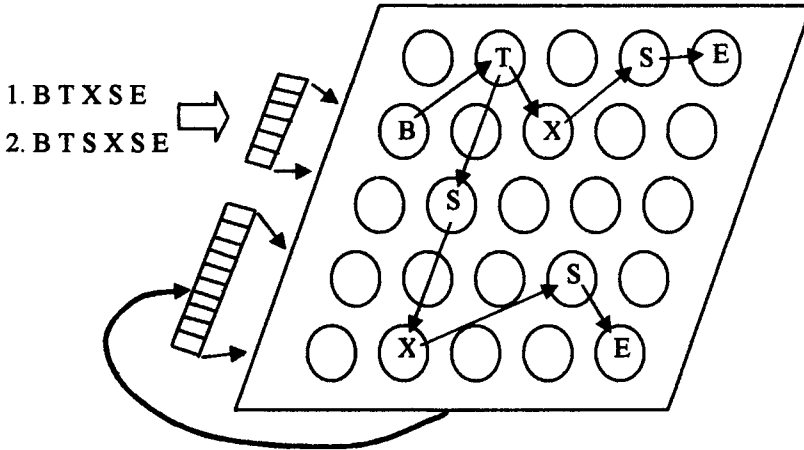


Fig. 3 – Diagram showing the memorized winning neurons for two sequences that end with the same sub-sequence ‘XSE’

Once states have formed, they override the recurrency mechanism, forcing the model to use a single representation for the future inputs in the sequence rather than the original two representations (Figure 4). The advantage of forming states in this manner is that it provides the model with a powerful ability to generalize beyond its original memorizations. The model’s THL mechanism conforms to the SOM’s winner-take-all philosophy by selecting the alternative winner as the neuron whose future-context is the best match to that of the current input sequence. Given that tracing back through the future-context may identify multiple alternative winners, the criteria of best matching winner classifies the strongest sequence stored in the model as the winner. Furthermore, THL is only used to enhance the functional relationship between the winner and the alternative winner, if the future-context for the alternative winner is stronger than that of the winner itself. Thus, the model has a preference for always using the dominant sequence and it will use the THL mechanism to re-wire its internal pathways in order to use any dominant sequence.

Constructing the lateral connections between functionally-related neurons is equivalent to identifying states in a grammar. Once the strength of these lateral connections exceeds a certain threshold they override the standard recurrency mechanism, affecting the representation of the previous winning neuron that is fed back (Figure 4). Instead of feeding back a representation of the previous winning neuron, the lateral connections may force the model to feed back a representation of the functionally-related neuron. The consequence of this is that the rest of the sequence is processed as if the functionally-related neuron had been selected rather than the actual winner. For example, Figure 4 shows that when the first 'S' symbol in sequence 2 is presented to STORM, its winning neuron is functionally linked to the winner for the 'T' symbol from sequence 1. As the latter winning neuron is the dominant winner for this state, its location is fed back as context for the next symbol in sequence 2.

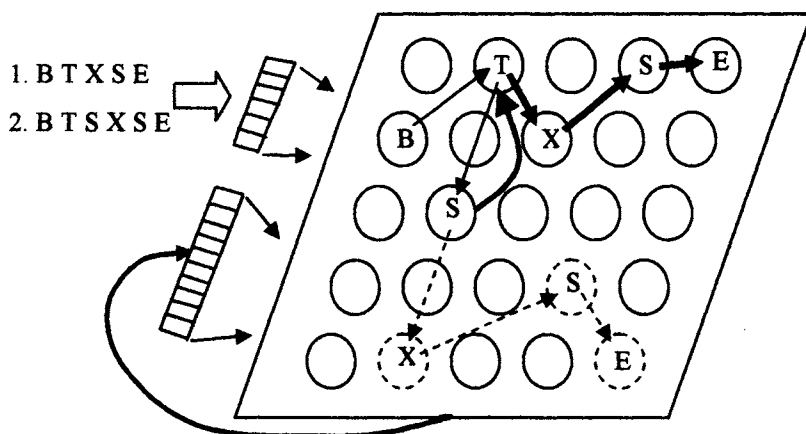


Fig. 4 – Functional override in winning-neuron selection algorithm. The functional relationship (shown in grey) between the third symbol 'S' in the second sequence and the second symbol 'T' in the first sequence, forces the model to process the remaining elements in the second sequence (namely 'XSE') using the same winning neurons as for the first sequence.

While a state is formed based on similarities in future context, there may be cases where the future context, for the respective input symbols that make up the state, is dissimilar (Table 2). However, once a state been constructed, the future context in subsequent sequences containing that state will be processed in an identical manner, regardless of the future context itself. For example, when trained on the sequences in Table 2, the 'T' symbol from sequence 1 will form a state with the first 'S' symbol from sequence 2. This will result in both sequences 1 and 2 sharing the same winning neurons for their final three inputs (X S E). STORM will then be able to generalize this learnt state to its memorization of sequence 3, resulting in the same winning neurons being activated for the 'X X V V E' in test sequence 4 as in training sequence 3.

#	Training sequence
1	B T X S E
2	B T S X S E
3	B T X X V V E

#	Test sequence
4	B T S X X V V E

Table 2 – Generalization example. When trained on the first three sequences, STORM is able to construct a state between the ‘T’ in sequence 1 and the first ‘S’ in sequence 2. By generalizing this learnt state to its memorization of sequence 3, STORM is able to correctly process sequence 4 by activating the same winning neurons for the sub-sequence ‘X X V V E’ as would be activated in sequence 3.

5. Experiments

In order to quantify STORM’s grammar induction abilities, the model was applied to the task of predicting the next symbols in a sequence from the Reber grammar (Figure 1). Similar prediction tasks have been used in [8] and [3] to test the SRN’s grammar-induction abilities. The task involved presenting the model with symbols from a randomly generated sequence that was not encountered during training. The model then had to predict the next possible symbols in the sequence that could follow each symbol according to the rules of the grammar. STORM’s predictions are made by utilizing the locational representational values used in its context vector. As further explained in [1], the winning neuron for an input is the neuron whose weight vector best matches both the input symbol and the context representation of the last winning neuron’s location. STORM predicts the next symbol by finding the neuron whose context representation best matches that of the current winning neuron (i.e. the symbol part of the weight vector is ignored in the Euclidean distance calculation). This forces the model to find the neuron that is most likely to be the next winner. The symbol part of this neuron’s weight vector provides the next predicted symbol itself. This process is then repeated to find the second-best matching winner and the corresponding second predicted next symbol.

In accordance with established training criteria for artificial neural network models [17], the experiments were conducted on randomly generated separate training and test sets (i.e. sequences were unique with respect to all other sequences in both sets). Such an approach ensures that the model’s performance, assessed from the test set, is a true measure of its generalization abilities because the test sequences

were not encountered during training. The experiment was run ten times using models with randomly generated initial weights, in order to ensure that the starting state did not adversely influence the results.

The recursive depth parameter, as listed in Table 3, denotes the maximum number of sequential recursive transversals a sentence may contain (i.e. how many times it can go around the same loop). In order to ensure that the training and test sequences are representative of the specified recursive depth, the sets are divided equally between sequences of each recursive depth (i.e. a set of six sequences with a recursive depth (RD) of 2 will contain two sequences with an RD of 0, two sequences with an RD of 1 and two sequences with an RD of 2).

Parameter	Value
Number of epochs	1000
Learning rate α (linearly decreasing)	0.1
Initial neighbourhood σ (linearly decreasing)	5
Positive / negative temporal Hebbian learning rate	0.5 / 0.005
Number of training sequences	21
Number of test sequences	7
Maximum recursive depth (RD) of sequences	6
Model size	10×10

Table 3 - Experimental parameters for the first experiment

As shown in figure 5, six models learnt the grammar with over 89% accuracy during training and three of them became perfect grammar recognizers. However, this number fell by the end of training, with only two perfect models and an additional two models with over 90% performance accuracy. This equates to an average post-training performance of 71%. While less than half the models successfully learnt the grammar, it is worth noting that this is significantly better than for SRNs where Sharkey [18] showed that only two out of 90 SRNs became finite-state grammar recognisers in a similar experiment using the Reber grammar.

One of the proposed advantages of a discrete state-space model (page 3), is its ability to generalize to sequences longer than those encountered during training without the instabilities characteristic of standard DRN models. In order to test this proposition, a perfect finite-state recognizer (i.e. a model that scored 100%

prediction accuracy) from the first experiment (figure 5) was tested on a further three test sets. These sets contained sequences with recursive depths of 8, 10 and 12 and should constitute a much harder problem for any model trained only on sequences with a recursive depth of 6. These models that achieved 100% performance accuracy in the original experiments also achieved 100% accuracy on training sets with higher recursive depths. This proves that these models act as perfect grammar recognizers that are capable of generalizing to sequences of potentially any length.

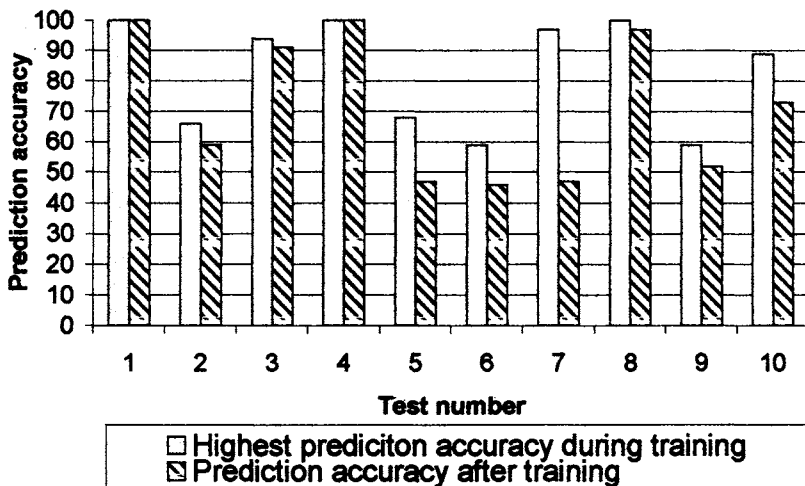


Fig 5 – Results from ten models trained on randomly generated separate training and test sets.

6. Conclusions and Future Work

We have presented a novel connectionist memory-rule based model capable of inducing the finite-state properties of an input language from a set of positive example sequences. In contrast with the majority of supervised connectionist models in the literature, STORM is based on an unsupervised recurrent SOM [1] and operates using a discrete state-space.

The model has been successfully applied to the task of learning the Reber grammar by predicting the next symbols in a set of randomly generated sequences. The experiments have shown that over half the models trained are capable of learning a good approximation of the grammar (over 89%) during the training process. However, by the end of training, only a fifth of the models were capable of operating as perfect grammar recognizers. This suggests that the model is unstable and that partial or optimal solutions reached during training may be lost by the end

of the training process. Despite this instability, a comparison between STORM and the SRN, when applied to a similar problem [3], shows that STORM is capable of learning the grammar perfectly much more often than its counterpart. Furthermore, experiments show that STORM's discrete state-space allow it to generalize its grammar recognition abilities to sequences far beyond the length of those encountered in the training set, without the instabilities experienced in continuous state-space DRNs.

Future work will initially involve analyzing the model to find where it fails. Once the model's abilities have been fully explored, its stability will be improved to increase the number of models that successfully become perfect grammar recognizers. STORM will then be enhanced to allow it to process more advanced grammars. Given that regular grammars are insufficient for representing natural language [19], the model must be extended to learn at least context-free languages if it is to be applied to real-world problems. However, despite such future requirements STORM's current ability to explicitly learn the rules of a regular grammar distinguish its potential as a language acquisition model.

References

1. McQueen, T. & Hopgood, A. & Tepper, J. & Allen, T. A Recurrent self-organizing map for Temporal Sequence Processing. In: Proceedings of 4th International Conference in Recent Advances in Soft Computing (RASC2002), Nottingham, 2002
2. Hopcroft J. & Ullman J. Introduction to Automata Theory, Languages and Computation, vol 1, Addison-Wesley, 1979
3. Cleeremans A, Schreiber D, McClelland J. Finite State Automata and Simple Recurrent Networks. In: Neural Computation. 1989, Vol 1, pp 372-381
4. Collier R. An historical overview of natural language processing systems that learn. Artificial Intelligence Review 1994; 8(1)
5. Chomsky, N. Aspects of the Theory of Syntax. MIT Press, 1965
6. Gold, E.M. Language Identification in the Limit. Information and Control 1967; 10:447-474
7. Horning, J.J. A study of grammatical inference. PhD thesis, Stanford University, California, 1969
8. Elman, J.L. Finding Structure in Time. Cognitive Science 1990; 14:179-211
9. Omlin, C. Understanding and Explaining DRN Behaviour. In: Kolen, J. and Kremer S (eds) A Field Guide to Dynamical Recurrent Networks. IEEE Press, New York, 2001, pp 207-227

10. Kolen, J. Fool's Gold: Extracting Finite State Machines From Recurrent Network Dynamics. In: Cowan J, Tesauro G and Alspector J (eds) *Advances in Neural Information Processing Systems 6*. Morgan Kaufmann, San Francisco CA, 1994, pp 501-508
11. Kohonen T. *Self-Organizing Maps*, vol 1. Springer-Verlag, Germany, 1995
12. Baretto, G and Arajo, A. Time in Self-Organizing Map: An Overview of Models. *International Journal of Computer Research: Special Edition on Neural Networks: Past, Present and Future* 2001; 10(2):139-179
13. Pinker, S. *Words and Rules*. Phoenix, London, 2000
14. Marcus, G. F. Children's Overregularization and Its Implications for Cognition. In: P. Broeder and J. Murre (eds) *Models of Language Acquisition: Inductive and Deductive approaches*. Oxford University Press, Oxford, 2000, pp 154-176
15. Cohen, N.J. and Squire, L.R. Preserved learning and retention of pattern-analyzing skill in amnesia: Dissociation of knowing how and knowing that. *Science* 1980; 21:207-210
16. Voegtlin, T. Recursive Self-Organizing Maps. *Neural Networks* 2002; 15(8-9):979-991
17. Hopgood, A. A. *Intelligent Systems for Engineers and Scientists*, 2nd edition, CRC Press LLC, Florida, 2001, pp 195-222
18. Sharkey N, Sharkey A, Jackson S. Are SRNs sufficient for modelling language acquisition?. In: Broeder P, Murre J. (eds) *Models of Language Acquisition: Inductive and Deductive Approaches*. Oxford University Press, Oxford, 2000, pp 33-54
19. Lawrence S, Giles C, Fong S. Natural Language Grammatical Inference with Recurrent Neural Networks. *IEEE Transactions on Knowledge and Data Engineering* 2000; 12(1):126-140

SESSION 1a:
AI TECHNIQUES I

Modelling Shared Extended Mind and Collective Representational Content

Tibor Bosse¹, Catholijn M. Jonker¹, Martijn C. Schut¹, and Jan Treur^{1,2}

¹Vrije Universiteit Amsterdam, Department of Artificial Intelligence

{tbosse, jonker, schut, treur}@cs.vu.nl

<http://www.cs.vu.nl/~{tbosse, jonker, schut, treur}>

²Universiteit Utrecht, Department of Philosophy

Abstract

Some types of animals exploit the external environment to support their cognitive processes, in the sense of patterns created in the environment that function as external mental states and serve as an extension to their mind. In the case of social animals the creation and exploitation of such patterns can be shared, thus obtaining a form of shared mind or collective intelligence. This paper explores this shared extended mind principle for social animals in more detail. The focus is on the notion of representational content in such cases. Proposals are put forward and formalised to define collective representational content for such shared external mental states. A case study in social ant behaviour in which shared extended mind plays an important role is used as illustration. For this case simulations are described, representation relations are specified and are verified against the simulated traces.

1. Introduction

Behaviour is often not only supported by internal mental structures and cognitive processes, but also by processes based on patterns created in the external environment that serve as external mental structures; cf. [5, 6, 7 & 8]. Examples of this pattern of behaviour are the use of ‘to do lists’ and ‘lists of desiderata’. Having written these down externally (e.g., on paper, in your diary, in your organizer or computer) makes it unnecessary to have an internal memory about all the items. Thus internal mental processing can be kept less complex. Other examples of the use of extended mind are doing mathematics or arithmetic, where external (symbolic, graphical, material) representations are used; e.g., [4 & 12]. In [16] a collection of papers can be found based on presentations at the conference ‘The Extended Mind: The Very Idea’ that took place in 2001. Clark [6] points at the roles played by both internal and external representations in describing cognitive processes: ‘Internal representations will, almost certainly, feature in this story. But so will external representations, ...’[6, p. 134]. From another, developmental angle, also Griffiths and Stotz [9] endorse the importance of using both internal and external

representations; they speak of ‘a larger representational environment which extends beyond the skin’, and claim that ‘culture makes humans as much as the reverse’ [9, p. 45].

Allowing mental states, which are in the external world and thus accessible for any agent around, opens the possibility that other agents also start to use them. Indeed, not only in the individual, single agent case, but also in the social, multi-agent case the extended mind principle can be observed, e.g., one individual creating a pattern in the environment, and one or more other individuals taking this pattern into account in their behaviour. For the human case, examples can be found everywhere, varying from roads, and traffic signs to books or other media, and to many other kinds of cultural achievements. Also in [17] it is claimed that part of the total team knowledge in distributed tasks (such as air traffic control) comprises external memory in the form of artefacts. In this multi-agent case the extended mind principle serves as a way to build a form of social or collective intelligence, that goes beyond (and may even not require) social intelligence based on direct one-to-one communication.

Especially in the case of social animals external mental states created by one individual can be exploited by another individual, or, more general, the creation and maintenance, as well as the exploitation of external mental states can be activities in which a number of individuals participate. For example, presenting slides on a paper with multiple authors to an audience. In such cases the external mental states cross, and in a sense break up, the borders between the individuals and become *shared extended mental states*. An interesting and currently often studied example of collective intelligence is the intelligence shown by an ant colony [2]. Indeed, in this case the external world is exploited as an extended mind by using pheromones. While they walk, ants drop pheromones on the ground. The same or other ants sense these pheromones and follow the route in the direction of the strongest sensing. Pheromones are not persistent for long times; therefore such routes can vary over time.

In [3] the shared extended mind principle is worked out in more detail. The paper focusses on formal analysis and formalisation of the dynamic properties of the processes involved, both at the local level (the basic mechanisms) and the global level (the emerging properties of the whole), and their relationships. A case study in social ant behaviour in which shared extended mind plays an important role is used as illustration.

In the current paper, as an extension to [3], the notion of *representational content* is analysed for mental processes based on the shared extended mind principle. The analysis of notions of representational content of *internal* mental state properties is well-known in the literature on Cognitive Science and Philosophy of Mind. In this literature a relevant internal mental state property *m* is taken and a representation relation is identified that indicates in which way *m* relates to properties in the external world or the agent’s interaction with the external world; cf. [1, 10 & 15, pp. 184-210]. For the case of extended mind an extension of the analysis of notions of representational content to *external* state properties is needed. Moreover, for the case of external mental state properties that are *shared*, a notion of *collective* representational content is needed (in contrast to a notion of representational content for a single agent).

Thus, by addressing the ants example and its modelling from an extended mind perspective, a number of challenging new issues on cognitive modelling and representational content are encountered:

- How to define representational content for an *external* mental state property
- How to handle *decay* of a mental state property
- How can *joint* creation of a *shared* mental state property be modelled
- What is an appropriate notion of *collective* representational content of a shared external mental state property
- How can representational content be defined in a case where a behavioural choice depends on *a number of mental state properties*

In this paper these questions are addressed. To this end the shared extended mind principle is analysed in more detail, and a formalisation is provided of its dynamics. It is discussed in particular how a notion of collective representational content for a shared external mental state property can be formulated. In the literature notions of representational content are usually restricted to internal mental states of one individual. The notion of collective representational content developed here extends this in two manners: (1) for external instead of internal mental states, and (2) for groups of individuals instead of single individuals. It is reported how in a case study of social behaviour based on shared extended mind (a simple ant colony) the proposals put forward have been evaluated. The analysis of this case study comprises multi-agent simulation based on identified local dynamic properties, identification of dynamic properties that describe collective representational content of shared extended mind states, and verification of these dynamic properties.

2. State Properties and Dynamic Properties

Dynamics will be described in the next section as evolution of states over time. The notion of state as used here is characterised on the basis of an ontology defining a set of physical and/or mental (state) properties that do or do not hold at a certain point in time. For example, the internal state property ‘the agent A has pain’, or the external world state property ‘the environmental temperature is 7° C’, may be expressed in terms of different ontologies. To formalise state property descriptions, an ontology is specified as a finite set of sorts, constants within these sorts, and relations and functions over these sorts. The example properties mentioned above then can be defined by nullary predicates (or proposition symbols) such as pain, or by using n-ary predicates (with $n \geq 1$) like `has_temperature(environment, 7)`. For a given ontology *Ont*, the propositional language signature consisting of all *state ground atoms* (or *atomic state properties*) based on *Ont* is denoted by *APROP*(*Ont*). The *state properties* based on a certain ontology *Ont* are formalised by the propositions that can be made (using conjunction, negation, disjunction, implication) from the ground atoms. A *state* *S* is an indication of which atomic state properties are true and which are false, i.e., a mapping $S: \text{APROP}(\text{Ont}) \rightarrow \{\text{true}, \text{false}\}$.

To describe the internal and external dynamics of the agent, explicit reference is made to time. Dynamic properties can be formulated that relate a state at one point in time to a state at another point in time. A simple example is the following dynamic property specification for belief creation based on observation:

‘at any point in time t_1 if the agent observes at t_1 that it is raining, then there exists a point in time t_2 after t_1 such that at t_2 the agent believes that it is raining’.

To express such dynamic properties, and other, more sophisticated ones, the temporal trace language TTL is used; cf. [11]. To express dynamic properties in a precise manner a language is used in which explicit references can be made to time points and traces. Here *trace* or *trajectory* over an ontology Ont is a time-indexed sequence of states over Ont. The sorted predicate logic temporal trace language TTL is built on atoms referring to, e.g., traces, time and state properties. For example, ‘in the output state of A in trace γ at time t property p holds’ is formalised by $\text{state}(\gamma, t, \text{output}(A)) \models p$. Here $\models$ is a predicate symbol in the language, usually used in infix notation, which is comparable to the Holds-predicate in situation calculus. Dynamic properties are expressed by temporal statements built using the usual logical connectives and quantification (for example, over traces, time and state properties). For example the following dynamic property is expressed:

‘in any trace γ , if at any point in time t_1 the agent A observes that it is raining, then there exists a point in time t_2 after t_1 such that at t_2 in the trace the agent A believes that it is raining’.

In formalised form:

$$\forall t_1 \ [\text{state}(\gamma, t_1, \text{input}(A)) \models \text{agent_observes_itsraining} \Rightarrow \\ \exists t_2 \geq t_1 \ \text{state}(\gamma, t_2, \text{internal}(A)) \models \text{belief_itsraining} \]$$

Language abstractions by introducing new (definable) predicates for complex expressions are possible and supported.

A simpler temporal language has been used to specify simulation models. This language (the *leads to* language) offers the possibility to model direct temporal dependencies between two state properties in successive states. This executable format is defined as follows. Let α and β be state properties of the form ‘conjunction of atoms or negations of atoms’, and e, f, g, h non-negative real numbers. In the *leads to* language $\alpha \rightarrow_{e, f, g, h} \beta$, means:

If state property α holds for a certain time interval with duration g , then after some delay (between e and f) state property β will hold for a certain time interval of length h .

For a precise definition of the *leads to* format in terms of the language TTL, see [14]. A specification of dynamic properties in *leads to* format has as advantages that it is executable and that it can often easily be depicted graphically.

3. Representation for Shared Extended Mind

Originally, the different types of approaches to representational content that have been put forward in the literature on Cognitive Science and Philosophy of Mind, [1, 13 & 15, pp. 191-193, 200-202] are all applicable to internal (mental) states. They have in common that the occurrence of the internal (mental) state property m at a