

Volume 27 Number 2 June 2003

ISSN 0350-5596

Informatica

**An International Journal of Computing
and Informatics**

Special Issue:

Perception and Emotion Based Reasoning

Guest Editor:

Aladdin Ayesh



The Slovene Society Informatika, Ljubljana, Slovenia

EDITORIAL BOARDS, PUBLISHING COUNCIL

Informatica is a journal primarily covering the European computer science and informatics community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Executive Editor – Editor in Chief

Anton P. Železnikar
Volaričeva 8, Ljubljana, Slovenia
s51em@lea.hamradio.si
<http://lea.hamradio.si/~s51em/>

Executive Associate Editor (Contact Person)

Matjaž Gams, Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Phone: +386 1 4773 900, Fax: +386 1 219 385
matjaz.gams@ijs.si
<http://ai.ijs.si/mezi/matjaz.html>

Executive Associate Editor (Technical Editor)

Drago Torkar, Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Phone: +386 1 4773 900, Fax: +386 1 219 385
drago.torkar@ijs.si

Rudi Murn, Jožef Stefan Institute

Publishing Council:

Tomaž Banovec, Ciril Baškovič,
Andrej Jerman-Blažič, Jožko Čuk,
Vladislav Rajkovič

Board of Advisors:

Ivan Bratko, Marko Jagodič,
Tomaž Pisanski, Stanko Strmčnik

Editorial Board

Suad Alagić (Bosnia and Herzegovina)
Vladimir Bajić (Republic of South Africa)
Vladimir Batagelj (Slovenia)
Francesco Bergadano (Italy)
Leon Birnbaum (Romania)
Marco Botta (Italy)
Pavel Brazdil (Portugal)
Andrej Brodnik (Slovenia)
Ivan Bruha (Canada)
Se Woo Cheon (Korea)
Hubert L. Dreyfus (USA)
Jozo Dujmović (USA)
Johann Eder (Austria)
Vladimir Fomichov (Russia)
Georg Gottlob (Austria)
Janez Grad (Slovenia)
Francis Heylighen (Belgium)
Hiroaki Kitano (Japan)
Igor Kononenko (Slovenia)
Miroslav Kubat (USA)
Ante Lauc (Croatia)
Jadran Lenarčič (Slovenia)
Huan Liu (Singapore)
Ramon L. de Mantaras (Spain)
Magoroh Maruyama (Japan)
Nikos Mastorakis (Greece)
Angelo Montanari (Italy)
Igor Mozetič (Austria)
Stephen Muggleton (UK)
Pavol Návrat (Slovakia)
Jerzy R. Nawrocki (Poland)
Roumen Nikolov (Bulgaria)
Franc Novak (Slovenia)
Marcin Paprzycki (USA)
Oliver Popov (Macedonia)
Karl H. Pribram (USA)
Luc De Raedt (Belgium)
Dejan Raković (Yugoslavia)
Jean Ramaekers (Belgium)
Wilhelm Rossak (USA)
Ivan Rozman (Slovenia)
Claude Sammut (Australia)
Sugata Sanyal (India)
Walter Schempp (Germany)
Johannes Schwinn (Germany)
Zhongzhi Shi (China)
Branko Souček (Italy)
Oliviero Stock (Italy)
Petra Stoerig (Germany)
Jiří Šlechta (UK)
Gheorghe Tecuci (USA)
Robert Trapp (Austria)
Terry Winograd (USA)
Stefan Wrobel (Germany)
Xindong Wu (Australia)

Guest Editorial

Special Issue on Perception and Emotion Based Reasoning

1 Introduction

It is a great pleasure to present a special issue on perception and emotion based reasoning in *Informatica*: An International Journal of Computing and Informatics. This special issue emerged from a successful special session in IASTED AIA2002 conference on the subject of perception and emotions. It became evident during the discussions within the session that there is increasing research in computer modelling of cognitive aspects of human reasoning, hence this special issue.

In our special issue we have 10 papers. These papers were selected by peer review from 15 submitted papers of high quality. The review process was long and strenuous.

The papers of this special issue cover the aspects of modelling perception and emotions for reasoning about knowledge, environment, actions or a combination of these elements. Some of these papers are application-oriented whilst others are concerned with theory and algorithmic development. The techniques used are as varied. We see connectionist approaches, symbolic and hybrid techniques used.

In the first paper, ordered alphabetically, Ayesh uses cognitive maps, a technique increasingly used by agents and robotics researchers, to represent the relationship between emotions, perceptions and objects. Formulas presented enable the mapping of the environment objects to perceptual and emotional models. Using this mapping, given perceptions are used to infer the agent's emotions towards a detected object and thus its reaction towards that object. An algorithmic translation of these formulas is provided.

In the second paper, Byl studies the influence of emotions on perception. The result is the Emotionally Motivated Artificial Intelligence (EMAI) architecture. Within this architecture, Byl provides us with an interesting study of different emotions and related cognitive aspects such as pleasantness and responsibility that influence the agent's decision. A well-formulated study of emotions and their effects on perception is presented in conclusion.

In the third paper, Lucas *et al.* deploy emotions as a technique to enhance neuro-fuzzy predicators. This paper is concerned with the use of neural nets for prediction. As neural nets require training, this training is achieved through an emotionally motivated learning algorithm. The results are demonstrated with comparisons to other, non-emotions based, neural nets and neuro-fuzzy models.

In the fourth paper, Damas and Custódio merge neural networks and statistical methods to develop emotion-based decision and learning algorithms. Their interest is to develop an adaptive control system for

robots and agents comprising memory resources and actions management subsystems.

In the fifth paper, Davis and Lewis present in their paper some computational modeling of emotions. The paper's background comes from psychology with a focus on the relationships between emotions, goals and autonomy. These relationships are established through a 4-layered architecture. The technique used to represent emotions and provide the inferencing mechanism is symbolic.

In the sixth paper, Fatourehchi *et al.* present an emotions-based learning algorithm that uses emotional critics to direct the learning process. In contrast with the Damas and Custódio paper, the Fatourehchi *et al.* technique uses fuzzy controllers as the basis to develop a control mechanism within and using multi-agents systems. Examples, simulation results and their analysis are provided.

In the seventh paper, Gadanho and Custódio provide a contrasting view to the use of emotions in learning and robot control. In their paper, reinforcement learning is revisited in the light of emotions. Gadanho and Custódio present an interesting architecture that consists of adaptive system, perceptual system, behaviour system and goal system. The objective is to deal with tasks comprising multi-goals.

In the eighth paper, Maçãs and Custódio extend the DARE architecture for multi emotion-based agents. The architecture consists of symbolic analysis, cognitive analysis and perceptual analysis layers as its main components. The architecture is developed into a multi-agents environment and results of experiments are presented.

In the ninth paper, Neal and Timmis pose a question about the usefulness of Timidity as an emotional mechanism to control robots. This mechanism is biologically motivated and uses neural networks for modelling. Comparisons with traditional neural nets are made and experimental results are presented.

In the last paper, Rzepka *et al.* use emotions in information retrieval agents. They present an interesting application of emotions contrasting with other papers. The emotional agents are used over the Internet to assist in searching the World Wide Web. It attempts to provide human-like interfacing, using techniques such as conversation, for user profiling. Experiments and results are presented.

There are a variety of applications and techniques presented in the papers of this special issue. Some techniques are developed especially for cognitive modeling whilst others are re-working of established and proven techniques. Robots and agents, understandably,

seem to dominate the experimental side of most of the presented papers. This special issue is targeted for researchers working in cognitive modeling and human-like machines – e.g. cognitive and humanoid robots and intelligent embedded agents. It would also be of interest to software engineers who are developing alternative solutions using artificial intelligence techniques.

Acknowledgment

The guest editor would like to express his appreciation to the participant authors for their high quality work. Also many thanks go to the referees: Franz Kurfess, Zhongzhi Shi, George Tecuci, Marco Botta, Se Woo Cheon, Ralf Birkenhead, Peter Innocent, Jenny Carter, and John Cowell, for their careful consideration and reviewing of papers in support of this special issue. Finally, the guest editor is indebted to Marcin Paprzycki and Matjaz Gams for their support and guidance during the preparation of this special issue.

Aladdin Ayesh

Perception and Emotion Based Reasoning: A Connectionist Approach

Aladdin Ayesh

Email: aayesh@dmu.ac.uk

Center for Computational Intelligence,

De Montfort University, The Gateway, Leicester LE1 9BH UK

Keywords: Connectionism, Cognitive Maps, Perception, Emotions, Automated Reasoning, Reasoning about Actions.

Received: October 18, 2002

Our reasoning process uses and is influenced by our perception model of the environment stimuli and by our memorization of related experiences, beliefs, and emotions that are associated with each stimulus, whilst taking in considerations other factors such as time and space. These two processes of modelling and memorization happen in real time while interspersing with each other in a manner they almost seem as if they are one process. This is often referred to as cognition. In this paper we provide a simplified model of this complicated relationship between emotions, perceptions and our behaviour to produce a model that can be used in software agents and humanized robots.

1 Introduction

Human reasoning process is influenced by the perception model of the environment stimuli that humans often develop based on the memorization of related experiences, beliefs, and emotions that are associated with each stimulus, whilst taking in considerations other factors such as time and space. These two processes, i.e. modelling and memorization, happen in real time while interspersing with each other in a manner so they almost seem as if they are one process. This is often referred to as cognition.

In this paper, we provide a simplified model of this complicated relationship between emotions, perceptions and our behaviour to produce a model that can be used in software agents and humanized robots. To do so, we deploy some aspects of psychology [1-4] and cognitive maps [5-7]. However, effective computational modelling of our cognitive faculties is very difficult. This did not prevent several attempts to provide reasoning systems that enable us to reason about actions and effects [8-11], about objects and time [9, 12-15], decisions and concepts [16-21] and so on. Many of these attempts suffered either from limitation in modelling [22] or limitation in practical inferencing [23, 24]. However, we can divide these attempts into two main types: connectionist approaches and logicians' approaches. In this classification, we exclude the attempts that may have been successful in their respective domains (e.g. [25]) without formal theory or formalized explanation that may lead to some generalization.

The main body of the paper examining inferencing under a set of perceptions and associated emotions to trigger eventually a reaction in response to environment's objects or stimuli. Consequently, a

connectionist approach is presented here to develop a reasoning mechanism that models and use perceptions and emotions factors. It is a connectionist approach in the sense that it is based on Fuzzy Cognitive Maps [5, 6] to represent the environment and relations between different components that define this environment such as objects, concepts, and features. The meaning of these relations between the different components is inter-related to perceptions and emotions. The result is presented as Triangular Object Modelling (TOM) technique. The inferencing process is then studied and analysed. We conclude with a critical analysis of the technique highlighting future developments.

2 Modelling Techniques

Studying human's mind and cognitive faculties, e.g. recognition and learning, may take two routes [26]. The first route is to look at the mind as a symbols manipulation processor (e.g. [10, 27]). The second route is to view it as a signal processor formed of a web of connections [18, 26].

2.1 Symbol Manipulation Techniques

The symbolic route, which is computationally represented by symbolic artificial intelligence, is dominated by the logic approach. In this approach, logic theories [28-31] are used to explain and imitate our thinking and learning abilities by means of knowledge formation and inferencing [4, 8, 26, 32, 33].

Knowledge formation is often found in the form of logical theories of belief and knowledge [32, 34, 35]. It takes a philosophical approach to formally represent and reason about the notions of belief and knowledge. However, several computational intelligence researchers are currently eschewing more in favour of connectionist approaches or hybrid approaches such as fuzzy logic,

belief networks, and cognitive maps [5, 18] because of the computational complexity of logical models, which limits the applicability of such models.

Inferencing may be presented in the form of problem solvers [36] or in the form of AI planning [11]. These two fields are the more traditional of symbolic processing fields. Situation calculus is a well-known example of formal AI planning languages [10]. In addition, situation calculus is a good example of the difficulty of using formal logic as a representational and reasoning tool [11, 24].

2.2 Signal Processing Techniques

The signal-processing route [18, 26], which is often referred to as the connectionists approach, is based on the neuro-psychology and neuro-biology fields [2, 37, 38]. The study of the brain shows that it is formed of several millions of neuron cells connected to each other. Signals pass through these neurons producing different results. The connectionist approach aims to produce models of that web of neural connection to model our knowledge and inference processes.

Connectionist models often require high computational resources to produce useful results [39]. However, the current surge in computational power provided with the development of fast processors and cheap fast memory enables such models to be deployed. This may also explain the increasing interest in network based models such as neural nets and cognitive maps.

2.3 Cognitive Maps

Cognitive maps are graphical and formal representation of crisp cause-effect relationships among the elements of a given environment [5, 40]. They were originated in economics and political sciences. However, they are becoming increasingly popular with computational intelligence researchers especially a version blended with fuzzy logics [5-7, 40]. They are similar to neural nets in the sense that they consist of nodes that are linked together. However, they differ from neural nets, and for that matter from other graph-based approaches in the fact that they represent semantically defined relationships. The Fuzzy Cognitive Maps (FCM) version is represented in the form of fuzzy signed directed graphs with feedbacks. They model the world as a collection of concepts and causal relations between these concepts [5, 6]. This provides, in our opinion, the middle ground between the pure connectionist and the symbolic AI approaches. However, there are only a few studies in formal representation and inferencing using cognitive maps [5, 7, 41]. In this paper, we adapt a version of cognitive maps to enable our proposed Triangular Object Modelling (TOM) in which perceptions and emotions are interconnected to objects and factored in the reasoning mechanism that is explained in section 4.

3 Triangular Object Modelling (TOM)

Our proposed technique represents each object that may reside in the memory of an agent as a triangular multi-layered cognitive map forming a Triangular Object Modelling (TOM). This triangular object modelling is multi-layered in the sense that each node of this cognitive map may consist of a cognitive map. This representation is used within a memory architecture we named Observer-Memory-Questioner (OMQ).

3.1 OMQ Memory

TOM representation is used within the memory component of OMQ (Observer-Memory-Questioner) model, which was developed and presented in previous papers [42, 43]. The memory subsystem consists of short, surface, long and archive memory components. The design of these components was inspired by psychology research on human memory and its workings [1, 44-46].

The human memory [45, 46] is often represented as consisting of two components: short (or working) memory and long memory. In our design, we felt the need to introduce two additional components, namely surface memory and archive memory. The task of these two new memories is in support of short memory and long memory respectively. Archive memory represents the long-long memory, which is a step prior to ‘forgetfulness’. On the other hand, surface memory is the background part of a working memory in which short memory is the frontier. In other words, surface memory is where all relevant information and experiences related to objects in the short memory to be stored. Whilst short memory would only contain information and experiences directly related to the objects of current interest to the agent or robot.

The detailed description of these components and their workings was covered in [47] from which we borrow figure 1.

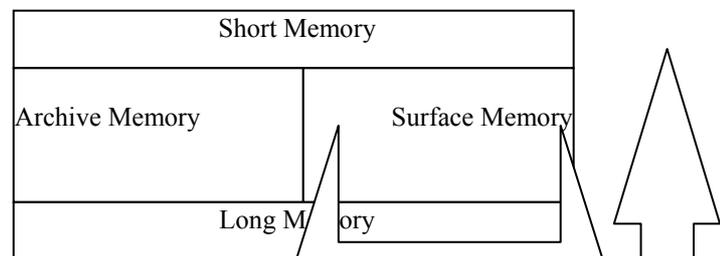


Figure 1 – Memory

The following is a summary of the definitions.

Definition 1 – Short memory is the part of the mind in which current objects of interest are held with limitation of time and/or capacity.

Definition 2 – Long memory is the part of the mind in which objects’ information is stored in relation to concepts, emotions and/or re-

evance
a men
aniza
tained
e. ♦

Definition 3 – Surface memory is a mental organization of mind in which relevant information to current objects of the short memory is maintained. The organization of surface memory is prioritised according to time, emotions and/or relevance. ♦

Definition 4 – Archive memory is a mental organization of objects that is the result of a re-organization of the long memory in which objects’ information is re-categorized and either maintained in relation to a concept or deleted. Consequently, the archive memory information has a low priority in the retrieval process. ♦

We discussed these components in further details in previous papers [47, 48]; however, Observer and Questioner components do not have complete implementation.

3.2 TOM Architecture

TOM architecture is based on a technique that represents each object, which may reside in the memory of an agent, as a triangular multi-layered cognitive map. The three main nodes in that map are: object, perceptions and emotions.

Each node may consist of one or more cognitive maps. The nodes in each of these maps are the elements that belong to one of the primary nodes. As an example, emotions are formulated from cognitive maps that links between different types of emotions such as safety-fear, like-desire, and so on. We formalize figure 1 in definition 5.

Definition 5 – an object may be defined using TOM model as a tuple (P, E) where P is a set of perceptions and E is a set of emotions in which:

$$P = (p1/\mu1, p2/\mu2, \dots, pn/\mu n) \text{ and } p1 \cap p2 \cap \dots \cap pn = \emptyset; \text{ and}$$

$$E = (e1/\mu1, e2/\mu2, \dots, en/\mu n) \text{ and } e1 \cap e2 \cap \dots \cap en = \emptyset; \text{ and}$$

$$\sum E \cap P \neq \emptyset. \blacklozenge$$

Definition 6 – given two objects Obj1 and Obj2 if $Obj1 \cap Obj2 \neq \emptyset$ we say Obj1 and Obj2 are related to each other with strength equal to $\Sigma(P_{Obj1} \cap P_{Obj2}) \cup (E_{Obj1} \cap E_{Obj2})$.

The strength of relationships is identified by a fuzzy value. The fuzzy value is driven from an extended interval [-1 0 1], unlike the standard fuzzy interval. Having a minus value of strength indicates that the relationship is of a type ‘opposite’ whilst having 1 indicates that the two nodes within the map are effectively the same. These facts are used to optimize the map by eliminating irrelevant or duplicated links. Next, we present two assumptions ‘the weakest link’ assumption and ‘face off’ assumption.

Definition 7 – the weakest link assumption: we identify the weakest link to be a link with strength less than 0 in which case the link is eliminated.

Definition 8 – the face off assumption applies only to objects that have a relationship strength of one between them. In this case, these objects are the same and they would be merged by a link of strength one. The links of

one of these objects to its emotions and perceptions will be dropped, as they are accessible through its equivalent.

The determination of the link strength is currently done by presetting a threshold by the user. Work is being carried out to enhance the system with fuzzy-genetic algorithms to enable automated updates and optimization.

In figure 2, the model will receive as an input a set of perceptions, which will trigger a set of objects and emotions. However, objects are also associated with emotions and therefore may trigger emotions that were not triggered by the input perceptions. These emotions may activate some perceptions that can be used to guide the reaction of the model. At the current stage, there is not a great use of this fact. To facilitate this fact an imagination model needs to be developed. Such a model will enable TOM-based agents to build models of consequences and to construct future perceptions and emotions. Thereafter, these models can be used to derive action selection and behaviour. This leads to the question of inferencing that we cover next.

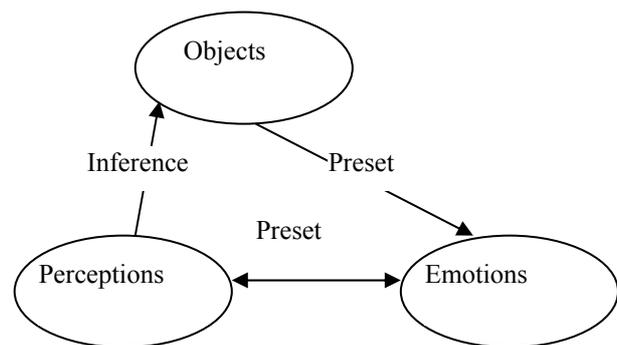


Figure 2 TOM’s basic model.

3.3 Pedagogic domain

A pedagogic domain has been devised here to help in demonstrating the TOM system and it’s working. To simplify matters, we define a fixedly preset set of emotions, perceptions and objects. We limit these to the minimum.

Firstly, emotions are fixedly preset to ‘safe’, ‘like’, and ‘desired’. ‘Safe’ reflects the stress levels. ‘Like’ reflects attraction levels. ‘Desired’ reflects goal-oriented attention levels. Each of these emotions is defined as a set of two values: ‘low’ and ‘high’. These two values can be defined in terms of exact or fuzzy sets.

Perceptions are fixedly preset to three feature types that are ‘size’, ‘colour’, and ‘shape’. Each one of these features takes one value from fixedly preset values. The collection of all these values describes the object to which these values relate. These preset values are as follows:

Size (Si) = {large, small, medium}

Colour (Co) = {bright, dark, grey}

Shape (Sh) = {4-edged, 3-edged, many-edged, uneven}

The values of these sets can be defined as fuzzy sets. This is discussed further in the implementation section on extending TOM using fuzzy sets.

The artificial world domain constructed here contains three types of objects: ‘predator’, ‘box’ and ‘food’. ‘Predator’ is perceived to be dangerous to our robot or agent. It is identified by being large with uneven shape. ‘Box’ is a desirable object, which is used by the robot to build a refuge from the predator. ‘Food’ is an essential part for the robot to sustain it’s self. Figure 3 shows an example of the map that may be constructed to describe the object predator.

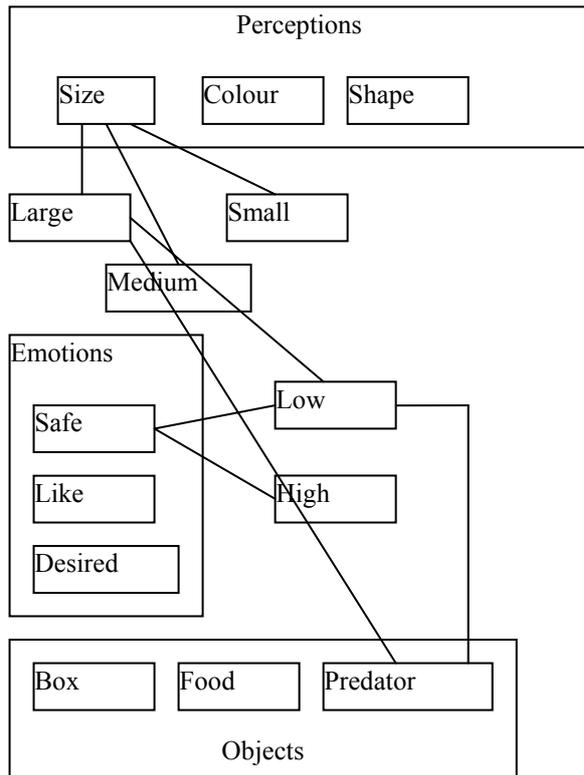


Figure 3 Predator’s map example.

The perception of size refers to the perception of the predator being large from the agent’s viewpoint rather than the actual physical size. Similarly, the emotion of safe refers to the feeling of being safe or less safe indicated by the level of emotional stress the agent may feel during the course of interacting with the environment’s object, which is in this case the predator.

Figure 3 can be summarized as follows:

$$\begin{aligned}
 P &= \{Si, Co, Sh\} \\
 Si &= \{La, Sm, Me\} \\
 Co &= \{Br, Da, Gr\} \\
 Sh &= \{F4e, T3e, M0e, U0e\} \\
 E &= \{Sa, Li, De\} \\
 Sa &= \{Lo, Hi\} \\
 Li &= \{Lo, Hi\} \\
 De &= \{Lo, Hi\} \\
 O &= \{Bo, Fo, Pr\}
 \end{aligned}$$

$$\begin{aligned}
 Bo &= \{(U, Br \vee Gr, F4e \vee F3e \vee M0e), \\
 &\quad (U, Hi, Hi)\} \\
 FO &= \{(Sm \vee Me, U, U), (Hi, Hi, U)\} \\
 Pr &= \{(La, Da, M0e \vee U0e)\}
 \end{aligned}$$

Note that each object is defined by set of Perceptions and a set of Emotions. Perceptions are represented as a tuple of values (Si, Co, Sh).

Each of the preset perceptions values connect to a preset emotion(s) as an example ‘large-size’ links to ‘low-safe’. A fuzzy representation and membership function will be used to determine these links and their strength in the adaptive version of TOM.

There is a globally defined value in the system, which is the undefined (U) value. This value is used, and consequently assigned, in the cases where none of the values given in the system definition applies. In other words, it defines the system ignorance of the presented information [49] or the information is unpredicted as it is the case of the box size in the given example where none of the possible size values takes a precedent over others. The semantics of this value is not our concern at this stage; therefore, we will assume it follows the Bochvar logic semantics as presented in [49].

4 Inferencing in TOM

Inferencing in TOM architecture relies on the perception input in determining two sets of intermediate outputs that are: set of objects and set of related emotions. The result is a set of pairs in which one is an object and the other is a related emotion. Each object may have more than one emotion associated. In this section, we discuss the inferencing process within TOM and its implementation.

4.1 Inferencing engine

In definition 9, we provide the basic definition of inferencing within TOM. We use the entailment operator $| -$ to notate the inferencing operation. Two forms of this entailment operator are used, we may refer to the first as a free form $| -$ in which inferencing is done over factors free from association. The second form is a bound form $| -_p$ in which the inferencing is done under the binding element that is P in this case.

Definition 9 – Given a set of input perceptions P if TOM $| - P$ then TOM $| -_p (O, E)$ where O is a set of objects and E is a set of emotions. ♦

Definition 9 states that no inferencing can be done unless the input set of perceptions P is derivable from the model TOM. If so, then under the set of perceptions P TOM can entail a tuple of objects and emotions (O, E). We needed this condition as we limit our system to the preset perceptions. In future development we hope to waive this restriction. The following corollary clarifies the meaning of TOM $| -_p (O, E)$ further.

Corollary 1 – Given a set of input perceptions P if TOM $|_{-p}$ (O, E) it means:

$$\forall p \in P. ((\exists o \in O \vee \exists e \in E) \vee (\exists o \in O \wedge \exists e \in E))$$

whereby: $p \rightarrow o$; $p \rightarrow e$

Where \rightarrow identifies a connection between two map elements. ♦

It is clear that to implement this process some rules and restrictions need to be established. These rules aim to counter the possibilities of having the derived set of objects or derived set of emotions to be empty. The most likely is that the combination of the existing set of perceptions and their emotions do not lead to any objects. That means the system is encountering a new type of object. Rule 1 is countering this case by initiating a new object.

Rule 1 Initiating new Object – Given $P \neq \emptyset$, if $|_{-p} O = \emptyset$ and $|_{-p} E \neq \emptyset$ then new O to be asserted and to be associated with P and E. ♦

The system may encounter two conflicts: object conflict and emotion conflict. Definition 6 and rule 2 define and resolve the object conflict. Definition 7 and rule 3 define and resolve the emotion conflict.

Definition 10 Object Conflict – Given $P \neq \emptyset$, if $|_{-p} O \neq \emptyset$ with cardinal greater than 1 and $\exists p1 \in P$ and $\exists p2 \in P$ whereby $\exists o1 \in O$ and $\exists o2 \in O$ we say there is an object conflict if the following conditions are true: $p1 \rightarrow o1$; $p2 \rightarrow o2$; $o1 = \neg o2$. ♦

Rule 2 Object Conflict Resolution – If there is an object conflict, determine the strength of connection and choose the highest strength in determining the derived object. ♦

Determining the strength of connection depends on various factors. At the current stage, we use a type of a fuzzy membership grade to determine the strength of individual connections and then choose the highest degree of membership.

Definition 11 Emotion Conflict – Given $P \neq \emptyset$, if $|_{-p} E \neq \emptyset$ with cardinal greater than 1 and $\exists p1 \in P$ and $\exists p2 \in P$ whereby $\exists e1 \in E$ and $\exists e2 \in E$ we say there is an emotion conflict if the following conditions are true: $p1 \rightarrow e1$; $p2 \rightarrow e2$; $e1 = \neg e2$. ♦

Rule 3 Emotion Conflict Resolution – If there is an emotion conflict, determine the strength of connection and choose the highest strength in determining the derived emotion. ♦

In TOM, we cannot have a perception that does not have an emotion associated to it. However, some perceptions may not trigger any particular emotions on their own. These perceptions will have a neutral emotion, which we view as zero value of the perception-emotion based system.

Definition 12 – Neutral emotion may be defined as zero value of emotions. In other words, if an object O or a perception P is not connected to an emotion, then we infer that the emotion is neutral. ♦

Rule 4 Neutral Emotion – Given $P \neq \emptyset$ if $\exists p \in P. \forall e \in E. \neg(p \rightarrow e)$ then $p \rightarrow N$ ♦

Theorem 1 - Given $P_i \neq \emptyset$, $|_{-p} E = \emptyset$ iff TOM $|_{-P} = \emptyset$. ♦

Proof – The proof of this theorem is intuitive and can be derived directly from Rule 4.

Corollary 2 – Given $P \neq \emptyset$, if $|_{-p} O = \emptyset$ it is not necessary the case that $|_{-p} E = \emptyset$. ♦

4.2 Implementation

In implementing TOM we define four classes. TOM agent acts as a controller that initiates the other parts of the TOM architecture. The other three classes are Perceptions, Emotions and Objects. These three classes are effectively cognitive maps with identifiers.

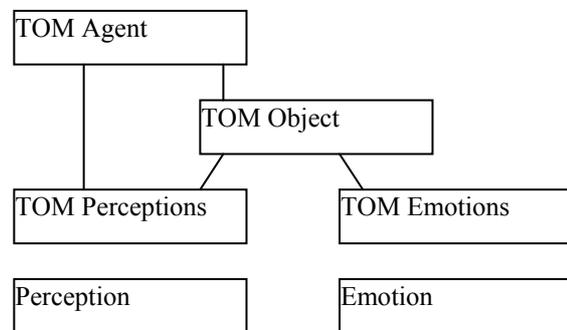


Figure 4 TOM class diagram.

Notice that TOM Perceptions and TOM emotions are composed of one or many perception and emotion objects respectively. TOM agent has five main operations that are initiate (I), initiate Object (IO), establish (E), assign (A) and derive (D).

Initiate (I) is the agent constructor. Its job is to initiate the agent with the provided perceptions and emotions and provoke object initiation and assignment operators.

Initiate Object (IO), establish (E) and assign (A), operations are used in handling new objects that are to be introduced into the system (Rule 1) or to a TOM-agent. It may be worth mentioning here that even though agents are used, which allows potentially for multi-agent systems to be developed, the current implementation only uses one agent. Multi-agent implementation will lead to several questions in relation of how these agents communicate, how would they be used in multi-agent controller for robots and so on.

Derive (D) operator is the Inferencing operator by which the system retrieve the connected perceptions, objects and emotions given one of them, i.e. object, perceptions or emotions. Algorithm 1 shows how these operations may be used together to initiate a new object within the system and to associate that object to the relevant perceptions and emotions.

Algorithm 1 – Initiate Agent – Given agent TOM-A the TOM-A operator initiates the agent as follows:

```
TOM-A.IM = E(E, P);
TOM-A.OM = IO(). ♦
```

Algorithm 2 – Establish emotions and perceptions E

```
E(E, P) :-
  For every p ∈ P
  For every e ∈ E
  Request w
  E(E, e);
  M(e, p) |- L ⊗ w;
  Return |-> M. ♦
```

Algorithm 3 – Initiate Object IO

```
Given P ≠ ∅, if P.D (O) = ∅ and P.D (E) ≠ ∅ then:
I(new_O);
Mo:
A(new_O, E);
A(new_O, P);
Return |-> Mo. ♦
```

Algorithm 4 – Assign Object A

Given an object O and set of features F, which can be either a set of emotions or perceptions, A(new_O, F) will request a set of weights of relevance W and link between O and F as follows:

```
For every f ∈ F
Request (w ∈ W)
Assert (O, f) |- L;
// L is a link tag identifying the link between O and f.
Assert (L, w). ♦
```

Algorithm 5 – Inferencing operation – D

```
Given a set of perception P
For every p P
  Do until p.connection (O) is empty
  {
  //O is the set of known objects
  Select (O,p) |- p.objects;
  }
  Do until p.connection (E) is empty
  {
  //E is the set of known emotions
  Select (E,p) |- p.emotions;
  }
  If p.objects contains more than one member
  Then Resolve-Object (p.objects, p.emotions, p);
  If p.emotions contains more than one member
  Then Resolve-Emotion (p.objects, p.emotions, p). ♦
```

5 Future Work

The system is by no means complete. The weaknesses lay in the restrictions we imposed on it. First, the system, at its current stage, cannot learn new emotions or perceptions. Secondly, it does not count for concepts and the more complicated knowledge structure proposed by OMQ model and our memory architecture [42, 43, 47]. However, TOM architecture answers the question of

inferencing to some degree, which was not discussed in any of the previous work.

5.1 Extending TOM Representation

In terms of representation, there are several extensions to be made. Firstly, a fuzzy representation using a modified version of fuzzy cognitive maps will be implemented. That will be extended later on to an adaptive version where the links are constructed, deconstructed and updated according to the robot perception of the environment. Problems to be addressed is the addition of perceptual experiences including definition of perceptions, objects and their associated emotions.

5.2 Extending TOM Inferencing

In this paper, we focused on the inferencing operator in terms of identifying objects from perceptions and their associated emotions. As an extension on this work, a ‘perception and emotions’ based planner is to be constructed and consequently tested on robots. This would require the extension of TOM inferencing mechanism to enable the determination of actions to be executed given set of perceptions and the inferred objects and emotions. Other practical problems need to be addressed such as the limitation of sensors on the robots used.

5.3 Completing OMQ System

TOM and its inferencing mechanism will be fed back into the completion of OMQ architecture design and implementation. Subsequently, the TOM model should be able to deal with more complicated knowledge structures. Sensors fusion is currently one of the problems to be addressed in completing OMQ system. In addition, TOM needs to be extended to allow learning of perceptions and emotions in order to widen its use [43].

6 Conclusion

In this paper, we attempted to answer the Inferencing question that emerged from previous work [42, 43]. Consequently, we presented Triangular Object Modelling (TOM) as a way of modelling objects in relation to perception and emotion. As a result, a connectionist approach using a modified version of cognitive maps has been developed to provide an inferencing method that utilises perceptions and emotions. Implementation of TOM and future developments was discussed.

7 References

- [1] A. Baddeley, *Working Memory*. Oxford: Clarendon Press, 1986.
- [2] R. D. Gross, *Psychology: The Science of Mind and Behaviour*. London, UK: Hodder & Stoughton, 1992.
- [3] H. C. Lindgren, *Psychology, an introduction to a behavioral science*. New York, London: Wiley, 1971.
- [4] A. L. Wilkes, *Knowledge in Minds: Individual and Collective Processes in Cognition*. UK: Psychology Press (of Erlbaum(uk) Taylor & Francis), 1997.
- [5] C. Carlsson and R. Fuller, "Adaptive Fuzzy Cognitive Maps for Hyperknowledge Representation in Strategy Formation Process," presented at International Panel Conference on Soft and Intelligent Computing, 1996.
- [6] B. Kosko, "Fuzzy Cognitive Maps," *International Journal of Man-Machine Studies*, pp. 65-75, 1986.
- [7] M. P. Wellman, "Inference in Cognitive Maps," *SIAM Journal on Computing*, vol. 36, pp. 1 - 12, 1994.
- [8] R. C. Moore, "A Formal Theory of Knowledge and Action," in *Formal Theories of the Commonsense World*, J. R. Hobbs and R. C. Moore, Eds. Norwood, New Jersey: Ablex Publishing Corporation, 1985.
- [9] J. F. Allen, "Towards General Theory of Actions and Time," in *Readings in Planning*, J. Allen, J. Hendler, and A. Tate, Eds. San Mateo, CA, USA: Morgan Kaufmann Publishers, INC., 1990/1984, pp. 464-479.
- [10] J. McCarthy and P. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence," in *Readings in Planning*, J. Allen, J. Hendler, and A. Tate, Eds. San Mateo, CA, USA: Morgan Kaufmann Publishers, INC., 1990, pp. 393-435.
- [11] A. Ayeshe, "An Investigation Into Formal Models Of Change In Artificial Intelligence," in *School of Computing and Mathematical Sciences*. Liverpool, UK: Liverpool John Moores University, 1999.
- [12] A. Ayeshe and G. Kelleher, "Aspects of Temporal Usability Via Constraint-Based Reasoning: The Use of Constraints as a Debugging Mechanism Within General Descriptions of The User's Possible Actions," presented at The IASTED International Conference on Artificial Intelligence and Soft Computing., Cancun, Mexico, 1998.
- [13] Y. Zhang and H. Barringer, "A Reified Temporal Logic For Nonlinear Planning," Manchester University, Manchester, Technical Report UMCS-94-7-1, July 1994.
- [14] A. Cesta and A. Oddi, "A Formal Domain Description Language for a Temporal Planner," presented at 4th Congress of the Italian Association For AI, Florence, Italy, 1995/1996.
- [15] I. Meiri and J. Pearl, "Temporal Constraint Network," *AI 49*, pp. 61-95, 1991.
- [16] C. Carlsson and R. Fuller, "Fuzzy Multiple Criteria Decision Making: Recent Developments," *Fuzzy Sets and Systems*, vol. 78, pp. 139 - 153, 1996.
- [17] N. Guarino, "Concepts, Attributes, and Arbitrary Relations," in *Some Linguistic and Ontological Criteria for Structuring Knowledge Bases*: citeseer.nj.nec.com/172862.html.
- [18] P. R. Van Looche, *The Dynamics of Concepts: A Connectionist Model*. Berlin: Springer-Verlag, 1991.
- [19] A. Doan, "Modeling Probabilistic Actions for Practical Decision-Theoretic Planning," presented at The Third International Conference on Artificial Intelligence Planning Systems (AIPS 96), Edinburgh, Scotland, 1996.
- [20] J. Pearl, "From Conditional Oughts to Qualitative Decision Theory," presented at Uncertainty in AI 9th Conference, USA, 1993.
- [21] P. Haddawy and S. Hanks, "Utility Models for Goal-Directed Decision-Theoretic Planners," University of Washington, Seattle, USA, Technical Report 93-06-04, June 15, 1993 1993.
- [22] V. Lifschitz, "On the Semantics of STRIPS," in *Readings in Planning*, J. Allen, J. Hendler, and A. Tate, Eds. San Mateo, CA, USA: Morgan Kaufmann Publishers, INC., 1990, pp. 523-530.
- [23] R. Reiter, "The Frame Problem in the Situation Calculus: A Simple Solution (Sometimes) and a Completeness Result for Goal Regression," in *Artificial Intelligence and Mathematical Theory of Computation: Papers in the Honor of John McCarthy*, V. Lifschitz, Ed. San Diego: Academic Press, INC.; Harcourt Brace Jovanovich, Publishers, 1991, pp. 359-380.
- [24] R. Reiter, "Proving Properties of States in The Situation Calculus," *AI 64*, pp. 337-351, 1993.
- [25] P. Morris and R. Feldman, "Automatically Derived Heuristics For Planning Search," presented at AICS '89, Dublin City University, 1989.
- [26] A. C. Grayling, "Philosophy 2: Further Through the Subject," . New York, USA: Oxford University Press, 1998.
- [27] B. Smith and D. W. Smith, "The Cambridge Companion to Husserl," . Cambridge, UK: Cambridge University Press, 1995.
- [28] J. Dix, J. E. Posegga, and P. H. Schmitt, "Modal Logics For AI Planning," presented at First International Conference On Expert Planning Systems, Brighton, UK, 1990.
- [29] S. Haack, *Philosophy of Logics*. Cambridge: Cambridge University Press, 1978.
- [30] N. Rescher, *Many-Valued Logic*. New York: McGraw-Hill, 1969.

- [31] Y. Murakami, *Logic and Social Choice*. London, New York: Routledge & Kegan Paul Ltd. Dover Publications Inc., 1968.
- [32] R. Turner, *Truth and Modality for Knowledge Representation*. London: Pitman Publishing, 1990.
- [33] M. Ayers, *Locke: Epistemology & Ontology*. London & New York: Routledge, 1993.
- [34] J. Hintikka, *Knowledge and Belief: an Introduction to the Logic of the Two Notions*: Cornell University Press, 1962.
- [35] D. Perlis, “Languages with Self-Reference II: Knowledge, Belief, and Modality,” *AI 34*, pp. pp 179-212, 1988.
- [36] G. J. Klir, *Architecture Of Systems Problem Solving*. New York: Plenum Press, 1985.
- [37] J. Mira and F. Sandoval, “From Natural to Artificial Neural Computation,” . International Workshop on Artificial Neural Networks, Spain: Springer, 1995.
- [38] D. S. Levine, *Introduction to Neural & Cognitive Modeling*. London: Lawrence Erlbaum Associates, Publishers, 1991.
- [39] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Edition ed. New Jersey: Prentice Hall, 1999.
- [40] R. Axelrod, *Structure of Decision: the Cognitive Maps of Political Elites*. Princeton, New Jersey: Princeton University Press, 1976.
- [41] A. Ayesh, “Neuro-Fuzzy Concepts Maps (NFCM),” . De Montfort University: Work in progress, 2002.
- [42] A. Ayesh, “Argumentative Agents-based Structure for Thinking-Learning,” presented at IASTED International Conference Artificial Intelligence and Applications (AIA 2001), Marbella, Spain, 2001.
- [43] A. Ayesh, “Thinking-Learning by Argument,” in *Intelligent Agent Technology: Research and Development*, N. Zhong, J. Liu, S. Ohsuga, and J. Bradshaw, Eds. New Jersey: World Scientific, 2001.
- [44] J. L. McClelland and D. E. Rumelhart, “Distributed Memory and Representation of General and Specific Information,” in *Human Memory: A Reader*, D. R. Shanks, Ed. London: Arnold, 1997, pp. 273-314.
- [45] D. R. Shanks, “Human Memory: A Reader,” . London: Arnold, 1997.
- [46] L. R. Squire, B. Knowlton, and G. Musen, “The Structure and Organization of Memory,” in *Human Memory: A Reader*, D. R. Shanks, Ed. London: Arnold, 1997, pp. 152-200.
- [47] A. Ayesh, “Memory Architecture for Argumentation-based Adaptive System,” presented at IASTED International Conference Applied Informatics (AI 2002), Innsbruck, Austria, 2002.
- [48] A. Ayesh, “Towards Memorizing by Adjectives,” presented at AAAI Fall Symposium on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems, 2001.
- [49] A. Ayesh, “Self Reference in AI,” in *Computer Science Dept*. Colchester: University of Essex, 1995.

Emotional Influences on Perception in Artificial Agents

Penny Baillie-de Byl,
Department of Mathematics and Computing, University of Southern Queensland.
penny.baillie@usq.edu.au

Keywords: agents, emotion, perception, decision-making, affective computing.

Received: October 17, 2002

This paper proposes a model of emotionally influenced perception in an affective agent. Via a multidimensional representation of emotion, a mechanism called the affective space is used to emotionally filter sensed stimuli in the agent's environment. This filtering process allows different emotional states in the agent to create dissimilar emotional reactions when the agent is exposed to the same stimuli. In humans, emotion cannot be a mechanism that enhances intelligence and then isolated from other psychological and physiological functions. As this paper suggests, emotion is an integral part of the human as a biological being and therefore they cannot be turned on or off on a whim. However, this luxury is afforded to the artificial agent. This approach builds on contemporary affective agent architectural concepts as it not only gives an agent the ability to use emotion to produce human-like intelligence, but it also investigates how emotion should affect the agent's other abilities, such as perception.

1 Introduction

The word *agent* is used within the AI (Artificial Intelligence) domain to refer to a number of different applications. The most popular use of the term pertains to an autonomous artificial being that has the ability to interact intelligently within a temporally dynamic environment. Just how the agent achieves its intelligent interaction has become a popular research topic. In the mid 1990s a small group of researchers became convinced that true human-like intelligence could not be modelled successfully in artificial beings without the inclusion of emotion-like mechanisms. Thus began the field of *Affective Computing*.

Humans sense their environment with five (possibly more) senses for detecting external stimuli and others for tracking their internal states (e.g. hunger). Artificial agents must also implement a number of mechanisms that track not only their external environment but also their internal states in order to interact intelligently with their environment and other agents. Agents therefore, by their very nature perceive, however, is it for the same ends as human perception?

Brunswik [1] wrote, "*Perception (in humans), then, emerges as that relatively primitive, partly autonomous, institutionalized, ratiomorphic subsystem of cognition which achieves prompt and richly detailed orientation habitually concerning the vitally relevant, mostly distal aspects of the environment on the basis of mutually vicarious, relatively restricted and stereotyped, insufficient evidence in uncertainty-geared interaction and compromise, seemingly following the highest probability for smallness of error at the expense of the highest frequency of precision.*"

These characteristics of human perception are the exact abilities that Affective Computing researchers are attempting to achieve in artificial agents in order to increase information-processing efficiencies. These facilities emulate the human thought processes of flexible and rational decision making, reasoning with limited memory, limited information and relatively slow processing speed, social interaction and creativity.

There are agents that can sense human emotional states [2], agents that can produce outward emotional behaviour [3, 4], and agents that are motivated by their emotions [5, 6]. The agents that internally represent emotional states for the purpose of goal setting and motivation [7] have mechanisms that perceive their environment and internal states for the purpose of calculating their emotions and producing appropriate behaviour.

Most of us have heard the phrase, "his mind is clouded by emotion", but to what extent is this type of emotion-perception influence occurring in artificial agents?

This paper presents the Emotionally Motivated Artificial Intelligence (EMAI) model whose sensory input is *emotionally filtered* before processing. It begins by examining the affect of emotion on perception in humans. Next, an overview of the agent architecture is given. This section includes details about the calculations used by an agent to determine emotional states. Following this, the factors of human perception are examined with respect to their emulation in the artificial agent. Next an example of the influence of emotional states on the agent's perception of environmental stimuli is discussed. This incorporates a brief overview of the

agent's emotion-based decision making technique. The paper concludes with a summary of the agent's evaluation and a thought for future research.

2 The Molecules of Emotion and Perception

Research has revealed that the neuropeptide receptors observed as responsible for emotional states and originally thought only to exist in the amygdala, hippocampus and hypothalamus have now been detected in high concentrations throughout the body. This has included the backside of the spinal cord, the nervous system's first synapse where bodily sensations and feelings are processed. Therefore, all sensory information passing between synapses via the emotion producing neuropeptides undergoes an *emotional filtering* process [8]. This operation assists in the brain's ability to deal with the deluge of sensory input that it receives.

The nervous system also carries signals, not only from the body to the brain, but also from the brain to the body. Emotional states or moods occur when emotion-carrying peptides are produced in the body's neurons. The presence of different emotional neuropeptides can create dissimilar reaction in an individual when exposed to the same stimuli.

Worthington, as reported by Malim [9], found that subjects consistently perceived dim spots of light containing consciously unreadable words with a higher emotional rating as dimmer than other words. In another experiment by Lazarus and McCleary [10], subjects were presented with a series of nonsense syllables. Electric shocks were administered to the subjects when particular syllables were shown and the anxiety level measured. Later, the subjects were exposed to the syllables at a rate faster than consciously perceivable. It was found that the syllables associated with the electric shocks raised the anxiety of the subjects.

Leuba and Lucas [11] also conducted an experiment on perception and emotion involving the description of six pictures by three people when in each of three different emotional states. Each emotion was induced by hypnosis and then the pictures were shown. Interpretations of the scenes in the pictures related to the emotions of the viewer. For example, a picture of several university students sitting on the grass listening to the radio was interpreted by the same person as relaxing when they were *happy*, irresponsible when they were in feeling *judgmental* and competitive when they were *anxious*.

These research examples confirm that human perception is filtered by emotions as Pert [8] suggests occurs at a molecular level. If perception is affected by emotion in humans, then surely artificial agents that attempt to model the emotional intelligence of humans must also represent the emotional filtering process of perception. However, as the emotional filtering process occurs in humans biologically, it is not an inherent process within a machine or piece of software. In affective agent architectures emotions have been

modelled at the cognitive level based on a number of appraisal models of emotion [12, 13]. Therefore, the agent presented in this paper, implements the cognitive nature of perception.

3 The EMAI Architecture

The Emotionally Motivated Artificial Intelligence (EMAI) architecture is a complex set of mechanisms that process emotional concepts for their use in affective decision-making and reasoning. A full elucidation of this architecture can be found in [14]. As the purpose of this paper is to examine the influence that emotion have on the perception of such an agent, the discussion will be limited to the parts the architecture that achieve this. A condensed overview of the parts of the EMAI architecture we are concerned with is shown in Figure 1.

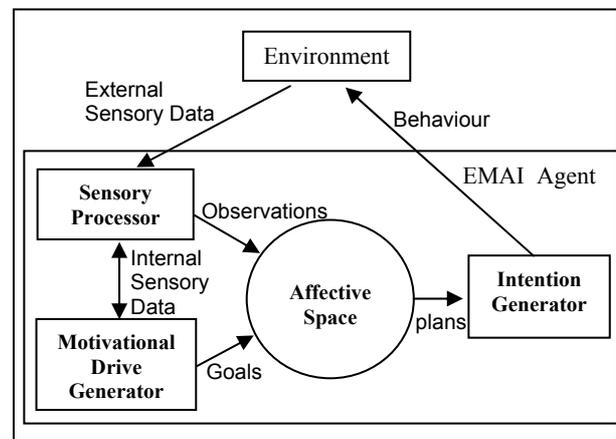


Figure 1 A Summary illustration of the EMAI architecture

There are two types of emotion mechanisms integrated in the EMAI architecture. The first mechanism emulates fast primary emotions [15] otherwise known as motivational drives. These drives can be classified according to their source, pleasure rating and strength. In an EMAI agent, these drives are used to initiate behaviour in the agent. They can include concepts such as *hunger*, *fatigue* or *arousal*. The strength of the drives is temporally dynamic and at particular threshold levels the agent will set goals, that when successfully achieved, will pacify the drives. For example, overtime the strength of the hunger drive will increase. At a certain point, the agent will become so hungry that it will set appropriate goals to ensure it obtains food. On the consuming of food the strength of the agent's hunger drive will decrease.

The agent's goals are generated by a motivational drive generator which consists of drive mechanisms and a set of *internal state registers* representing the primary emotions. Each register is represented by a single gauge and stores the value for a particular drive, for example *hunger*. The number of internal state registers implemented depends on the application for which the EMAI agent is being used.

The second type of emotion implemented in the EMAI architecture is secondary emotion. This category

of emotion refers to the resultant mental (and in turn physical) states generated by attempts to satisfy the goals. These emotions include feelings such as *happiness, anger, sorrow, guilt* and *boredom*. Secondary emotions are represented in the EMAI architecture as values in the affective space.

The affective space is a six-dimensional space defined by six appraisal dimensions. The affective space, based on the psychological model of Smith and Ellsworth [13], defines 15 emotions (*happiness, sadness, anger, boredom, challenge, hope, fear, interest, contempt, disgust, frustration, surprise, pride, shame and guilt*) with respect to the dimensions of *pleasantness, P, responsibility, R, effort, E, certainty, C, attention, A and control, O*.

The values of the pure emotion points for each of the 15 emotional states in the model are shown in Table 1.

Table 1 Mean Locations of Emotional Points (in the range -1.5 - +1.5) as Compiled in Smith and Ellsworth’s Study

Emotion	P	R	C	A	E	O
Happiness	-1.46	0.09	-0.46	0.15	-0.33	-0.21
Sadness	0.87	-0.36	0	-0.21	-0.14	1.51
Anger	0.85	-0.94	-0.29	0.12	0.53	-0.96
Boredom	0.34	-0.19	-0.35	-1.27	-1.19	0.12
Challenge	-0.37	0.44	-0.01	0.52	1.19	-0.2
Hope	-0.5	0.15	0.46	0.31	-0.18	0.35
Fear	0.44	-0.17	0.73	0.03	0.63	0.59
Interest	-1.05	-0.13	-0.07	0.7	-0.07	0.41
Contempt	0.89	-0.5	-0.12	0.08	-0.07	-0.63
Disgust	0.38	-0.5	-0.39	-0.96	0.06	-0.19
Frustration	0.88	-0.37	-0.08	0.6	0.48	0.22
Surprise	-1.35	-0.97	0.73	0.4	-0.66	0.15
Pride	-1.25	0.81	-0.32	0.02	-0.31	-0.46
Shame	0.73	1.13	0.21	-0.11	0.07	-0.07
Guilt	0.6	1.13	-0.15	-0.36	0	-0.29

Each appraisal dimension (explained in the next section) is used to produce a six coordinate point that defines an agent’s emotional state. Figure 2 illustrates the locations of the pure emotions with respect to the dimensions of pleasantness and control.

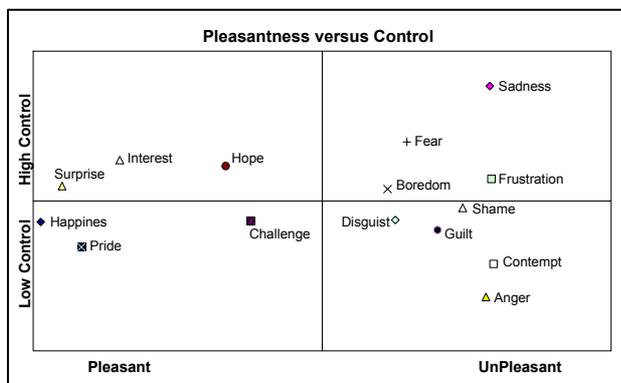


Figure 2 Empirical Location of Emotional States with Respect to the Pleasantness and Control Dimensions

In addition to representing the agent’s emotional state, the agent uses the affective space to associate emotions with all stimuli both internal to the agent (as internal sensory data) and within its environment (as external

sensory data). The stimuli are perceived by the agent as part of an event. An event is a behavioural episode executed by the agent. Stimuli can be any tangible element in the agent’s environment including the actions being performed, smells, objects, other agents, the time of day or even the weather.

The sensory processor of the agent is where high level observation takes place. This information is filtered through the affective space before it is used by the agent to generate outward behaviour (determined by the intention generator). It is at this point that the information has been *perceived*. Therefore all information perceived by the agent is influenced by the agent’s emotional state.

Before a stimulus or event can be perceived by the agent, the agent must calculate an associated emotion for each. The emotion associated with a stimulus is determined by examining each of the appraisal dimensions with respect to the agent’s last encounter with the stimulus.

3.1 Assessing the Appraisal Dimensions

The six appraisal dimensions are orthogonal, and no single emotion can be identified without taking into account each of these six appraisal dimensions. Each of these dimensions will now be reviewed.

3.1.1 Pleasantness

This dimension relates to an individual’s expression of liking or disliking towards a stimulus be it an event, object or another agent. An EMAI agent assesses and updates the pleasantness dimension as an assessment of the affect of the stimulus on the agent’s goals during an encounter. Pleasantness *P* is the average summation of the rating of pleasantness the agent has given to a stimulus each time the agent has come into contact with it:

$$P = \left(\sum_{i=1}^m p_{s_i} \right) / m \tag{1}$$

where *m* is the number of times the agent has come in contact with the stimulus, *s*, and *p_s* is the pleasantness rating of *s*. For example, assume the agent has driven the same car five times. The first time the car performs as expected and the agent is pleased with the car. In this first instance the agent may set the pleasantness rating to 8 on a scale from 1 to 10 where 1 is unpleasant and 10 is very pleasant. The second time the agent drives the car it breaks down. For this instance the agent rates the pleasantness of the car as 2. For the next three contacts with the car the agent rates the pleasantness as 2, 7 and 9. After these five contacts with the car, the agent, using Eq. (1) assesses the overall pleasantness rating of the car to be (8+2+2+7+9)/5 which equates to 5.6.

3.1.2 Responsibility

This dimension correlates with an individual’s sense of personal involvement and amount of blame or credit attributed towards the self when interacting with a stimulus. The two extremes of measure are self-

responsibility and others-responsibility. In the EMAI architecture, responsibility is a measure of the agent's relationship and attachment toward a stimulus. For example, the agent will calculate a high responsibility for an object where the agent considers it has ownership. More precisely, if the agent were in a team situation and within the team there were different ranks (for example, team leader, second in charge and third in charge), and the team was performing some task and the task failed then the team leader would feel most responsible, the second in charge less responsible and so on through the ranks.

Responsibility R is calculated using the function r that returns the level of responsibility related to a stimulus. The function works by determining the nature of the relationship between the agent and s . For example, if the agent were the owner of or in charge of s , $r(s)$ would return a high value. Responsibility is calculated as follows:

$$R = r(s) \quad (2)$$

3.1.3 Effort

The values along this dimension are gauged from an agent's exertion with respect to a stimulus that affects the agent either mental or physical. In the EMAI architecture, effort is a function of the depletion of resources used when interacting with an object or performing an event divided by the length of time spent interacting. For an EMAI agent, at the beginning of performing an event the agent's physical state is recorded. During the event, the agent's physical state may be affected by the event's stimuli. At the completion of an event, the change in the agent's physical state is used to calculate effort. For example, if the agent were to perform the task of digging a hole, during the execution of the event the agent's physical state may deteriorate because the agent may get *tired* and *hungry*. When the agent finishes digging the hole, the change in *tiredness* and *hunger* during the task performance will be directly related to the effort involved in the event.

Effort E is the averaged summation of the effort associated with a stimulus each time the agent has been in contact with it:

$$E = \left(\sum_{i=1}^m f_{s_i} \right) / m \quad (3)$$

where m is the number of times the agent has come in contact with the stimulus, s , and f is the amount of effort involved with s .

3.1.4 Attention

This dimension is the rating of an individual's regard for a stimulus with respect to the level of concentration exerted towards it during interaction. All EMAI agents are programmed with a maximum attention capacity, and each of these agents can perform one or more tasks, that utilise its maximum capacity. In EMAI, attention is measured as the total amount of an agent's attention that is utilised in performing one or more events concurrently.

Attention A is determined by averaging the summation of all the attention ratings the agent has assigned to a stimulus. Each time the agent is involved with a stimulus or event, the agent records how much concentration was exerted during the encounter and uses these values to calculate A :

$$A = \left(\sum_{i=1}^m a_{s_i} \right) / m \quad (4)$$

where m is the number of times the agent has come in contact with the stimulus, s , and a is the amount of attention required by the agent when involved with s .

3.1.5 Control

This dimension refers to an agent's authority to manipulate and direct a stimulus. It assesses the agent's ability to control the role of a stimulus during the satisfaction of the agent's goals. Each EMAI agent is initially programmed with its control value over other stimuli as 0. Over time, as an EMAI agent evolves, its control values toward stimuli changes (either increases, or decreases) according to the outcome of performed behaviours. Almost always there will be one or more stimulus involved in an event. So initially, the EMAI agent's control over this event is 0, and the control over each of the individual stimuli involved in this event is also 0. Based on the outcome of this event (that is, success or failure), the control of the agent towards each of the stimuli involved in the event will be either increased by 1 if the event is successful or reduced by 1 otherwise. The overall control over the event is then the average of the control over each individual stimulus in the event. For example, assume that an agent is driving a car on the Princes Highway from Sydney to Melbourne. Here, there are four stimulus involved in this event. They are: the car, the Princes Highway, the source city Sydney, and the destination city Melbourne. Initially, the control the agent has over these four stimuli is 0. Now, assuming that this event was successful, the agent will increment the control value of each of these stimuli by 1, and calculate the overall control value towards this event to be the average control over all the stimuli in the event. In this case, it will be 1 $((1+1+1+1)/4)$. Further, assume the agent now performs the same event successfully at another time. Now, its control over each of the stimuli will be increased by 1, and the overall control towards the event will be 2. But, if on the third trip, the agent drives the car along Pacific Highway from Sydney to Brisbane, the initial control over Sydney and car is 2, respectively, but the control over Pacific Highway and Brisbane is 0, respectively. So, the overall control over this event will be 1 $((2+2+0+0)/4)$. If this event was successful, the overall control will be 2 $((3+3+1+1)/4)$.

Control O is calculated by averaging the summation of the amount of control the agent has had over an stimulus or event during every encounter the agent has had with the stimulus or event:

$$O = \left(\sum_{i=1}^m o_{s_i} \right) / m \quad (5)$$

where m is the number of times the agent has come in contact with the stimulus s , and o is the amount of control the agent had over s .

3.1.6 Certainty

This dimension refers to an individual’s assessment of a stimulus as to the reliability that its affects or behaviours can be predicted. This is calculated in an EMAI agent by considering the degree of success or failure the agent has had in past encounters with the stimulus. For example, if the agent has carried out 50 attempts at driving from Sydney to Melbourne on the Princes Highway, and it succeeded in 30 of its attempts, and failed the other 20, then the certainty of success on the 51st attempt will be 0.6 and the certainty of failure will be 0.4.

Certainty C is calculated by determining the probability of success for the agent’s involvement with a stimulus. Given that S is a function that returns the number of times that s has been used by or involved with the agent for a successful event, C can be calculated as:

$$C = S(s) / m \tag{6}$$

where m is the number of times the agent has come in contact with the stimulus s .

3.2 Expressing An Agent's Emotional State

An agent's emotional state, Ω , can be expressed as,

$$\Omega = \{P, E, C, A, R, O\} \tag{7}$$

Given Ω , the emotional state of the agent can be deduced, for the purpose of expression in natural language, by determining the distance that Ω is from each of the 15 pure emotion points. To do this a simple linear distance function is applied¹. To determine a word that best describes the agent's emotion state of Ω the distance between Ω and each of the pure emotions ($\Omega_1 \dots \Omega_{15}$) is calculated using

$$\Delta_{\Omega} = \sqrt{(P - P_j)^2 + (E - E_j)^2 + (C - C_j)^2 + (A - A_j)^2 + (R - R_j)^2 + (O - O_j)^2} \tag{8}$$

where 15 values are calculated for $j = 1, \dots, 15$. The pure emotion closest to the agent's emotional state, expressed as a written word, Em , closest to Ω is then determined by using

$$Em = \text{emotion_name}(\min(\bigcup_{j=1}^{15} \Delta_{\Omega})) \tag{9}$$

where the function min returns the pure emotion point in closest proximity to Ω and the function $emotion_name$ converts the pure emotion point into a string.

¹ While more complex distance functions could be implemented and examined, for simplicity this will not be examined further in this investigation of the EMAI architecture.

For example, assume an agent with an emotional state point of $\Omega = [0.15, 0.87, 0.35, -0.3, 0.1, -0.5]$. To find the name of the emotion that best describes the item’s emotional state, the first step is to find the distance between this point and the 15 pure emotion points in the Affective Space using Eq. (2). The results are shown in Table 2.

Table 2 Distance Between Item’s Emotional State and Pure Emotions in Affective Space

Emotion	Distance (Δ_{Ω})
Happiness	0.0208
Sadness	0.0222
Anger	0.0218
Boredom	0.0215
Challenge	0.0159
Hope	0.0146
Fear	0.0170
Interest	0.0211
Contempt	0.0168
Disgust	0.0174
Frustration	0.0193
Surprise	0.0268
Pride	0.0164
Shame	0.0080
Guilt	0.0084

It can be seen from Table 2 that the agent's emotional state can best be described as *shame*.

3.3 Assigning Emotion to Stimuli

An EMAI agent primarily perceives a stimulus and associates an emotion with it based on how the agent assess the stimulus with respect to the six appraisal dimensions. The emotion associated with a stimulus, s , is expressed in the same manner as was used for the agent's emotional state in Eq. (7), thus

$$\Omega_s = \{P_s, E_s, C_s, A_s, R_s, O_s\} \tag{10}$$

As events occurring in an EMAI agent's environment rarely consist of just one stimulus, stimuli are rarely processed individually. Based on the outcome of processing an event, E , the agent will assign a weighting w , to the emotional state of each of the stimuli. As the weighting of a stimuli and resulting emotional state with respect to an event are dynamic, the time, t , at which the emotional state is being calculated must also be taken into consideration. Therefore, the emotional state resulting from an event E , written as $\Omega_{E,t}$ is calculated as

$$\Omega_{E,t} = \sum_{s=1}^n w_{s,t} \Omega_{s,t} \tag{11}$$

where n is the number of stimuli associated to event E , and

$$\sum_{s=1}^n w_s = 1 \quad \text{and} \quad 0 \leq w_s \leq 1$$

After the event, each of the stimuli involved in the event have their emotional associates updated with respect to the change in the emotional state of the agent evoked by the outcome of the event, $\Omega_{O,t+1}$. $\Omega_{O,t+1}$ represents the emotional state of the agent after an event has occurred where O stands for the outcome emotion and $t+1$ is the time at which the event ended. This value is not the same as the emotional state assigned to the event after it has been executed. This is calculated later in this section.

A change in the emotional state of the agent occurs when the values for each of the appraisal dimensions (P, E, C, A, R, O) are updated during and after an event. While each of six appraisal values of an individual stimuli involved in the event influence how these values are changed for the agent's emotional state, the final emotional state cannot be determined before the event occurs. The agent can only speculate. For example, an event the agent believes will make the agent *happy* may fail during execution, may take longer to complete than initially thought or may require extra effort. These factors would change the values of the appraisal dimensions independently of any influence over these values by the stimuli of an event or the event itself. The resulting emotional state in this example may be *sad* rather than the expected *happy*. Therefore, $\Omega_{O,t+1}$ cannot be calculated by combining the appraisal dimensions of the stimuli of an event, but can only be determined after the event has occurred. Only then can an analysis of the appraisal dimensions take place. This would include values from the appraisal dimensions of the stimuli in the event and also takes into consideration changes in the agent's physical and mental states. The new emotional state of the agent is used to update the values of the appraisal dimensions for each of the event stimuli. The agent attributes a change in its emotional state to be the result of the event and therefore, updates the emotional state of the event and its stimuli accordingly.

Having said this, $\Omega_{O,t+1}$ can be *predicted* by combining the appraisal of an event with the agent's current emotional state, thus

$$\Omega_{O,t+1} = \Omega + w_{E,t+1} (\Omega - \Omega_{E,t}) \quad (12)$$

where n is the number of stimuli associated with the event E , and

$$0 \leq w_E \leq 1$$

The change in the emotional state of an event is calculated using

$$\Delta_\Omega = \Omega_{O,t+1} - \Omega_{E,t} \quad (13)$$

After this has been calculated the emotional states for each stimuli in the event can be updated as

$$\Omega_{s,t+1} = \Omega_{s,t} + w_{s,t+1} \Delta_\Omega \quad (14)$$

Instead of the stimulus taking on the final emotional state of the event, the previous emotional state of the stimulus is taken into account along with the effect the stimulus had in the resulting emotional state for the event. If the event's resulting emotional state is the same

as its initial state and $w_{s,t} = w_{s,t+1}$ then the emotional state for the stimulus will not change.

4 Results

The way in which emotional state of an EMAI agent influences its perception and in turn its decision making process will now be examined.

4.1 Emotions Influencing Perception

In psychology, perception is viewed as being influenced by motivation, emotion, social and cultural factors [9]. Each of these factors has the following effect on an individual:

- **readiness:** a greater inclination to react to a stimulus
- **precedence:** ensuring priority stimulus are processed before others
- **selection:** the choice of one stimulus over another
- **interpretation:** the effect of a stimulus is predicted before it is experienced

An EMAI agent's perception is influenced by the factor of emotion and influences the agent's behaviour with respect to the four effects listed above.

An agent's inclination to react to a particular stimulus is influenced by agent's current emotional state, its internal states, and the emotion the agent associates with the stimulus. For example, if the agent is hungry it will be ready to react to a food stimulus.

The precedence that an agent gives to the processing of a stimulus is dependant on the agent's current emotional states and its internal state. The agent's internal state determines which of the agent goals have the higher priority. If there are a number of goals with the same priority, the agent determines which goal to attempt using emotion-based decision making. For example, if the agent would prefer to be happy, it will perform an action to satisfy the goal that would make it most happy. This also makes the agent prioritise the stimuli that are processed. Stimuli involved in the fulfilment of the agent's goals are given higher priority. The same processing of agent goals and priority also give the agent the ability to select between the stimuli that are processed.

Finally, the way in which stimuli are associated with emotions in the EMAI architecture, give agents the ability to interpret or predict the effect that a stimulus will have on the agent. It is this prediction process that allows the agent to select which goal and stimuli to process.

Whenever an EMAI agent encounters a stimulus, that stimulus is perceived with respect to the agent's current emotional state and the last emotion associated with the stimulus. For example, if in a previous encounter the agent associated a stimulus, s , with the emotion surprise, Ω_s , the next time it encounters the stimulus it will evaluate how the emotion surprise would affect the agent's current emotional state. Figure 2, shows

an agent's emotional state at two independent time intervals. Ω_1 is happiness and Ω_1 is anger. Using Eq. (14) with a weighting of 0.5, the resulting perceived associated emotion is shown in Figure 3 as $\Omega_{s,1}$ (surprise-happiness) when the agent is happy (Ω_1) and $\Omega_{s,2}$ (close to challenge) when the agent is angry (Ω_2).

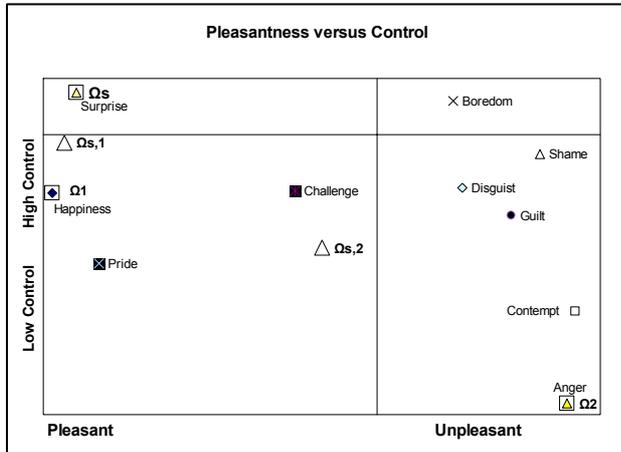


Figure 3. The agent's emotional state and its perception of a stimulus.

4.2 Emotion-based Reasoning

The way in which emotion affects the perception of stimuli in an EMAI agent also influences its decision-making process. This procedure is twofold. Firstly, the agent prioritises its behaviours by ordering them according to strength of the associated internal state register (representing a primary emotion). Secondly, the agent further orders its intended behaviours by calculating the resulting emotional effect that performing the behaviour would have on the agent's emotional state.

Given a number of behaviours that have the same priority, the agent will select a behavioural event that will most likely update the agent's emotional state to a more preferred emotional state. For example, if the agent had two events of equal urgency from which to select, the agent would further prioritise these events emotionally. The agent calculates the emotional point for each event and then interpolates how this event, when performed, will update the agent's emotional state. If the agent would prefer to have an emotional state closer to *happy* it would select the event that would, when combined with its current emotional state, make the agent *happy*. Assume the agent is experiencing hunger according to a high level on the internal register representing this state. The prioritised goal would be *eat*. The agent's environment may contain a number of stimuli that could help satisfy the eat goal (e.g. apple and chocolate). If eating chocolate would make the agent more happy than eating the apple, the agent would choose the chocolate.

As the agent's perception of the stimuli involved in a behavioural event changes with the agent's current emotional state, an event chosen that would change the agents mood to happy from angry will be different from

an event that would change the agents mood to happy from guilty.

Lets consider two events, E_1 and E_2 where

$$\Omega_{E_1} = [0.44, -0.5, -0.12, 0.08, -0.07, -0.63]$$

and

$$\Omega_{E_2} = [0.44, -0.17, 0.73, 0.03, 0.63, 0.7]$$

Ω_{E1} is best described as *contempt* and Ω_{E1} is best described as *fear* using Eq. (8).

If the agent were in a happy mood such that

$$\Omega = [-1.49, 0.09, -0.46, 0.15, -0.33, -0.21]$$

the predicted outcome of each event could be calculated using Eq. (12) (assuming a weighting of 0.5) thus

$$\begin{aligned} \Omega_{O,E_1} &= \Omega + w_{E_1,t} (\Omega - \Omega_{E_1,t}) \\ &= [-1.49, 0.09, -0.46, 0.15, -0.33, -0.21] + \\ &\quad 0.5 \times ([-1.49, 0.09, -0.46, 0.15, -0.33, -0.21] - \\ &\quad [0.44, -0.5, -0.12, 0.08, -0.07, -0.63]) \\ &= [-0.51, -0.205, -0.29, 0.115, -0.2, -0.42] \end{aligned}$$

and

$$\begin{aligned} \Omega_{O,E_2} &= \Omega + w_{E_2,t} (\Omega - \Omega_{E_2,t}) \\ &= [-1.49, 0.09, -0.46, 0.15, -0.33, -0.21] + \\ &\quad 0.5 \times ([-1.49, 0.09, -0.46, 0.15, -0.33, -0.21] - \\ &\quad [0.44, -0.17, 0.73, 0.03, 0.63, 0.7]) \\ &= [-0.51, -0.04, 0.135, 0.09, 0.15, 0.245] \end{aligned}$$

The agent would predict, using Eq. (9) that E_1 would make it feel *happy* and E_2 would make it feel *fear*. These calculations are shown graphically in Figure 4².

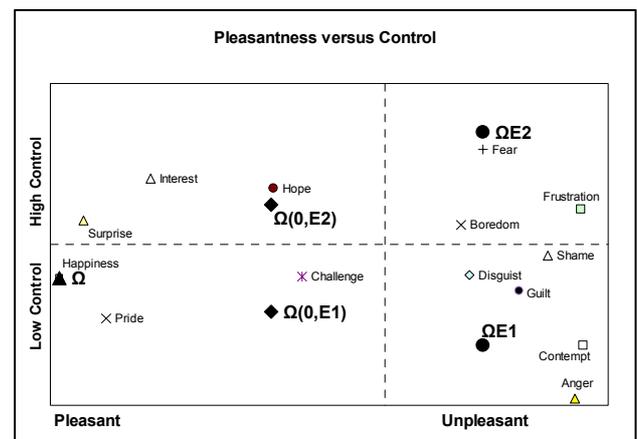


Figure 4 The predicted resulting emotions from events E_1 and E_2 on a happy agent.

If however, the agent was in a guilty mood, that is

² Although, in this 2D figure, E_2 appears to be closer to hope, linearly in six-dimensions it is not.

$$\Omega = [0.6, 1.31, 0.15, -0.36, 0, -0.29]$$

the outcomes from E_1 and E_2 would be perceived differently. Using Eq. (12) and in turn Eq. (9), $\Omega_{O,E1}$ and $\Omega_{O,E2}$ would be described as shame with the point for $\Omega_{O,E2}$ lying closer to pure shame than $\Omega_{O,E1}$ in the affective space. The emotions for E_1 , E_2 , the agent and the predicted emotions are shown in Figure 5.

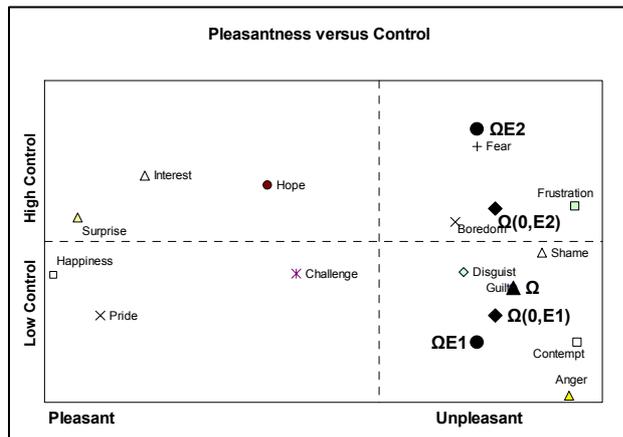


Figure 5 The predicted resulting emotions from events E_1 and E_2 on a guilty agent.

It can be seen from this example that the agent's current emotional state influences how stimuli and their associated events are perceived by the agent. Once a prediction has been made as to the outcome of a number of events the agent can select from these which event will become the agent's outward behaviour based on a preferred emotional state. For example, if the agent would prefer to be in a happy emotional state, it would select the event that was predicted to make the agent's outcome emotion closest to happiness in the affective space.

5 Summary

Emotions are a difficult concept to define let alone integrate into the domain of artificial intelligence. Emotions have been studied in fields such as philosophy, physiology, neurology and psychology. All have their own and often distinct ideas and models explaining how emotions are generated and affect behaviour. One distinct mechanism in all agent architectures is perception and if agents are to be designed that integrate emotion to enhance general intelligence, the influence that emotion has on the agent's senses should also be examined.

The EMAI architecture was developed to examine the multidimensional appraisal of emotion and the use of such a construct in artificial emotion processing. As emotion and perception are complexly intertwined in humans, it was concluded that such a relationship should exist in an affective agent. Perception in an EMAI agent does not occur at the sensing level, however, all incoming sensory data is emotionally filtered through the affective space. The current emotional state of an EMAI

agent affects how the agent perceives any incoming information.

The evaluation of the use of the EMAI architecture to model a computerised character, presented in [14], determined if the model was sufficiently capable of using its set of highly integrated mechanisms for generating motivation, goal setting, emotional intelligence and event prioritisation and scheduling. The results gave positive feedback about the EMAI architecture's ability to produce reasonable emotional states and associated behaviours. The data also confirmed the agent's ability to set and execute goals using motivational mechanisms related to the agent's physical and mental states.

What the future holds for the field of affective computing is unclear. As it is very much in its infancy, researchers need to continue to examine and assess the elementary concepts of emotion generation and emotion influence. No one theory stands out from the rest as being the ideal. The complexities of human emotions may be too extreme to include them holistically within an artificial intelligence at this time. Only those segments of emotional behaviour that are advantageous to the goals of an artificial being should be considered.

References

- [1] Brunswik, E. (1956) *Perception and the Representative Design of Psychological Experiments*. 2d ed., rev. & enl. Berkeley: Univ. of Calif. Press.
- [2] Picard, R. W. (2000) Toward Computers that Recognize and Respond to User Emotion?, *IBM Systems Journal*, Volume 39, No. 3 & 4, 2000.
- [3] El-Nasr, M. S. (1998) *Modeling Emotion Dynamics in Intelligent Agents*, M.Sc. Dissertation, American University in Cairo.
- [4] Reilly, W. S. N. (1996) *Believable Social and Emotional Agents*. Ph.D. Dissertation, Carnegie Mellon University.
- [5] Canamero, D. (1997) Modeling Motivations and Emotions as a Basis for Intelligent Behaviour, in *Proceedings of the First International Conference on Autonomous Agents, New York, 1997*, ACM Press, New York, pp.148-155.
- [6] Padgham, L. & Taylor, G. (1997) A System for Modeling Agents having Emotion and Personality, *Lecture Notes in Artificial Intelligence*, Springer-Verlag. vol.12, no. 9, pp. 59-71.
- [7] Baillie, P. Lukose, D & Toleman, M. (2002) 'Engineering Emotionally Intelligent Agents', in *Intelligent Agent Software Engineering*, eds. V. Plekhanova & S. Wermter, Idea Publishing Group, Hershey.
- [8] Pert, C. B. (1997) *Molecules of Emotion*, Simon and Schuster, New York.
- [9] Malim, T. (1994) *Cognitive processes. Attention, perception, memory, thinking and language*. London: MacMillan.

- [10] Lazarus, R. S., & McCleary, R. (1951) Autonomic discrimination without awareness: A study of subception. *Psychological Review*, 58, 113-122.
- [11] Leuba & Lucas (1945) The effects of attitudes on descriptions of pictures', *Journal of Experimental Psychology, American Psychological Association*, Washington, vol. 35, pp. 517-524.
- [12] Ortony, A., Clore, G.L. & Collins, A., (1988) *The Cognitive Structure of Emotions*. Cambridge University Press, Cambridge.
- [13] Smith, C. A. & Ellsworth, P.C. (1985) Attitudes and Social Cognition, in *Journal of Personality and Social Psychology*, American Psychologists Association, Washington, vol. 48, no. 4, pp. 813-838.
- [14] Baillie, P. (2002) The Synthesis of Emotions in Artificial Intelligences. Ph.D. Dissertation, University of Southern Queensland.
- [15] Koestler, A. (1967) *The Ghost in the Machine*, Penguin Books Ltd., London.

Enhancing the Performance of Neurofuzzy Predictors by Emotional Learning Algorithm

Caro Lucas

Control and Intelligent Processing Center of Excellence, Electrical and Computer Eng. Department,
University of Tehran, Tehran, Iran and School of Intelligent Systems,
Institute for studies in theoretical Physics and Mathematics, Tehran, Iran
lucas@ipm.ir

Ali Abbaspour, Ali Gholipour and Babak N. Araabi

Control and Intelligent Processing Center of Excellence, Electrical and Computer Eng. Department,
University of Tehran, Tehran, Iran
aabbaspr@ut.ac.ir, gholipoor@ut.ac.ir, araabi@ut.ac.ir

Mehrdad Fatourech

Electrical and Computer Eng. Department, University of British Columbia, BC, Canada
mehrdadf@ece.ubc.ca

Keywords: Emotional Learning, Prediction, Nonlinear Time Series, Neurofuzzy Model

Received: October 8, 2002

Neural networks and Neurofuzzy models have been successfully used in the prediction of nonlinear time series. Several learning methods have been introduced to train the Neurofuzzy predictors, such as ANFIS, ASMOD and FUREGA. Many of these methods, constructed over Takagi Sugeno fuzzy inference system, are characterized by high generalization. However, they differ in computational complexity. The emotional Learning, which is successfully used in bounded rational decision making, is introduced as an appropriate method to achieve particular goals in the prediction of real world data. For example, predicting the peaks of sunspot numbers (maximum of solar activity) is more important due to its major effects on earth and satellites. The emotional learning based fuzzy inference system (ELFIS) has the advantages of simplicity and low computational complexity in comparison with other multi-objective optimization methods. The efficiency of proposed predictor is shown in two examples of highly nonlinear time series. Appropriate emotional signal is composed for the prediction of solar activity and price of securities. It is observed that ELFIS performs better predictions in the important regions of solar maximum, and is also a fast and efficient algorithm to enhance the performance of ANFIS predictor in both examples.

1 Introduction

Predicting the future has been an interesting important problem in human mind. Alongside great achievements in this endeavor there remain many natural phenomena the successful predictions of which have so far eluded researchers. Some have been proven unpredictable due to the nature of their stochasticity. Others have been shown to be chaotic: with continuous and bounded frequency spectrum resembling white noise and sensitivity to initial conditions attested via positive Lyapunov exponents resulting in long term unpredictability of the time series. There are several developed methods to distinguish chaotic systems from the others, however model-free nonlinear predictors can be used in most cases without changes.

Comparing with the early days of using classical methods like polynomial approximators, neural networks have shown better performance, and even better are their successors: Neurofuzzy models [1], [2],

[3], [4]. Some remarkable algorithms have been proposed to train the neurofuzzy models [4], [5], [6], [7]. The pioneers, Takagi and Sugeno, presented an adaptive algorithm for their fuzzy inference system [5]. Some other methods, including adaptive B-spline modeling [6] and adaptive network-based fuzzy inference system [7], fulfill the principle of network parsimony which leads to high generalization of performance. Generalization is the most desired property of a predictor. The principle of parsimony says that the best models are those with the simplest acceptable structures and the smallest number of adjustable parameters.

Following the directions of biologically motivated intelligent computing, the emotional learning methodology has been introduced on the base of emotions which are argued, in contemporary psychology, to be better predictors of future achievements than IQ [8], [9]. The simulated approach is formulated on the base of an emotional signal which shows the emotions of a critic about the overall

performance of the system. The emotional signal can be produced by any combination of objectives or goals which improve the estimation or prediction. The loss function will be defined as a function of emotional signal and the training algorithm will be simply designed to minimize this loss function. Thus the need for elaborated definitions of loss function in multi objective problems, which results in high computational complexity, is simply handled by defining an appropriate emotional signal. The cost which should be paid is that the result will be just satisfying rather than optimizing. As a result, the model will be trained to provide the desired performance in a holistic manner. The emotional learning algorithm has three distinctive properties in comparison with other learning methodologies. For one thing, one can use very complicated definitions for emotional signal without increasing the computational complexity of algorithm or worrying about differentiability or renderability into recursive formulation problems. For another, the parameters can be adjusted in a simple intuitive way to obtain the best performance. Besides, the training is very fast and efficient. As can be seen these properties make the method preferable in real time applications like control and decision making, as have been presented in literature [10],[11],[12],[13],[14],[15],[16],[17],[18].

In this research the emotional learning algorithm has been used in the purposeful prediction of some real world data: the sunspot numbers and the price of securities. In predicting the sunspot number time series, the peak points, related to solar maximum regions, are more important to be predicted than the others due to their strong effects on space weather, communication systems and satellites. Additional achievements are fast training of model and low computational complexity. The main contribution of this paper is to provide accurate predictions using emotional learning for Takagi Sugeno neurofuzzy model. The results are compared with other methods of training neural and neurofuzzy models like RBF and ANFIS. The paper consists of six parts; the main aspects of Takagi-Sugeno fuzzy inference system along with associated learning methods are described in the second section. The third section deals with the various forms of utilizing emotional learning in the prediction problem. The results of applying the proposed prediction method to benchmark time series are reported and analyzed in sections four and five. Finally, the last section presents some remarkable properties of emotional learning and some concluding remarks.

2 NeuroFuzzy models

Two major approaches of trainable neurofuzzy models can be distinguished. The network based Takagi-Sugeno fuzzy inference system and the locally linear neurofuzzy model. The locally linear model is equivalent to Takagi-Sugeno fuzzy inference system

under certain conditions, and can be interpreted as an extension of normalized RBF network as well [2]. Therefore, the mathematical description of Takagi Sugeno neurofuzzy model which is the most general formulation will be described in this section.

The Takagi-Sugeno fuzzy inference system is constructed by fuzzy rules of the following type

$$\text{Rule}_i : \text{If } u_1 = A_{i1} \text{ And } \dots \text{ And } u_p = A_{ip} \quad (1)$$

$$\text{then } \hat{y} = f_i(u_1, u_2, \dots, u_p)$$

Where $i=1..M$ and M is the number of fuzzy rules. u_1, \dots, u_p are the inputs of network, each A_{ij} denotes the fuzzy set for input u_j in rule i and $f_i(\cdot)$ is a crisp function which is defined as a linear combination of inputs in most applications

$$\hat{y} = \omega_{i0} + \omega_{i1}u_1 + \omega_{i2}u_2 + \dots + \omega_{ip}u_p \quad (2)$$

Matrix form $\hat{y} = a^T(\underline{u}) \cdot W$

Thus the output of this model can be calculated by

$$\hat{y} = \frac{\sum_{i=1}^M f_i(\underline{u})\mu_i(\underline{u})}{\sum_{i=1}^M \mu_i(\underline{u})} ; \quad \mu_i(\underline{u}) = \prod_{j=1}^p \mu_{ij}(u_j) \quad (3)$$

Where $\mu_{ij}(u_j)$ is the membership function of j th input in the i th rule and $\mu_i(\underline{u})$ is the degree of validity of the i th rule. This system can be formulated in the basis function realization which clarifies the relation between Takagi-Sugeno fuzzy inference system and the normalized RBF network. The basis function will be

$$\phi_i(\underline{u}) = \frac{\mu_i(\underline{u})}{\sum_{j=1}^M \mu_j(\underline{u})} \quad (4)$$

as a result

$$\sum_{j=1}^M \phi_j(\underline{u}) = 1 \quad (5)$$

This neurofuzzy model has two sets of adjustable parameters; first the antecedent parameters, which belong to the input membership functions such as centers and deviations of Gaussians; second the rule consequent parameters such as the linear weights of output in equation (2). It is more common to optimize only the rule consequent parameters. This can be simply done by linear techniques like least squares [2]. A linguistic interpretation to determine the antecedent parameters is usually adequate. However, one can opt to use a more powerful nonlinear method to optimize all parameters together. Gradient based learning algorithms can be used in the optimization of consequent linear parameters. Supervised learning is aimed to minimize the following loss function (mean square error of estimation):

$$J = \frac{1}{N} \sum_{i=1}^N (y(i) - \hat{y}(i))^2 \quad (6)$$

where N is the number of data samples.

According to the matrix form of (2) this loss function can be expanded in the quadratic form

$$J = W^T R W - 2W^T P + Y^T Y / N \quad (7)$$

Where $R = (1/N)A^T A$ is the autocorrelation matrix, A is the $N \times p$ solution matrix whose i th row is $a(\underline{u}(i))$ and $P = (1/N)A^T y$ is the p dimensional cross correlation vector. From

$$\frac{\partial J}{\partial W} = 2RW - 2P = 0 \quad (8)$$

the following linear equations are obtained to minimize J :

$$RW = P \quad (9)$$

and W is simply defined by pseudo inverse calculation. One of the simplest local nonlinear optimization techniques is the steepest descent. In this method the direction of changes in parameters will be opposite to the gradient of cost function

$$\Delta W(i) = -\frac{\partial J}{\partial W(i)} = 2P - 2RW(i) \quad (10)$$

and

$$W(i+1) = W(i) + \eta \cdot \Delta W(i) \quad (11)$$

where η is the learning rate.

Other nonlinear local optimization techniques can be used for this purpose, e.g. the conjugate gradient or Levenberg-Marquardt which are faster than steepest descent. All these methods have the possibility of getting stuck at local minima. Some of the advanced learning algorithms, that have been proposed for the optimization of parameters in Takagi-Sugeno fuzzy inference system, include ASMOD (Adaptive B-Spline modeling of observation data) [6], ANFIS (Adaptive network based fuzzy inference system) [7] and FUREGA (fuzzy rule extraction by genetic algorithm) [2]. ANFIS is one of the most popular algorithms that has been used for different purposes, such as system identification, control, prediction and signal processing. It is a hybrid learning method based on gradient descent and least square estimation. ASMOD is an additive constructive algorithm based on k -d tree partitioning. It reduces the problems of derivative computation, because of the favorable properties of B-spline basis functions. Although ASMOD has a complicated procedure, it has advantages like high generalization and accurate estimation.

One of the most important problems in learning is the prevention of over fitting. It can be done by observing the error index of test data at learning iterations. The learning algorithm will be terminated, when the error index of test data starts to increase, in an average sense. Prevention of over fitting is the most common way of providing high generalization.

3 Emotional Learning

Satisficing approaches to decision making has, in recent years, been widely adopted for dealing with complex engineering problems [18]. New learning

algorithms like reinforcement learning, Q-learning, and the method of temporal differences [19], [20], [21], [22], [23] are characterized by their fast computation and in some cases lower error in comparison with classical learning methods. They can be interpreted as approximations to dynamic programming, which although furnishes a well known computational algorithm, via recursive solution of the Bellman-Jacobean-Hamilton equation and perhaps the best example of fully rational approach to decision making, is notorious for its computational complexity, sometimes referred to as the “curse of dimensionality” [24], [25]. Fast training is a notable consideration in control applications. Prediction applications also belong to the class of decision making problems where two desired characteristics are accuracy and low computational complexity.

The Emotional learning method is a psychologically motivated algorithm which is developed to reduce the complexity of computations in prediction problems with particular goals. In this method the reinforcement signal is replaced by an emotional cue, which can be interpreted as a cognitive assessment of the present state in light of goals and intentions. The main reason of using emotion in a prediction problem is to lower the prediction error in some regions or according to some features. For example predicting the sunspot number is more important in the peak points of the eleven-year cycle of solar activity, or accurate prediction of the peaks and valleys in the price of securities may be desired. This method is based on an emotional signal which shows the emotions of a critic about the overall performance of prediction. The emotional signal can be produced by any combination of objectives or goals which improve estimation or prediction. The loss function will be defined just as a function of emotional signal and the training algorithm will be simply designed to decrease this loss function. So the predictor will be trained to provide the desired performance in a holistic manner. If the critic emphasizes on some regions or some properties, this can be observed in his emotions and simply affects the characteristics of predictor. Thus the definition of emotional signal is absolutely problem dependent. It can be a function of error, rate of error change and many other features. Finding an appropriate formulation for emotion is not usually possible; in contrast a linguistic fuzzy definition of it is absolutely intuitive and plausible.

A loss function is defined on the base of emotional signal. A simple form is

$$J = \frac{1}{2} K \sum_{i=1}^N es(i)^2 \quad (12)$$

where $es(i)$ is the of emotional signal to the i th sample of training data, and K is a weighting matrix, which can be simply replaced by unity.

Learning is adjusting the weights of model by means of a nonlinear optimization method, e.g. the steepest descent or conjugate gradient. With steepest

descent, the weights are adjusted by the following variations:

$$\Delta\omega = -\eta \frac{\partial J}{\partial \omega} \tag{13}$$

where η is the learning rate of the corresponding neurofuzzy controller and the right hand side can be calculated by chain rule:

$$\frac{\partial J}{\partial \omega} = \frac{\partial J}{\partial es} \cdot \frac{\partial es}{\partial y} \cdot \frac{\partial y}{\partial \omega} \tag{14}$$

According to (12): $\frac{\partial J}{\partial es} = K \cdot es$

and $\frac{\partial y}{\partial \omega}$ is accessible from (3) where $f_i(\cdot)$ is a linear function of weights.

Calculating the remaining part, $\frac{\partial es}{\partial y}$, is not

straightforward in most cases. This is the price to be paid for the freedom to choose any desired emotional cue as well as not having to impose presuppose any predefined model. However, it can be approximated via simplifying assumptions. If, for example error is defined by

$$e = y_r - y \tag{15}$$

where y_r is the output to be estimated, then

$$\frac{\partial es}{\partial y} = -\frac{\partial e}{\partial e} \tag{16}$$

can be replaced by its sign (-1) in (14). The algorithm is after all, supposed to be satisficing rather than optimizing.

Finally the weights will be updated by the following formula:

$$\Delta\omega = -K \cdot \eta \cdot es \cdot \frac{\partial y}{\partial \omega} = -K \cdot \eta \cdot es \cdot \frac{\sum_{i=1}^M u_i \mu_i(\underline{u})}{\sum_{i=1}^M \mu_i(\underline{u})} \tag{17}$$

The definition of emotional signal and the gradient based optimization of the emotional learning algorithm in neurofuzzy predictors are clarified among two examples in the next sections.

4 Predicting the Sunspot numbers

Solar activity has major effects not only on satellites and space missions but also on communications and weather on earth. This activity level changes with a period of eleven years, called solar cycle. The solar cycle consists of an active part, the solar maximum, and a quiet part, solar minimum. During the solar maximum there are many sunspots, solar flares and coronal mass ejections. A useful measure of solar activity is the observed sunspot numbers. Sunspots are dark spots on the surface of the sun which last for several days. The SESC sunspot number is computed according to the Wolf's sunspot number $R=k(10g+s)$, where g is the number of sunspot groups, s is the total number of spots in all the

groups and k is a variable scaling factor that indicates the conditions of observation.

A variety of techniques have been used in the prediction of solar activity, most of which are based on the sunspot number time series. The sunspot number, which has been saved since 1700, shows low dimensional chaotic behavior and its prediction has been a challenging problem for researchers. However, good results are obtained by methods proposed in several articles [26], [27], [28], [29], [30]. In this research, both the monthly and the yearly averaged sunspot numbers are used to be predicted. Figure 1 shows the history of solar cycles on the base of yearly sunspot numbers. The error index in predicting sunspot numbers, similar to most of the previous studies, is the normalized mean square error (NMSE):

$$NMSE = \frac{\left(\sum_{i=1}^n (y - \hat{y})^2 \right)}{\left(\sum_{i=1}^n (y - \bar{y})^2 \right)} \tag{19}$$

In which y, \hat{y} and \bar{y} are observed data, predicted data and the average of observed data respectively.

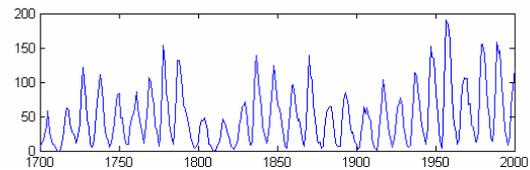


Figure 1: The yearly averaged sunspot number

As the first observation, the emotional learning algorithm has been used to enhance the performance of a neurofuzzy predictor, initially trained by ANFIS. The emotional signal is computed by a linguistic fuzzy inference system with error and rate of error change as inputs. Five and three Gaussian membership functions, negative large, negative, zero, positive and positive large, are used for the inputs (error and rate of error change, respectively) and the emotional signal is calculated by a center of average defuzzifier from the rule base depicted by the surface in figure 2.

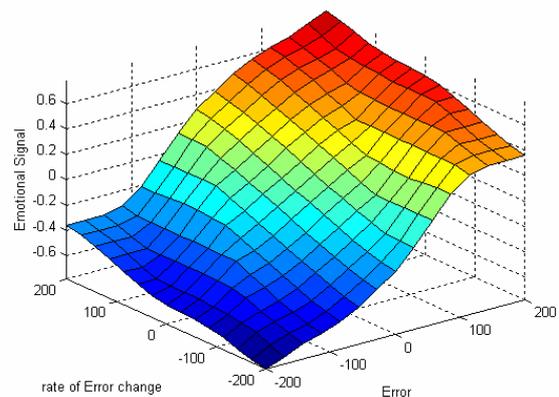


Figure 2: The surface generated by linguistic fuzzy rules of the emotional critic

There are seven Gaussian membership functions for the emotional signal as the output of fuzzy critic. The simulated fuzzy definition of the critic is motivated from our knowledge of emotions in human, and can be extended by inserting more inputs to the system. Figure 3 presents the targeted and predicted outputs of the test set (from 1920 to 2000). The lower diagram shows the results of best fitted data by ANFIS. The training is done with optimal number of fuzzy rules and epochs (74 epochs) and has been continued until the error of validation set had been started to increase. The other diagram shows the targeted and predicted values after using emotional learning. The emotional algorithm is used in one pass of the training data to fine tune the weights of neurofuzzy model which has been initially adjusted by ANFIS. The error index, NMSE, has been decreased from 0.1429 to 0.0853 after using emotional learning. The improvement of prediction accuracy, especially among the solar maximum regions, is noticeable. It's interesting that training ANFIS to the optimum performance takes approximately ten times more computation effort than the emotional learning to improve the prediction. Thus combining ANFIS with the emotional learning is a fast efficient method to improve the quality of predictions, at least in this example.

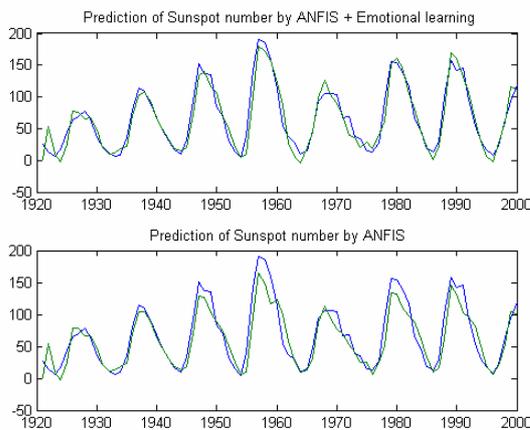


Figure 3: Enhancement in the prediction of sunspot numbers by emotional learning, applied to ANFIS: Targeted and predicted values; lower: by ANFIS, upper: by ANFIS + Emotional Learning

The next results are reported as a comparison of the quality of predicting the monthly sunspot numbers by the Emotional Learning based Fuzzy Inference System (ELFIS) with some other learning methods, the orthogonal least squares learning for the RBF network and Adaptive Network based Fuzzy Inference System (ANFIS). All methods are used in their optimal performance. Over fitting is prevented by observing the mean square error of several validation sets during training. ELFIS is constructed over Takagi Sugeno fuzzy inference system. The emotional signal is computed by a fuzzy critic whose linguistic rules are defined by means of error, rate of error change and the last targeted output. By defining appropriate

membership functions for each of the inputs and 45 linguistic fuzzy rules, the desired behavior of emotional critic is provided to show exaggerated emotions in the solar maximum regions. Figure 4 shows the surface generated by the fuzzy rules among the two dimensional space of the more important inputs (prediction error and last observed value of sunspot number). The emotional signal is used as the input to the learning formula (17) where the weights of neurofuzzy model (2) are adjusted. Just three Sugeno type fuzzy rules, like (1), are used in ELFIS to comply with the principle of parsimony. As a result, the matrix of adjustable weights has 9 elements (three weights for the three inputs of each rule). The specifications of methods, NMSE of predictions and computation times (on a 533 MHz Celeron processor) are presented in Table 1. It is observed that learning in ELFIS is at least four times faster than the others and is more accurate than ANFIS. Note that using a functional description of emotional signal rather than the fuzzy description generates a faster algorithm, but finding such a suitable function is not easy.

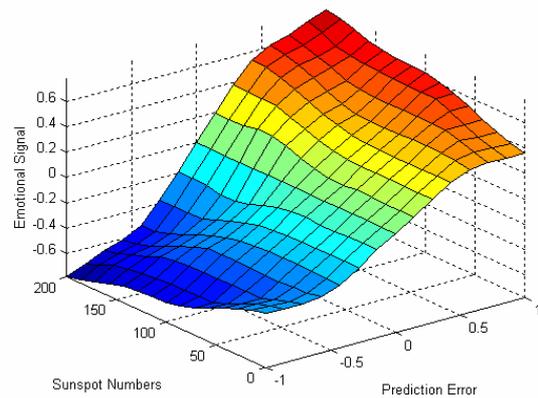


Figure 4: The surface generated by linguistic fuzzy rules of the emotional critic in ELFIS; in the prediction of monthly sunspot numbers

Table 1: Comparison of predictions by selected neural and neurofuzzy models

	Specifications	Computation Time	NMSE
ANFIS	8 rules and 165 epochs	89.5790 sec.	0.1702
RBF	7 neurons in hidden layer	84.7820 sec.	0.1314
ELFIS	3 Sugeno type fuzzy rules	22.3320 sec.	0.1386

Figures 5 to 7 show the predictions by RBF network, ANFIS and ELFIS respectively. These diagrams are a part of test set, especially the cycle 19 which has an above average peak in 1957. It's observable that ELFIS generates the most accurate predictions in the maximum region; however, NMSE of RBF is the least, indicating that RBF generates more accurate predictions through the total test set. By modifying the validation sets affecting the stop time of

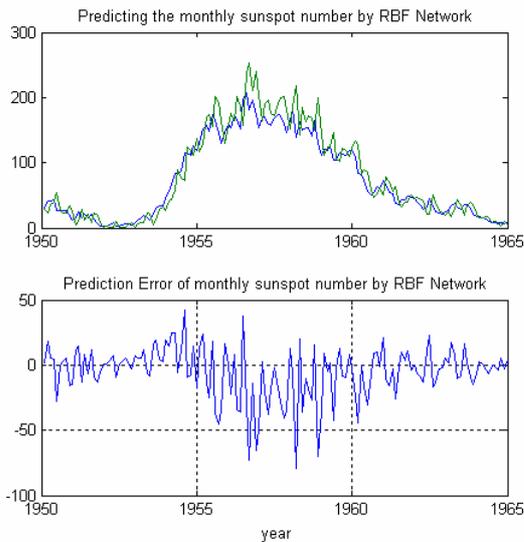


Figure 5: Predicting the sunspot numbers by RBF

learning procedure, even better NMSEs can be obtained in RBF, but this results in higher prediction errors especially in 1957.

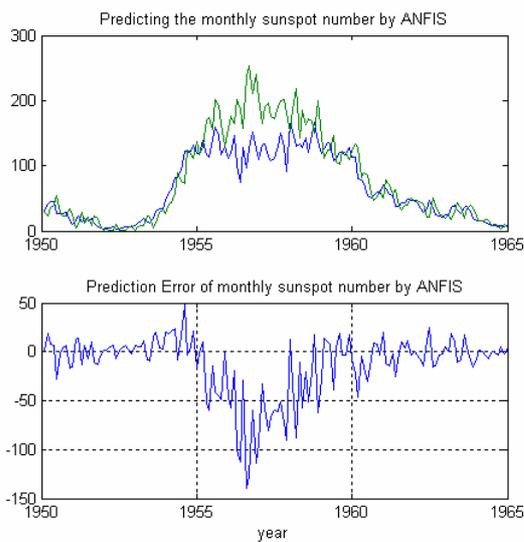


Figure 6: Predicting the monthly sunspot numbers by ANFIS

5 Predicting the Security Price

The second example is the prediction of securities such as stocks, treasury bonds and government bonds; etc. If there is a predictor that predicts the future exactly; then the best investment is on the maximum rate of return. For this reason, the performance of prediction is significant. Some researchers have used neural networks e.g. MLP and RBF for the prediction of securities. In this research, the emotional learning algorithm is applied to the network initially trained by ANFIS to predict the stock price of General Electric (GE) in S&P index 500. For this case one can use

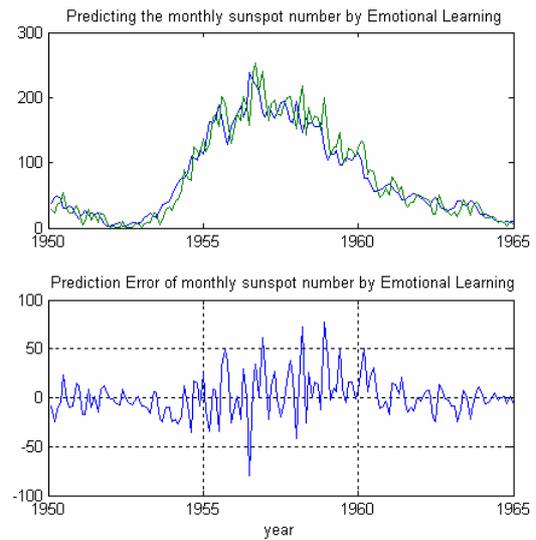


Figure 7: Predicting the monthly sunspot numbers by ELFIS

various definitions of emotional signal, as a function of prediction error and differential of error, or even any significant event like crossing of spot price with some well monitored moving average. Here the emotional signal is taken as the output of a linguistic fuzzy inference system with the error and the rate of error change as inputs. Five and three Gaussian membership functions are used for the inputs respectively. Figure 8 shows the surface generated by the fuzzy rules of emotional critic.

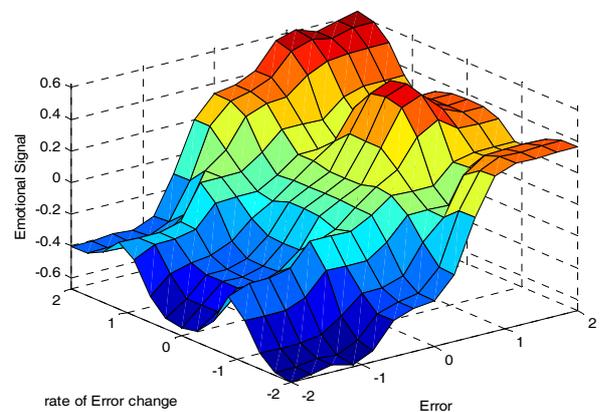


Figure 8: The output surface of a linguistic fuzzy inference system for producing emotional signal.

In this research, the daily closed price for the stock is considered. The model parameters, number of regressors and number of neurons are optimized to prevent over fitting. The stock price of 800 days and the price of 400 following days are used for train data and test data respectively. The result of predicting the stock price by ELFIS is presented in figure 9.

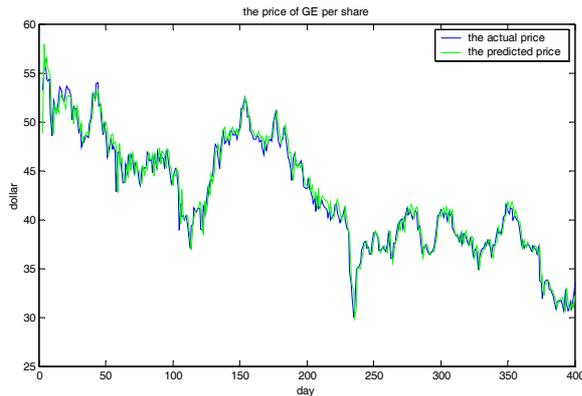


Figure 9: Predicting the security price using emotional learning plus ANFIS

Table 2 presents a comparison of the quality of predicting the daily closed stock price of General Electric (GE) by ELFIS with some other networks such as ANFIS, RBF, and MLP. These results are obtained by considering the over fitting and the optimal neurons in the hidden layer (on a 1.8 GHz Celeron processor). As this practical example shows, the emotional learning algorithm provides more accurate predictions with lower computational complexity.

Table 2: A comparison of various neural and neurofuzzy models in the stock price prediction

	Specifications	Computation Time	NMSE
MLP	37 neurons in hidden layer	6.5600 sec.	0.0347
ANFIS	12 rules and 257 epochs	13.8390 sec.	0.0370
RBF	31 neurons in hidden layer	7.2000 sec.	0.0395
ELFIS	12 Sugeno type fuzzy rules	1.8320 sec.	0.0358

6 Conclusion

Training a system to make decision in the presence of uncertainties is a difficult problem especially when computational resources are limited. Supervised training can not be used because the desired values for the decision variables are unknown. However, the desirability of past decisions can usually be assessed after the outcomes of their implementations are observed. Therefore, unsupervised training methods that do not utilize those assessments can not take full advantage of the available knowledge. Several approximate methods like back propagation through plant, and identification of the plant or (pseudo) inverse plant model have been successfully used in the past couple of decades [31], [32], [33], [34]. Behavioral and emotional approaches to control and decision making can also be classified in this category [35]. Besides providing biological

plausibility, they have the extra advantage of not being confined to cheap control problems like set point tracking [10]. The emotional approach is a step higher in the cognitive ladder and can be more useful in goal-aware or context-aware applications (e.g. dealing with multiple objectives in decision problems even when the objectives are fuzzy or can not be differentiated or directly evaluated with simple mathematical expressions).

The main contribution of this paper is the application of those ideas to prediction domain. Although prediction is easier to deal with because we do not have the further complexity of unknown plant, and so the proposed learning methods should also be compared to error minimization methodologies, model free prediction has also become of great importance in the past few decades, and there have been many past efforts to train neuro- and/or fuzzy predictors with alternative loss functions. In this paper, we have used the emotional learning interpolation to two very important benchmark problems. The motivation is not confined to achieving computational efficiency or improving the total prediction accuracy. In both problems, achieving more accurate results in desired regions or according to some important features is an important goal towards which some increase in error indices alongside the total test set can be tolerated. Specifically, one wishes to improve the prediction quality of solar activity (the sunspot number time series) in solar maximum regions (the peak points of sunspot number) at the expense of the prediction accuracy in less interesting regions. In the case of stock market prediction too, the quality of predictions in trend reversal regions (peaks and valleys) are of greater importance for supporting investment decisions.

The achievements reported in this paper are twofold. On the one hand, excellent prediction quality has been achieved for the two different benchmark problems with considerable reduction in computational complexity. On the other hand, a psychologically motivated framework for considering alternative or even multiple goals in decision making (in this case prediction) has been proposed, which is easy to apply even when the goal can not be expressed via well known mathematical expression, or is not differentiable. The goals are satisfied by tuning the predictor so that an emotional signal indicating how the present state is assessed to be non-conductive to the goals, is continually minimized (i.e. we shift gradually to states assessed as more satisfactory with respect to the goals). The proposed emotional learning based fuzzy inference system (ELFIS) has been used in the prediction of solar activity (the sunspot number time series) where the emotional signal is determined with emphasis on the solar maximum regions (the peak points of sunspot number) and it has shown better results in comparison with RBF network and ANFIS. In the prediction of security price, the emotional learning algorithm is defined by emotions of a fuzzy critic and results in good predictions. In fact

the use of a combination of error and rate of error change leads to late overtraining of the model. The definition of emotional signal is an important problem in emotional learning algorithm and provides higher degrees of freedom. In the prediction of security price, better performance can be obtained through the use of variables in addition to the lagged values of the process to be predicted (e.g. fundamentalist as well as chartist data).

References:

- [1] Brown M., Harris C.J. (1994), *Neuro fuzzy adaptive modeling and control*, Prentice Hall.
- [2] Nelles O. (2001), *Nonlinear system identification*, Springer Verlag, Berlin.
- [3] Bossley K.M. (1997) *Neurofuzzy Modeling Approaches in System Identification*, PhD thesis, University of Southampton, Southampton, UK
- [4] Leung H., Lo T., Wang S. (2001) Prediction of noisy chaotic time series using an optimal radial basis function neural network, *IEEE Tran. On Neural Networks*, 12(5), pp. 1163-1172.
- [5] Takagi T., Sugeno M. (1985), Fuzzy identification of systems and its applications to modeling and control, *IEEE Tran. On systems, Man and Cybernetics*, vol. 15, pp. 116-132.
- [6]. Kavli T. (1993) ASMODO: An algorithm for adaptive spline modeling of observation data, *Int. J. of Control*, 58(4), pp. 947-967.
- [7] Jang J.R. (1993) ANFIS: Adaptive network based fuzzy inference system, *IEEE Tran. On systems, Man and Cybernetics*, 23(3), pp. 665-685.
- [8] Goleman D. (1995) *Emotional Intelligence*, New York: Bantam Books.
- [9] Picard R.W., Vyzas E., Healey J. (2001) Toward machine emotional intelligence: Analysis of affective physiological state, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23 (10), pp. 1175-1191.
- [10] Lucas C., Shahmirzadi D., Sheikholeslami N. (2003) Introducing BELBIC: Brain Emotional Learning Based Intelligent Controller, *Accepted for publication in the International Journal of Intelligent Automation and Soft Computing (Autosoft)*.
- [11] Jazbi A., Lucas C. (1999) Intelligent control with emotional learning, *7th Iranian Conference on Electrical Engineering, ICEE'99*, Tehran, Iran, pp. 207-212.
- [12] Lucas C., Jazbi S.A., Fatourehchi M., Farshad M. (2000) Cognitive Action Selection with Neurocontrollers, *Third Irano-Armenian Workshop on Neural Networks*, Yerevan, Armenia.
- [13] Fatourehchi M., Lucas C., Khaki Sedigh A. (2001) An Agent-based Approach to Multivariable Control, *Proc. of IASTED International Conference on Artificial Intelligence and Applications*, Marbella, Spain, pp. 376-381.
- [14] Fatourehchi M., Lucas C., Khaki Sedigh A. (2001) Reducing Control Effort by means of Emotional Learning, *Proc. of 9th Iranian Conf. on Electrical Engineering ICEE'01*, Tehran, Iran, pp. 41-1 to 41-8.
- [15] Fatourehchi M., Lucas C., Khaki Sedigh A. (2001) Reduction of Maximum Overshoot by means of Emotional Learning, *Proceedings of 6th Annual CSI Computer Conference*, Isfahan, Iran, pp. 460-467.
- [16] Perlovsky L.I. (1999), Emotions, Learning and control, *proc. of IEEE Int. symp. On Intelligent control/Intelligent systems and semiotics*, Cambridge MA, pp. 132-137.
- [17] Ventura R., Pinto Ferreira C. (1999) Emotion based control systems, *proc. of IEEE Int. symp. On Intelligent control/Intelligent systems and semiotics*, Cambridge MA, pp. 64-66.
- [18] Inoue K., Kawabata K., Kobayashi H. (1996) On a Decision Making System with Emotion, *proc. 5th IEEE International Workshop on Robot and Human Communication*, pp. 461-465.
- [19] Sutton R.S., Barto A.G. (1998) *Introduction to reinforcement learning*, MIT press, Cambridge.
- [20] Barto A., Sutton R., Watkins C. (1990) Learning and sequential decision making, in *Learning and Computational Neuroscience*, MIT press, Cambridge.
- [21] Watkins C. (1989) *Learning from delayed rewards*, PhD thesis, University of Cambridge, UK.
- [22] Watkins C., Dayan P. (1992) Q-Learning, *Machine Learning*, 8, pp. 279-292.
- [23] Sutton R.S. (1989) Learning to predict by the method of temporal differences, *Machine Learning*, 3, pp. 9-44.
- [24] Ungar L.H. (2002) Reinforcement learning from limited observations, *Workshop on Learning and Approximate Dynamic Programming*, Playacar, Mexico.
- [25] Brown M., Bossley K.M., Mills D.J., Harris C.J. (1995) High dimensional neurofuzzy systems: Overcoming the curse of dimensionality, *proc. of IEEE International Conference on Fuzzy Systems*, New Orleans, USA, pp. 2139-2146.
- [26] Izeman A. J. (1985) Wolf J.R. and the Zurich sunspot relative numbers, *The Mathematical Intelligence*, vol.7, no.1, pp. 27-33.
- [27] Tong H., Lim K. (1980) Threshold Autoregressive limit cycles and cyclical data, *J. Roy. Statistics. Soc. B*, no.42, pp. 245-292.
- [28] Tong H. (1996) *Nonlinear time series: A dynamical system approach*, Oxford press, UK.
- [29] Weigend A., Huberman B., Rumelhart D.E. (1990) Predicting the future: a connectionist approach, *Int. J. Of Neural systems*, vol. 1, pp. 193-209.
- [30] Weigend A., Huberman B., Rumelhart D.E., (1992) Predicting sunspots and exchange rates with connectionist networks, in *Nonlinear Modeling and Forecasting*, Casdagli, Eubank: Editors, Addison-Wesley, pp. 395-432.
- [31] Werbos P.J. (1974) *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, PhD thesis, Harvard University, USA.
- [32] Rumelhart D.E., Hinton G.E., Williams R.J. (1986) Learning internal representations by error propagation, in D.E. Rumelhart and J.L. McClelland,

editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, chapter 8, MIT Press, Cambridge.

[33] Narendra K.S., Parthasarathy K. (1990) Identification and Control of dynamical systems using neural networks, *IEEE Tran. On Neural Networks*, 1(1), pp. 4-27.

[34] Xu X., He H.G., Hu D. (2002) Efficient Reinforcement Learning Using Recursive Least-Squares Methods, *Journal of Artificial Intelligence Research*, 16, pp. 259-292.

[35] Bay J.S. (1997) Behavior Learning in Large Homogeneous Populations of Robots, *IASTED International Conference on Artificial Intelligence and Soft Computing*, pp. 137-140.

Emotion-Based Decision and Learning Using Associative Memory and Statistical Estimation

Bruno D. Damas and Luis M. Custódio

Institute for Systems and Robotics

Instituto Superior Técnico, Av. Rovisco Pais, 1, 1049-001, Lisboa, Portugal

bdamas@isr.ist.utl.pt, lmmc@isr.ist.utl.pt

Keywords: Agents, Emotions, Decision, Learning

Received: October 31, 2002

An emotional architecture for artificial agents, inspired in Damásio's concept of somatic markers, is described in this paper. An associative memory, capable of providing estimates of action consequences for given situations, is also developed. The role emotions play on behaviour triggering is also discussed. This architecture has been successfully applied to robotic soccer, leading to an effective emotion-based machine learning.

1 Introduction

There is an apparent simplicity in most of the human decision processes: deciding which restaurant should we dinner at or choosing a movie to watch does not usually take us to an explicit and exhaustive enumeration and consideration of all the information, conditions and constraints considered relevant to the decision making process. Yet, despite this simplicity, most of the times our decisions lead us to advantageous situations, or, at least, prevent potentially undesirable ones.

António R. Damásio has pointed out the role of emotions on those human decision processes (1). According to Damásio, emotions have a key importance in the whole human rationality, as they effectively provide a selection mechanism that distinguish, in a situation where a decision is needed, the best alternatives, according to the individual past experience. This emotion-based decision mechanism is based on the concept of *somatic marker* (1).

Damásio defines an *emotion* as a perturbation in the state of a set of human body variables, caused by situations or thoughts. These variables include, for example, blood pressure, endocrinous glands activity, musculature parameters, to name but a few. Strong experiences lead to heavy perturbations of these variables, thus generating significant emotions. On the other hand, a *feeling*, according to Damásio, is considered an association of an emotion with the mental image of the situation that give rise to it. Damásio names a feeling that is kept in the human long-term memory as a *somatic marker*. From an artificial intelligence point of view, one can think those feelings as consisting of pairs (Perception, Emotion), where a *perception* is defined as the processed information that respects to a state of the world and the agent itself.

When a decision has to be made some neural dispositions, representing some of the previous acquired somatic markers, are triggered as a consequence of several possi-

ble future scenarios that are brought to the mind. When one of these scenarios is similar to the perception hold by a somatic marker the corresponding emotional variation is replicated in the human body. This allows a fast classification of that future scenario in terms of its desirability, and hence provides a benefit evaluation of the action that would lead to that hypothetical scenario. When a decision has to take place, emotions automatically discard some "action paths", while emphasising others, therefore contributing for narrowing the decision search space.

The model proposed here aims at the implementation, in an artificial agent, of the decision and learning capabilities provided by somatic markers in human beings. There are no references in this work to concepts like sadness, anger, fear or joy, since it is intended not to replicate human emotions on an artificial machine but rather to implement the functionality of emotions, namely their contribution to the decision making process.

In Section 2 the proposed architecture and all of its modules are presented, namely, the associative memory module, its management for finite resources, the decision module and the action switching module. In Section 3 some experiments performed to evaluate the performance of the proposed architecture are described and their results discussed. Finally, Section 4 presents some related work and Section 5 finishes the paper, drawing some conclusions and pointing directions to future investigation on the architecture presented in this paper.

2 The Proposed Architecture

An intelligent agent is usually considered as an entity that, for each perception, tries to maximize some kind of utility function over time, choosing its most adequate available action(2). Usually the instantaneous utility function is built upon a finite set of attributes, also called motivations

in this paper. Let us define the *Connotation Vector* as the vector collecting the attributes deviations from their optimal values, *i.e.*, deviation from the values for which the instantaneous utility function takes its maximum value. This vector has N_C components, where N_C is the number of attributes considered. It is easy to see that the instantaneous utility function is maximized when the connotation vector is equal to the null vector. The agent using this architecture should maximize

$$U = \frac{1}{T} \int_{t_0}^{t_0+T} u(t) dt, \quad (1)$$

where t_0 is the current time, T is a time horizon, U is the utility value and $u(t)$, the instantaneous utility, is obtained from the connotation vector according to

$$\begin{aligned} \mathbb{R}^{N_C} &\Longrightarrow \mathbb{R} \\ C(t) &\mapsto u(t) = \mu(C(t)) \end{aligned}, \quad (2)$$

where C , the connotation vector, is calculated using the function

$$\begin{aligned} \Omega_P &\Longrightarrow \mathbb{R}^{N_C} \\ P(t) &\mapsto C(t) = \chi(P(t)) \end{aligned}, \quad (3)$$

where P is a perception and Ω_P is the perception space. μ and χ are functions depending on the environment and on the agent objectives. Maximizing the instantaneous utility function is equivalent to minimizing the connotation vector, and the agent tries therefore to keep its connotation vector as close to the null vector as possible.

Notice that the emotional decision system of a human being also tries to keep a set of body variables as close as possible to their equilibrium values, usually called homeostatic values (1). There is a parallelism between the connotation vector of our artificial agent and the emotional response of an individual to some situation. In order to build a decision and learning system based on emotions and somatic markers two different mechanisms have to be implemented:

An associative memory: Somatic markers are essentially associations between perceptions and emotional variations. Therefore, a memory that holds a set of perceptions and the corresponding connotation vector variations must be built. This associative memory, given a perception P , should allow to estimate ${}^A\Delta C_P$, the connotation variation taken as a consequence of choosing action A . It should also provide ${}^A I_P$, a measure of the quality of the information used to estimate ${}^A\Delta C_P$.

A decision policy: Given ${}^A_i\Delta C_P$ and ${}^A_i I_P$ for every possible action A_i , which action should the agent carry out? That decision depends on C , the agent's current connotation vector, but the information ${}^A I_P$ also plays a role on this decision process — it introduces the notion of exploration, as it will be explained in section 2.3.

Fig. 1 provides a global view of the proposed model's architecture.

2.1 Estimation Using an Associative Memory

An associative memory Ψ is defined here as a set of quadruples

$$\psi_i = (t_i, P_i, A_i, \Delta C_i), \quad \text{with } 0 \leq i \leq N_\Psi,$$

where N_Ψ is the number of quadruples stored in memory, P_i is a perception acquired at instant t_i , A_i is the corresponding chosen action and ΔC_i is the connotation vector variation, taken as a consequence of performing A_i in the situation represented by P_i . The similarity between two pairs of perceptions and corresponding actions, (P_i, A_i) and (P_j, A_j) , is defined by

$$\rho_{ij} = \rho((P_i, A_i), (P_j, A_j)), \quad \text{with } \rho_{ij} \in [0, 1], \quad (4)$$

where ρ is an environment dependent similarity measure function which gives $\rho_{ij} = 0$ when two pairs are entirely distinct and $\rho_{ij} \approx 1$ when there is a perfect match between them. Then, for an arriving perception P , the connotation variation caused by selecting action A is estimated using

$${}^A\Delta\hat{C}_P = \frac{\sum_{i=1}^{N_\Psi} [\Delta C_i \cdot \rho((P, A), (P_i, A_i))]}{\sum_{i=1}^{N_\Psi} \rho((P, A), (P_i, A_i))}. \quad (5)$$

This estimate is obtained by simply averaging all records stored in memory, weighted by their similarity to perception P . Similar perceptions stored in memory cause the corresponding records to have a higher contribution for estimating the connotation variation.

An information measure is also obtained using

$${}^A I_P = \frac{1}{K} \sum_{i \in \Omega_k} \rho((P, A), (P_i, A_i)), \quad (6)$$

where Ω_k is the set formed by the K nearest neighbours of (P, A) , according to the similarity measure ρ . ${}^A I_P \approx 1$ when the associative memory has records similar to the pair (P, A) , meaning that the generated estimate of the connotation variation caused by action A in situation P is built using information from similar previous experiences; on the other hand, when ${}^A I_P \approx 0$ the estimate does not have access to similar situations, hence suggesting an unreliable estimation value. Note that the information gain, as defined by classical information theory, can be obtained by $I_c = 1 - {}^A I_P$, where I_c stands for "classical information". $I_c \approx 0$ when records similar to (P, A) are present in memory, meaning that there will not be a significative information gain if action A is chosen to be carried out in situation P (as the memory already contains the results of previous similar experiences).

Note that the estimation of the connotation vector variation is a classical statistical regression problem. Given a sample of points $(P_i, A_i, \Delta C_i)$ — the associative memory — it is desired to estimate a (possible stochastic) function $\Delta C = f(P, A)$ that maps the perception and action spaces to \mathbb{R}^{N_C} , where, as previously seen, N_C is the connotation vector dimension.

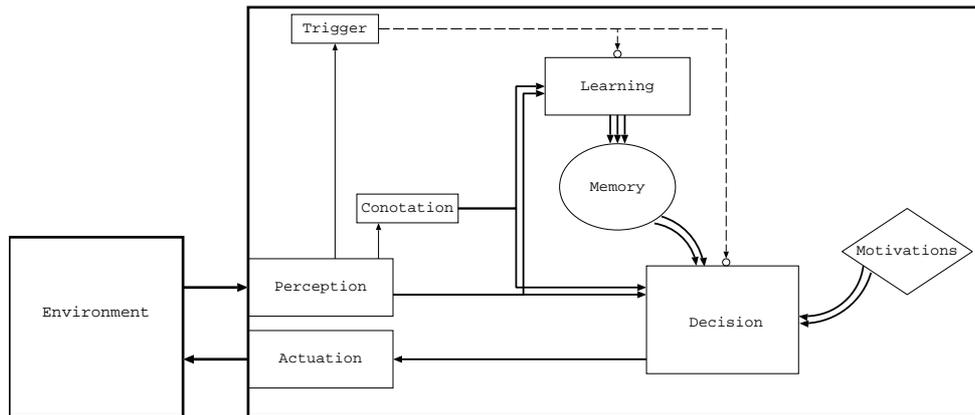


Figure 1: Architecture

2.2 Finite Resources: Memory Management

Computational limitations are inherent to any practical artificial agent, leading consequently to finite size associative memories. The special case of agents acting in real-time demanding environments strongly restricts the memory maximum capacity for which the computation of (5) is still performed in a reasonable time. In those cases an associative memory can easily become filled up. When a new record is ready to be inserted into memory, the agent must therefore pick and discard some stored quadruple. The choice policy of the record to be eliminated is a crucial one: on the one hand, it should not increase the mean estimation error over time and, if possible, should even contribute to decrease it; and on the other hand, a removal policy should not have a computation time higher than the estimation process itself. The complexity of the estimation performed using equation (5) is $\mathcal{O}(N_{\Psi})$ in terms of basic arithmetic operations and associative memory accesses. Several possible heuristics can be used:

Antiquity: The oldest record is always picked and removed. The associative memory thus becomes a *first-in first-out* queue;

Estimation Error: This heuristic picks the record whose elimination would produce the least variation in the estimate, *i.e.*, chooses a record that does not appreciably change the estimate supplied by associative memory, after the record being removed;

Variance: Records in memory where the local variability of ΔC is low are removed first, as they are unlikely to be useful in the estimation process;

Information: Records located in a highly dense region of the perception space have priority for elimination, since their deletion does not change noticeably the value of $^A I_P$ for perceptions P located in that region;

Meaning: This heuristic tries to preserve high $|\Delta C|$ records, *i.e.*, records associated with strong experiences.

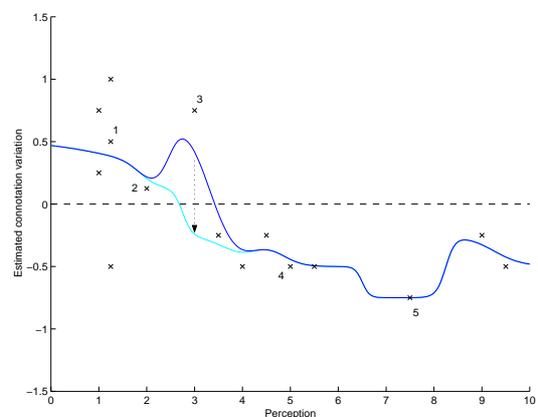


Figure 2: Connotation variation estimation for a 15 point memory. These points are represented by crosses, while the estimates with and without the presence of record 3 are represented respectively by the blue and cyan lines. Record 2 will be chosen for removal if meaning heuristic is used, as it corresponds to the smallest connotation variation. On the other hand, comparing points 1 and 4 shows that the latter has a lower local variance, and thus will be removed before the former if variance heuristic is used. Point 5 will hardly be eliminated if information based removal is carried out, due to its relative isolation. Finally, represented by an arrow is the estimation change produced by point 3 removal. Estimation error heuristic tries to remove points that do not change noticeably the estimation curve.

Fig. 2 illustrates the use of these heuristics. Estimation error based heuristic should pick a memory record i that minimizes the estimation variation over all the perception space,

$$H(i) = \sum_{k=1}^{N_A} \int_{\Omega_P} \left| A_k \Delta \hat{C}_P - A_k \Delta \hat{C}_P^{i-} \right| dP,$$

where N_A is the number of possible actions and $A_k \Delta \hat{C}_P^{i-}$ is the estimate obtained in the absence of record i . Computational constraints, however, force this heuristic to only

take into account the estimate error for the point i where the heuristic $H(i)$ is evaluated,

$$H(i) = \left|^{A_i} \Delta \hat{C}_{P_i} -^{A_i} \Delta \hat{C}_{P_i}^{i-} \right|. \quad (7)$$

The complexity of the error estimation based heuristic thus become $\mathcal{O}(N_\Psi)$, since complexity of (7) is $\mathcal{O}(1)$ if some auxiliary values are maintained in the associative memory records.

Local variance of record i is given by

$$S_i^2 = \frac{\sum_{j=1}^{N_\Psi} (\Delta C_j - \Delta \bar{C})^2 \rho_{ij}}{\sum_{j=1}^{N_\Psi} \rho_{ij} - 1} \quad (8)$$

where $\Delta \bar{C}$, the connotation variation local mean, is given by

$$\Delta \bar{C} = \frac{\sum_{j=1}^{N_\Psi} \Delta C_j \rho_{ij}}{\sum_{j=1}^{N_\Psi} \rho_{ij}}.$$

Choosing a record to eliminate based on local variance also is $\mathcal{O}(N_\Psi)$ if proper auxiliary values are kept in associative memory records. Information based heuristic, on the other hand, is $\mathcal{O}(N_\Psi^2)$ if (6) is used. Instead one can use the average similarity,

$$^A I'_P = \frac{\sum_{i=1}^{N_\Psi} \rho((P, A), (P_i, A_i))}{N_\Psi}, \quad (9)$$

to obtain a $\mathcal{O}(N_\Psi)$ information-based heuristic.

2.3 Decision

How do the agent select the action to execute, when, in a given situation represented by perception P , a connotation variation estimate, for every possible action A , $^A \Delta C_P$, is available? Recall that this agent tries to keep the connotation vector as close as possible to the null vector.

Consider $^A \hat{C}_P = C_P + ^A \Delta C_P$, the expected connotation vector if action A is performed. In most cases choosing an action that minimizes the Euclidian norm of $^A \hat{C}_P$ is inadequate, as this norm does not make a distinction between vector components with different importances. Suppose a weight w_i is assigned to each connotation component, with the usual restriction $\sum_{i=1}^{N_C} w_i = 1$. Selecting an action that leads to the most favourable expected situation becomes therefore dependent on the distance metric used for the connotation vector. One can consider, for example,

$$|C| = \sum_{i=1}^{N_C} w_i |c_i|, \quad (10)$$

where c_i is the i^{th} component of the connotation vector. On the other hand, if actions that lead to high absolute values for some of the components are to be avoided, then one can use the following metric:

$$|C| = \sum_{i=1}^{N_C} w_i c_i^2. \quad (11)$$

Since

$$\left| \frac{\partial |C|}{\partial ^A \hat{c}_i} \right| = 2w_i |^A \hat{c}_i|,$$

the derivative of the connotation norm is proportional to the absolute value of each connotation component. In this manner, vectors with a high absolute value of some of their components are effectively “punished” with a higher norm value.

When $|C|$ is normalized between 0 and 1 one can define $^A U_P$, the action A utility for perception P , as

$$^A U_P = 1 - |C|.$$

Normalizing the connotation components, $-1 \leq c_i \leq 1$, assures $|C|$ normalization when (10) or (11) are used.

Such a greedy strategy is not always the best thing to do, especially when the agent interacts with a complex and challenging environment. In those cases, preferring an exploration behaviour may reveal itself more useful. Remember that $^A I_P$ takes a value near 0 when the associative memory does not hold situations similar to the pair (P, A) ; in those situations, the agent should consider exploring the world.

The decision policy of the agent is consequently based on choosing the action A that maximizes

$$(1 - \beta) ^A U_P + \beta (1 - ^A I_P) \quad (12)$$

where $\beta \in [0; 1]$, an exploration coefficient, may vary with time.

2.4 Action Triggering

Deciding when to stop executing an action, and choosing and starting another one is a delicate problem. Periodically triggering the action switching does not allow an agent to quickly respond to unexpected events. Making the trigger period shorter does not solve the problem, as it may prevent an acceptable perception of action consequences, thus leading the agent to an erroneous learning. In practice, this may lead to a global behaviour not very different from a random behaviour selection (3).

Emotions are often pointed out as a source of behaviour interruption (3; 4; 5). Typically, a new behaviour is triggered when a significative emotional change happens. This corresponds to a new action being chosen whenever a relevant variation on the connotation vector is detected. Statistical tests are robust methods that take agent sensors noise into consideration, and consequently they are used in this architecture to detect connotation changes.

Let D_1 be a sample formed by the n most recent acquired connotation vectors, and let D_2 collect the n connotations directly preceding last action triggering instant. Suppose D_1 elements have a gaussian distribution, with mean μ_1 and variance σ^2 . Suppose also that D_2 elements are gaussian too, with same variance σ^2 but distinct mean μ_2 . A significative variation on the connotation vector is

assumed when hypothesis $\mu_1 = \mu_2$ is statistically rejected, i.e., when, for some pre-defined significance level α ,

$$|t_{obs}| > F_{t_{2n-2}}^{-1}((1-\alpha)/2),$$

with

$$t_{obs} = \frac{(\bar{D}_1 - \bar{D}_2) - (\mu_1 - \mu_2)}{\sqrt{(S_1^2 + S_2^2)/n}}, \quad (13)$$

where the statistic observed value, t_{obs} , has a *t-student* distribution with $2n - 2$ degrees of freedom and $F_{t_{2n-2}}^{-1}(x)$ is the *t-student* inverse distribution function. \bar{D}_1 and \bar{D}_2 are, respectively, D_1 and D_2 sample means.

Choosing a new action and learning the consequences of the previous one is hence triggered whenever the statistical test fails to prove an equality of means.

3 Results

In order to test the proposed architecture, a first experiment was conducted to check the estimation quality of the associative memory. Then, a soccer emotion-based agent was created and tested. This agent does not have any *a priori* knowledge of the consequence of its available behaviours, and therefore must learn how to score using exclusively the mechanisms developed in this work.

3.1 Estimation

Consider a scalar perception, gaussian with zero mean and standard deviation equal to 3. There are two possible actions: the first one is chosen with 70% probability, while the second has 30% chances of being executed. There is a connotation variation associated with every perception and with each action, as shown in Fig. 3. However, this connotation variation is corrupted with 0.2 standard deviation white noise. 10 000 points are generated, shown in Fig. 4, and presented sequentially to the associative memory. Two different size memories were tested, with 100 and 1 000 points. The similarity measure between two perceptions, ρ_{ij} , is assumed to be

$$\rho_{ij} = \begin{cases} e^{-\frac{(P_i - P_j)^2}{D}} & \text{if } A_i = A_j; \\ 0 & \text{otherwise,} \end{cases}$$

where D is a distance parameter whose value is equal to 0.5.

Fig. 5 illustrates the obtained estimates, for each of the five discarding policies presented in section 2.2. Fig. 6 shows the average estimation error, calculated for $-10 < P < 10$ using the true connotation variation value of Fig. 3 as a reference. Fig. 7 shows the mean value of $^A I_P$, using equation (6), with $K = 1$.

These results show how poorly a meaning-based heuristic performs. Keeping only the higher connotation variation records leads usually to severe estimation errors. Such a policy forces the agent to “see the world in black and

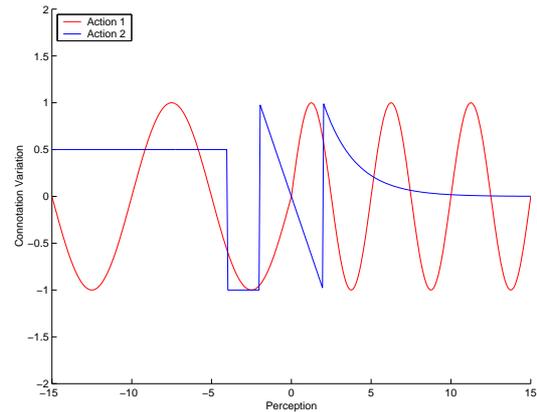


Figure 3: Connotation variation for each possible action.

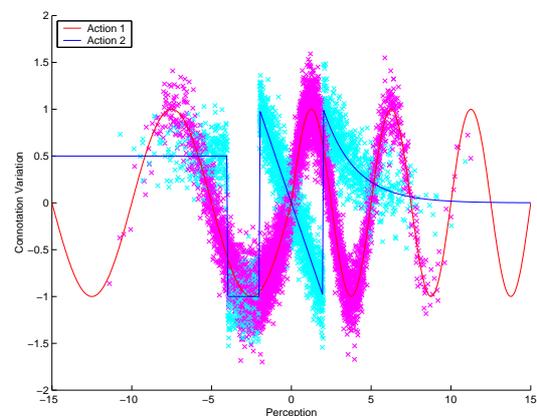


Figure 4: 10 000 points sample (corrupted with additive white noise).

white”. This policy, however, may prove itself useful when some critical (high ΔC) situations must be avoided, since such a “superstitious” agent may be able to prevent them. Information-based heuristic seems to work better when a low capacity memory is used; on the other hand, estimation error policy provides better results when a large associative memory is employed.

The similarity measure has an important role on estimation: increasing the value of D effectively acts as low-pass filter over the estimate. This may be desirable when handling with a sparse associative memory, but it might deteriorate the estimate when the memory size is large enough. Notice, in Fig. 8, how the estimate gets better if D is set to 0.1. Nevertheless, even with a memory with a dimension of only 100 records, estimation has proven to be quite accurate, as seen in Fig. 4.

3.2 Simulated Soccer

Robotic soccer provides a demanding and complex environment for artificial agents, thus being a natural test bed where decision and learning mechanisms can be developed and tested. The emotion-based architecture presented in this paper was applied to the simulation soccer league of

Robot World Cup Initiative, also known as *RoboCup* (6). Although simulation league teams comprise eleven players for each side, only two distinct situations were considered: a player with no opponents (a *solo* game) and a player against one opponent. Increasing further the number of players in the match necessarily leads to some questions and problems, such as the credit assignment problem in multi-agent systems. This is a complex problem that is not considered in this paper.

Solo Game: In the first experiment, an emotion-based agent is left alone in the field. It does not have a clue on its available actions consequences, although it has some *a priori* motivations: it wants to keep its stamina high, it wants to be near the ball and it wants the ball to be as near as possible to the opponent goal. These motivations are essential, since the agent needs a learning reference, *i.e.*, it must know the connotation of experienced perceptions. Its perception vector consists of information on positions of the ball and the player itself. There is also a set of available behaviours such as getting near the ball, dribbling to goal, kicking to goal and clearing the ball away, to name but a few.

The agent is then allowed to play a few matches — each match duration was set to 5 minutes — and its performance is evaluated. Table 1 shows the game results, while Fig. 9 presents the agent score evolution.

Game	Goal Score
1	8
2	13
3	22
4	21

Table 1: Lonely match: game results.

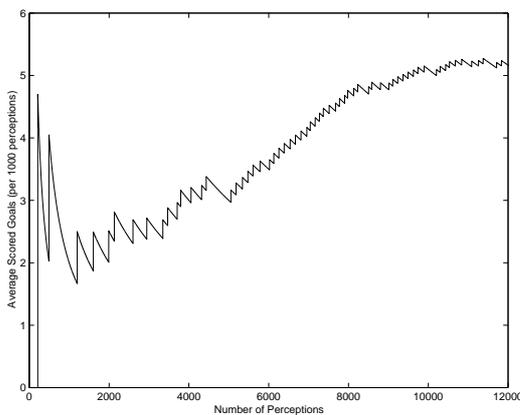


Figure 9: Lonely match: goal average.

Table 2 shows that some of the actions were completely discarded after some time. This may be considered an intelligent behaviour, since centering the ball or dribbling it

to the near corner hardly lead to a high motivation satisfaction. On the other hand, the emotion-based agent quickly develops a very efficient style: it constantly tries to get near the ball; when this happens, the agent then dribbles and/or shot it to opponent goal direction. Sometimes, however, it just stands facing the ball; this happens when the agent gets tired.

One vs. One: The presence of an opponent enlarges the perception and action sets. This opponent is modelled as a finite state machine with state transitions represented in Fig. 10.

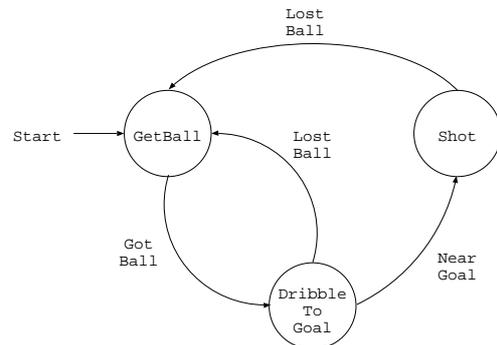


Figure 10: Opponent state machine.

Satisfying emotion-based agent motivations then becomes a trickier task, since the opponent player permanently tries to steal the ball, driving it afterwards to the agent own goal.

Table 3 presents game scores for a sequence of twelve matches of ten minutes each. Opponent player easily wins first games, since at that time the emotion-based agent is still trying to learn action consequences.

Game	Score
1	2–6
2	0–10
3	5–9
4	6–9
5	5–11
6	3–16
7	9–7
8	9–6
9	11–8
10	16–4
11	10–8
12	13–6

Table 3: One vs. one: score and ball possession.

However, after some games the emotion-based agent learned how to play against the reactive agent, beating it on the subsequently matches. Table 4 shows the dispended time of each action for the diverse matches.

4 Related Work

Until a decade ago, most of Artificial Intelligence researchers associated emotions to human rationality loss: emotions were believed to induct “a partial loss of control of high level cognitive processes” (7). Recent neurological evidence, however, has shown the fundamental role that emotions play in human decision and learning (1; 8; 9). There has been, especially since the publication of Damásio’s “Descartes’ Error” book, an increasingly interest on artificial emotions and their assistance to decision and cognition in artificial agent design.

Velásquez (10; 11; 12) presents an emotion architecture based on Damásio’s work, as well on the Society of Mind concept (13). Also following Damásio’s reference book, Gadanho *et al.* models a hormonal and emotional system where emotions provide a reinforcement value to a Q-Learning decision scheme (3). In this work, Gadanho also uses emotions to trigger learning and behaviour switching on an autonomous robotic agent. Ventura *et al.* proposes the double processing paradigm, where stimuli are processed simultaneously by a fast, perceptual layer — corresponding to primary emotions — and a slow, cognitive layer — inspired on Damásio’s secondary emotions (14; 15; 16; 17; 18). While the latter implements a learning mechanism based on somatic markers, the former, using *a priori* knowledge, can provide a quick response when the agent is confronted with a situation demanding urgent action. The perceptual image of a stimulus, created by the perceptual layer, also contributes to narrow the search space of the cognitive layer.

Finally, several other models of emotions have been built in the last few years, most of them oriented to human-machine interaction, such as the Oz Project (19) or the recognition of emotions developed by Picard (20).

5 Conclusions

The model presented in this paper is strongly inspired on Damásio’s concept of somatic markers. It uses an associative memory to implement those markers in artificial agents. One can think of such a memory as a collection of records, each of them corresponding to an “artificial somatic marker”. This paper also proposes some fast estimation and memory management mechanisms, which make this model suitable to real-time, “short time to think” agents.

Emotions also inspired the development of a statistical mechanism for deciding when to interrupt behaviour, *i.e.*, when to start executing another action.

It was shown in this paper how well a limited resourced memory estimates consequences of actions. Future work will fall upon the development of more sophisticated removal heuristics, as well as studying more deeply the kinds of relations that exist between memory size and similarity measure between perceptions, and how can they contribute to better estimation.

A soccer-playing emotion-based agent was also developed and successfully tested. This agent was able to effectively learn on a challenging and demanding environment, in the presence of an opponent whose objectives consisted only in preventing the emotion-based agent from satisfying its own. While developing such an agent, some difficulties were raised when defining an *exploration/exploitation* compromise. Nevertheless, the emotion-based agent was able to beat the reactive agent after some played matches.

The triggering mechanism presented in this paper does not solve the cause-effect problem, although it performs better than a periodic action switching. The proposed model does not consider either the credit assignment problem in multi-agent systems. Future work will also fall upon both these problems.

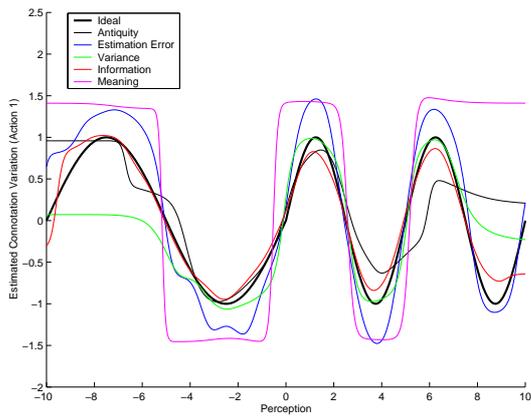
Acknowledgement

This work has been developed under the framework of a research project founded by the Portuguese Foundation for Science and Technology project PRAXIS/P/EEI/12184/1998.

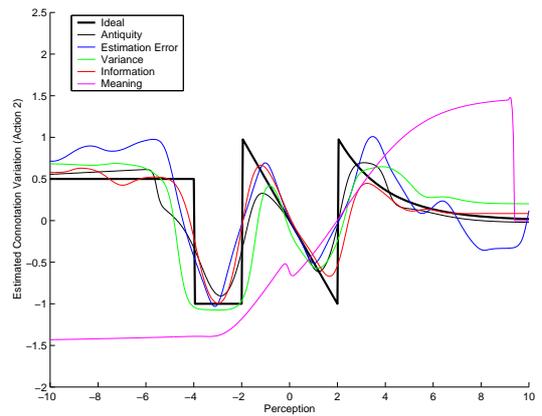
References

- [1] Damásio, A. *O Erro de Descartes: Emoção, Razão e Cérebro Humano*, Publicações Europa-América, Lisboa, 1995
- [2] Russel, S. and Norvig, P., *Artificial Intelligence: A Modern Approach*, Prentice-Hall International Editions, New York, 1995
- [3] Gadanho, S. and Hallam, J. *Emotion-triggered Learning in Autonomous Robot Control* in *Workshop: Grounding Emotions in Adaptive Systems*, pag. 31–36, Dolores Cañero, Chisato Numaoka, and Paolo Petta, ed., August 1998
- [4] Sloman, A. and Croucher, M., “Why Robots Will Have Emotions”, in *IJCAI’81 — Proceedings of the Seven International Joint Conference on Artificial Intelligence*, pag 2369–71, 1981
- [5] Simon, H., “Motivational and Emotional Controls of Cognition”, in *Psychological Review*, 74:29–39, 1967
- [6] <http://socrob.isr.ist.utl.pt>
- [7] Sloman, A., *What sort of control system is able to have a personality?*, 1995
(ftp://cs.bham.ac.uk/pub/groups/cog/_affect/Aaron.Sloman.vienna.ps.Z)
- [8] Damásio, A. *O Sentimento de Si*, Publicações Europa-América, 2000
- [9] LeDoux, J., *The Emotional Brain*, Simon & Schuster, New York, 1996

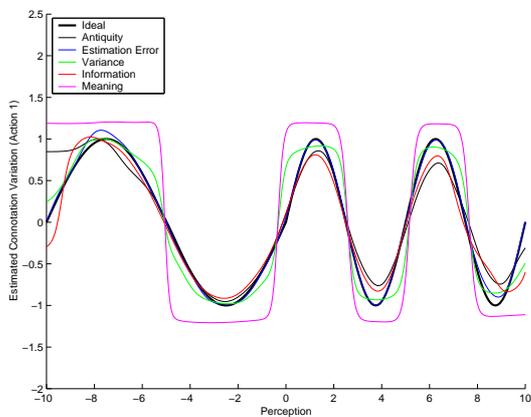
- [10] Velásquez, J., *Modeling Emotion-Based Decision-Making*, 1998
(<http://alpha-bits.ai.mit.edu/people/jvelas/research.html>)
- [11] Velásquez, J., *When Robots Weep: Emotional Memories and Decision-Making*, 1998
(<http://alpha-bits.ai.mit.edu/people/jvelas/research.html>)
- [12] Velásquez, J., *Modeling Emotions and Other Motivations in Synthetic Agents*, 1997
(<http://alpha-bits.ai.mit.edu/people/jvelas/research.html>)
- [13] Minsky, M., *The society of mind*, Simon and Schuster, New York, 1985
- [14] Ventura, R. *Emotion-Based Agentes*, Master Thesis, 2000
- [15] Ventura, R. and Pinto-Ferreira, C., “Emotion-based agents”, in *Proceedings AAAI-98*, pag. 1204, AAAI, AAAI Press and The MIT Press, 1998
- [16] Ventura, R. and Pinto-Ferreira, C., “Meaning Engines — Revisiting the Chinese Room”, in *Workshop: Grounding Emotions in Adaptive Systems*, pag. 68–70, Dolores Cañamero, Chisato Numaoka, and Paolo Petta, ed., Agosto 1998
- [17] Ventura, R., Custódio, L. and Pinto-Ferreira, C., “Artificial Emotions — Goodbye, Mr. Spock!, in *Progress in Artificial Intelligence, Proceedings of IBERAMIA’98*, pag. 395–402, Ed. Colibri, 1998
- [18] Ventura, R., Custódio, L. and Pinto-Ferreira, C., “Emotions — The missing link?, in *Emotional and Intelligent: The Tangled Knot of Cognition*, Dolores Cañamero, ed., pag. 170–175, 1998
- [19] Bates, J., Loyall, A. and Reilly, W. *An Architecture for Action, Emotion, and Social Behaviour*, 1992
(<http://www.cs.cmu.edu/Groups/oz/papers/CMU-CS-92-144.ps.gz>)
- [20] Picard, R., *Affective Computing*, 1995 (<http://www.media.mit.edu/~picard>)



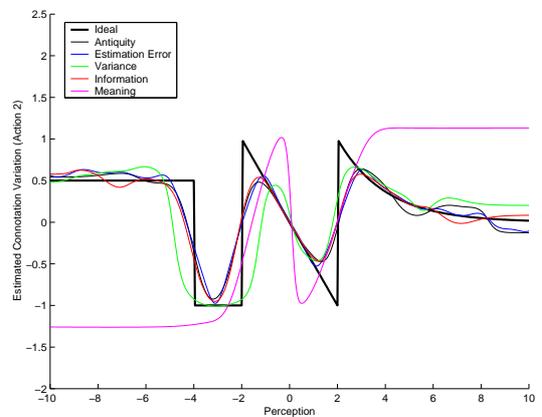
(a) Action 1, 100 points capacity associative memory.



(b) Action 2, 100 points capacity associative memory.

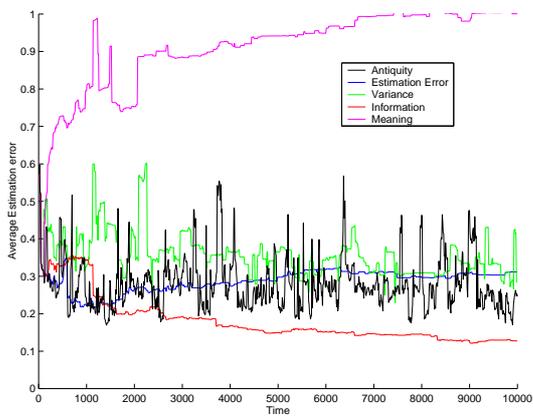


(c) Action 1, 1 000 points capacity associative memory.

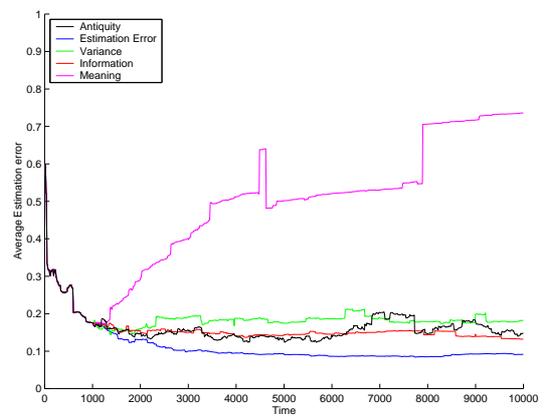


(d) Action 2, 1 000 points capacity associative memory.

Figure 5: Connotation variation estimate, ${}^A\Delta\hat{C}_P$.

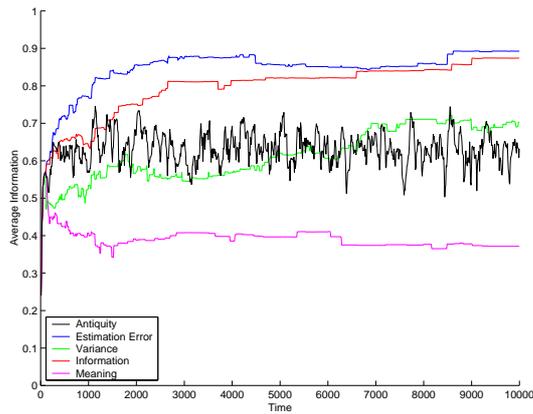


(a) 100 points capacity.

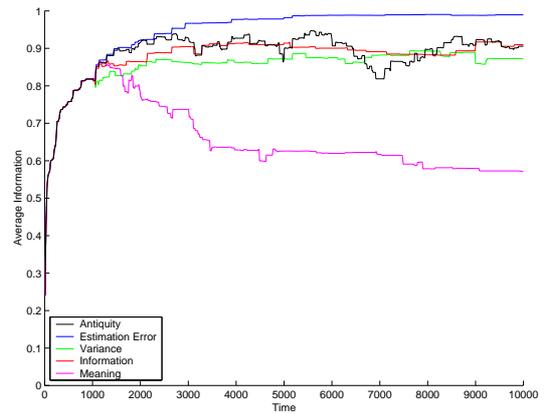


(b) 1 000 points capacity.

Figure 6: Estimate average error over time.

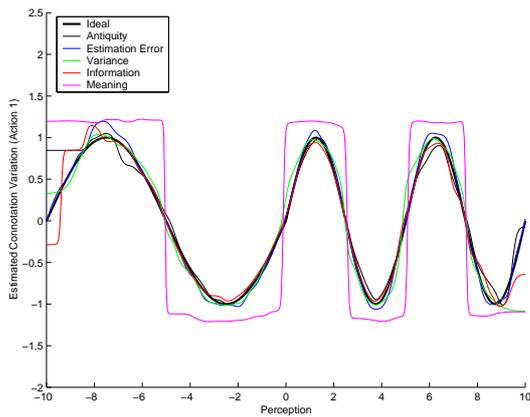


(a) 100 points capacity.

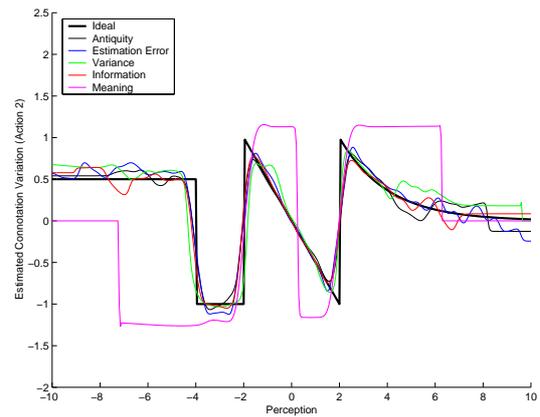


(b) 1 000 points capacity.

Figure 7: Average information $^A I_P$.



(a) Action 1, 1 000 points capacity associative memory.



(b) Action 2, 1 000 points capacity associative memory.

Figure 8: Connotation variation estimate, $^A \Delta \hat{C}_P$ ($D = 0.1$).

Action	Dispended time(%)			
	Game 1	Game 2	Game 3	Game 4
<i>GetBall</i>	48.5	49.0	55.1	51.2
<i>FaceBall</i>	14.1	13.7	9.7	17.0
<i>HoldBall</i>	3.1	3.1	0	0
<i>DribbleToFarCorner</i>	6.1	3.1	0.5	0
<i>DribbleToNearCorner</i>	5.6	3.1	0	0
<i>DribbleToGoal</i>	4.3	4.3	8.8	7.8
<i>Shot</i>	9.2	20.5	26.3	23.9
<i>ClearBall</i>	6.1	0.6	0	0
<i>CenterBall</i>	3.1	2.5	0	0

Table 2: Lonely match: time dispended to each action

Action	Dispended time(%)					
	Game 1	Game 2	Game 3	Game 4	Game 5	Game 6
<i>Shot</i>	3.4	1.2	4.2	1.8	1.3	1.0
<i>ClearBall</i>	3.8	1.9	2.8	0	0.4	0
<i>CenterBall</i>	2.2	0.4	0	0.4	0	0
<i>GetBall</i>	27.8	34.0	40.7	48.9	54.1	50.8
<i>FaceBall</i>	9.1	4.3	1.1	1.1	2.6	1.3
<i>HoldBall</i>	11.8	4.3	0.7	0.4	0.9	0.3
<i>DribbleToFarCorner</i>	11.8	9.3	2.8	1.8	0.9	1.9
<i>DribbleToNearCorner</i>	3.1	10.0	5.2	5.1	0.4	10.0
<i>DribbleToGoal</i>	9.6	14.3	13.7	23.9	33.6	22.5
<i>TrackOpponent</i>	9.6	8.9	12.2	13.0	0.9	6.8
<i>MarkOpponent</i>	7.6	11.6	16.5	3.6	4.8	5.5

Action	Game 7	Game 8	Game 9	Game 10	Game 11	Game 12
<i>Shot</i>	3.4	16.3	20.0	18.4	19.7	21.9
<i>ClearBall</i>	0.8	0	0	0.2	0.3	0.3
<i>CenterBall</i>	0	0	0	0.7	0	0
<i>GetBall</i>	46.8	50.8	54.2	50.5	58.0	61.2
<i>FaceBall</i>	0.8	2.1	0.3	0.7	0.9	0
<i>HoldBall</i>	0.4	0.6	0.6	0	0	0
<i>DribbleToFarCorner</i>	0.4	1.2	1.7	1.8	1.7	3.9
<i>DribbleToNearCorner</i>	6.1	1.5	3.2	1.6	1.4	0
<i>DribbleToGoal</i>	35.0	22.1	19.7	24.4	12.2	6.9
<i>TrackOpponent</i>	5.3	5.1	2.3	0.4	4.1	5.8
<i>MarkOpponent</i>	1.1	0.3	0	1.4	1.7	0

Table 4: One vs. One: time dispended to each action

Computational Models of Emotion for Autonomy and Reasoning

Darryl N. Davis and Suzanne C. Lewis

Department of Computer Science, University of Hull,
Cottingham Road, Kingston-upon-Hull, UK, HU6 7RX.

D. N. Davis@dcs.hull.ac.uk, <http://www2.dcs.hull.ac.uk/NEAT/dnd/index.htm>

S. C. Lewis@dcs.hull.ac.uk, <http://www2.dcs.hull.ac.uk/NEAT/index.html>

Keywords: Emotion, Computational Models, Autonomy, Perception, Reasoning

Received: October 25, 2002

Recent evidence suggests that the emotions play a crucial role in perception, learning and rational decision making. Despite arguments to the contrary, all artificial intelligent systems are, to some extent, autonomous. This research investigates how emotion can be used as the basis for autonomy. We propose the use of an emotion-based control language that maps over all layers of a computational architecture. We report on how theoretical work and both design and computational experiments with this concept are being used to direct perception, behavior selection and reasoning in cognitive agents.

1 Introduction

Definitions of intelligence in artificial systems have involved the use of many concepts including deliberation, reasoning [19] and the selection of appropriate behaviors for any given situation [4, 17]. In reasoning, information is granted belief status, either partially in probabilistic systems or wholly in logic based reasoning systems, and used as the basis for further deliberation. This deliberation may give rise to altered belief states and leads to the selection of goals, plans and behavior. Typically an agent chooses from alternative responses because of design decisions or learning. The choice is made on the basis of information or control metrics or a-priori ranking of alternative behaviors.

Underlying these perceptual and reasoning processes is the concept of autonomy. Alterman [1] suggests task-effective artificially intelligent systems need not be designed in terms of autonomy. Intelligence arises out of the interaction of the system and the user. However when system goals and resource allocation are in conflict, considerable interaction is required with a user. This interaction, itself a system goal (whether implicit or explicit), may never be satisfied unless the system can decide to perform the appropriate actions. To decide between actions, given that not all preconditions for any action will be specified at design time, requires the system to be, in some sense, autonomous.

Emotions and their nature have been studied for a considerable time, with many contrasting theories and views of emotion being formed. A traditional perspective of emotion is of something that is irrational and detracts from reasoning. However, recent evidence [8] suggests that emotions are an essential part of human intelligence, and play a crucial role in perception, rational decision-making and learning.

Most major current theories of emotion agree that emotions constitute a very powerful motivational system

that influences perception and cognition in many important ways. For example neurons in the amygdala are driven particularly strongly by stimuli with emotional significance, indicating an important role in the coding of the emotional significance of sensory data. Further research suggests that motivation and emotion serve as filters that guide perception and affect the evaluation of perceptual information [3]. This view is supported by Izard [16] who argues that emotion is a guiding force for perception. If emotion is a primary source of motivation, it must play a significant role in both initiating and providing descriptors for the types of disequilibria described by Mearleu-Ponty [18] as underlying behavior in biological agents.

From a computational perspective, Sloman considers that intelligent machines will necessarily experience emotion (-like) states [24]. Following on from Simon [23], this developing theory of mind considers how perturbant control states ensue from attempting to achieve multiple goals, or goals at odds with resource availability and environment affordances. Perturbant states will arise in any information processing infrastructure where there are insufficient resources to satisfy all current and prospective goals. This will occur not only at the deliberative belief and goal management levels but over all layers of the architecture as goals are mapped onto internal or external behaviors and actions. An agent must be able to recognise and regulate these emotion-like states or compromise its task effectiveness.

The aim of this research is to investigate theories of emotion and understand how they can be used to underpin computational autonomy, to direct and inform perception and behavior selection and to form a better model of computational reasoning. This paper describes this ongoing research and the integration of an emotional model into two different types of computational architectures.

2 Psychology and Emotions

Research has shown that emotion affects many different aspects of cognitive function including memory [5], reasoning and social interaction [15]. There has never been any doubt that emotion disrupts reasoning under certain circumstances and that misdirected or uncontrolled emotion can lead to irrational behavior. However, evidence from Damasio and other sources also suggests the contrary and that emotions play a fundamental role in rational and intelligent behavior such as decision-making and reasoning.

The Somatic Marker Hypothesis [8], for instance, states that decisions, made in circumstances whose outcome could be potentially either harmful or advantageous, and which are similar to previous experience, induce a somatic response used to mark future outcomes. When the situation arises again the somatic marker will signal the danger or advantage. Thus, when a negative somatic marker is linked to a particular future outcome it serves as an alarm signal to be wary of that particular course of action. If instead, a positive somatic marker is linked it becomes an incentive to make that particular choice.

The appraisal approach to emotion has cognition as the core element in emotion. The OCC (Ortony, Clore and Collins) model [21] synthesises emotions as outcomes to situations. Emotions arise out of a valenced reaction to situations consisting of events, objects and agents. The emotion type elicited is dependent upon appraisals made at each branch of the model. The OCC model allows for an emotional state to be a situation itself, so emotions can trigger additional emotions or the same one repeatedly. The OCC model is well suited to computational modeling as shown in the work of Elliot [11].

The goal-oriented approach suggests that emotions arise from evaluations of events relevant to goals. Again cognition is central to the elicitation of emotion. Oatley and colleagues [20] argue that emotions are caused by cognitive evaluations that may be conscious or unconscious. Each kind of evaluation gives rise to a distinct signal that reflects the priority of the goal, which then influences resultant behaviors. Frijda [13] uses a similar definition of emotion and states that certain stimuli elicit certain emotional phenomena because of the individual's concerns and the relevance of the stimuli to the satisfaction of these concerns.

Duffy posed the question, at which particular degree does a characteristic become an 'emotion' or at which degree is it a 'non-emotion' [10]. For example, a raised heartbeat is characteristic of both emotional and non-emotional behavior. When does the difference in the characteristic occur? Is emotion a distinguishable state or a difference in the degree certain response characteristics exhibit?

According to Duffy, the phenomena that are described as emotions occur in a continuum or a number of continua. The responses called 'emotional' do not appear to follow different principles of action to other responses of the individual. She states that all responses, 'emotional' or 'non-emotional', are reactions of an organism as it adapts

to a situation. Emotion represents a change in the energy level, or the degree of reactivity of an individual. For example, situations, which are interpreted as threatening, are characteristically responded to with increased energy. Small changes in energy level may occur during 'interest' or 'boredom', whereas 'anger' is associated with a more extreme change.

Duffy supports the goal-oriented view that emotions are only experienced in situations of significance to the individual. The intensity of the 'emotion' is proportional to the degree of importance associated with a particular goal and the degree of threat or promise the situation bears for that goal. The emotion experienced is also affected by the background and information that the individual has about the particular situation.

Many theories use the concept of basic emotions. For example, the OCC model contains twenty-two different emotion types. Oatley and Johnson-Laird cite four basic emotions derived from evolutionary origins: happiness, sadness, fear and anger. A further five are derived from innate biological substrates: attachment, parental love, sexual attraction, disgust and interpersonal rejection. However, other theorists question the notion of basic emotions. Scherer and Duffy oppose the view of basic emotions and examine evidence that emotions are patterns of interrelated changes. Using basic emotions in a theory can lead to what Scherer calls 'bunching' of the different emotional states around a limited number of types. Conversely, the scope of emotional states in both Duffy's and Scherer's theories is considerably broader.

Scherer points towards the existence of a large number of universal 'response elements' as opposed to basic emotions [22]. His concept of modal emotions attempts to address many questions. For example, why does the same situation not necessarily provoke the same emotional expression nor the use of the same label in two individuals? Like Duffy, Scherer sees emotion as a number of changes that occur over time in response to an event. He defines emotion as "*a sequence of interrelated synchronised changes in the states of all organismic subsystems in response to the evaluation of an external or internal stimulus event that is relevant to central concerns of the organism*". The emotional state results from the cumulative evaluation of relevant changes in internal or external stimulation. Scherer proposes that such organisms make five types of checks: novelty, intrinsic pleasantness; relevance to meeting plans; ability to cope with the perceived event; and compatibility of the event with self-concept and social norms. An appraisal according to these checks is carried out which elicits an emotional response. Scherer believes that the information from these checks is needed in order to choose how to respond. Some combinations of evaluation checks would be frequently encountered, giving rise to the same recurring patterns of state changes. The term 'modal emotions' refers to states resulting from these recurring stimulus evaluation check patterns. Although some patterns occur more frequently, the number of potential emotional states is virtually infinite.

3 Autonomy, Goals and Emotion

Many frameworks are used for thinking about, designing and building intelligent systems. The use of rational BDI (Belief-Desire-Intention) models [6] is understandable, as they provide formal systems with well-defined properties. The limitations of such systems, e.g. logical omniscience and resource constraints, are known. Goal competition due to incompatible goals, insufficient resources or skills is a major research issue. Ferber [12] categorizes goal interaction in multi-agent systems as one of three categories: indifference, cooperative and antagonistic. Certainly in the latter case, and even for cooperative agents or goal interaction, perturbant states can arise. Such agent societies and intelligent systems need some means to manage these states or risk compromising their autonomy and reactivity, and hence their task effectiveness. Even the most rational agent architecture will be compromised if it lacks the mechanisms to cope with the emergent effects of antagonistic goal conflicts.

One stance is to place a computational analogue to emotion at the core of an agent. This provides an agent with an internal model that maps across different levels and types of processing. Emotion provides an internal basis for autonomy and a means of valencing information processing events. It provides an internal model of use in ordering motivation and goals, and the means for choosing actions and regulating behavior. This emotional core can be used to recognise and categorise transient, episodic, trajectory and persistent control states.

Sloman [25] also differentiates between episodic and persistent mental phenomena. His architectures for functioning minds include primary, secondary and tertiary emotions. Primary emotions are analogous to arousal processes in the emotion theories introduced above and have a reactive basis. Secondary emotions are those initiated by appraisal mechanisms and have a deliberative basis. Tertiary emotions are cognitive perturbances – typically negatively valenced emergent states – such as arising from goal or motivator conflicts in an information processing architecture, for example a multi agent society. Any agent architecture that supports multiple motivations or goals is liable to this type of dysfunction. Perturbant states can arise through resource inadequacy or mismanagement while pursuing multiple and not necessarily incompatible goals. Most computational systems face this type of problem even if their underlying theory does not. Possible solutions are particularly relevant to the design of goal-oriented and agent systems.

4 Four Layer Computational Architecture

Earlier research on agents focused on an architecture that supports motivation [9]. The current framework builds on that architecture. It is used to pursue alternative computational perspectives on architectures of mind. Here the interplay of cognition and emotion is emphasized through mechanisms that support appraisal, motivation, tasks and roles. Emotions are accepted to be

part mental (appraisal) states with descriptive (valencing) and causal (arousal) processes. This concept is used to provide a control or regulatory framework to model the different forms of emotion inducing events. The fundamental tenet of this work is that all agent events and actions, internal and external, can be described in terms of this model of emotion.

A salient feature of many definitions of emotion is that they are described in terms of goals, roles (or norms) and responsive behaviors. This enables different aspects of motivational behavior to be consistently defined over different levels of the architecture in terms of an emotion-based control language. Global drives are those associated with the agent's overall and persistent purpose. Temporally-local drives are related to ephemeral states or events within the agent's environment or itself. Emotional autonomy allows an agent to select and attempt to maintain an ongoing globally-temporal *disposition* towards its roles. The nature of this is temporarily affected and perhaps modified through current goals and motivations. Over time events occur that modify, stall, negate or satisfy goals. Such events can be described within a model of emotion. An emotion-based control language can therefore be used to mediate the interaction of global roles and the temporally-local drives that reflect the current focus of the agent.

An agent's internal environment can be defined in terms of its perception of external events, objects and agents and the behaviors (whether internal or external) they afford. Such descriptions can be organised according to control state theory [9]. The control language used to navigate this internal environment needs to be consistent across many levels and types of control state from autonomous reflexes to extensive deliberation associated with goal satisfaction or belief management. Various combinations of qualitatively different behavior are required of an agent as it attempts to achieve different categories of goals associated with a role. Different problem-solving trajectories, described in terms of goal-achieving behaviors, exist for any one role. A greater range exist where an agent has multiple and not necessarily contingent roles. Some trajectories while impossible are supported or attended to for any number of reasons; for example, the motivational intensity associated with a preferred goal or role. The possible trajectories depend on an agent's design. An agent is autonomous to the extent that it can choose to pursue specific motivational trajectories. An agent is rational to the extent that it follows feasible or achievable trajectories.

Figure 1 shows emotion used as the core to a motivation based model of agenthood. This architecture emphasizes four distinct processing layers: a reflexive layer analogous to the autonomic systems in biological agents, a reactive or behavioral layer, a deliberative layer and a regulating reflective layer. The broad picture is of high and low level processes co-existing and interacting in an asynchronous, parallel and holistic manner. The majority of the higher level processes tend to remain dormant and

state persistent; activated only when sufficiently required. The agent's processing exists in relation to the agent's environmental stance; i.e. what roles the agent has adopted, what objects, agents and events occur in the environment and how they affect the logistics of goal and role satisfaction. Motivator processing, planning and other cognitive processes are not merely abstract, nor just reactions to the current state of an agent's external environment but exist in relation to an agent's long term goals. Motivations, goals and the behaviors they subsume are all influenced by components of the emotion engine.

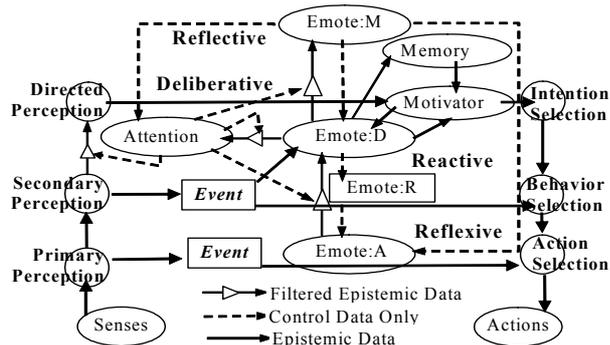


Figure 1. Sketch of the emotion engine based four-layer architecture. Overall the process is of information assimilation and synthesis, and information generation that typically map onto internal and external behaviors

If emergent behaviors are to be recognized and described in terms the emotion based control language and then managed, there must be a design synergy across the different layers of the architecture. Processes at the deliberative level (for example *Emote:D*, *Attention* and *Motivator* in Figure 1) can reason about emergent states arising from anywhere in the architecture using explicit (motivator or goal) representations (see [9]) and the internally consistent control language. In an earlier architecture, the reflective processes (*Emote:M*) were used to classify the processing patterns of the agent in terms of combinations of a set of basic emotions and favored emotional responses (or disposition). Subsequent rejection of the concept of basic emotions, for theoretical and computational reasons, required a redesign of this component. The emotion-changing reactive behaviors (*Emote:R*) are used to pursue a change in disposition through changing the functional behavior of the lowest-level autonomous processes (*Emote:A*). This module is modeled using multiple communities of cellular automata (or *hives*). The behaviors associated with this module, and set by the *Emote:R* module, are those that govern the internal behavior of single cells, the communication between adjoining cells in communities and inter-community communication. Emotion is discretely valenced at the cell level as positive-neutral-negative. Ordinal measures across the valence of all the cells at the community level provide the basis for ascending control signals. Various threshold models have been used to determine if arousal occurs; for example a community of cells with a high aggregate valence, or a

high degree of valence contrast across the cell community.

Emotions can be instantiated by events both internal and external at a number of levels, whether primary, e.g. ecological drives, or by events that require substantive cognitive processing. Emotions can be invoked through cognitive appraisal of agent, object or event related scenarios, including for example the unwanted postponement or abandonment of a goal. To move to a preferred aspect of the possible emotional landscape, an agent may need to instantiate other motivators and accept temporarily unwanted dispositions. An agent with emotional autonomy needs to accept temporary emotional perturbation if it facilitates goal satisfaction at some future time.

In the model shown in Figure 1, intense emotions or arousal events effectively override the reactive-level filters, activating the deliberative components of the emotion engine. Deliberative appraisal of an emotion inducing event can initiate lateral activation at the deliberative layer, affecting memory, attention and motivator management. Memory responds to emotional context as an aid to the storage and recall of memories about external events, objects and agents. Attention management makes use of the emotional state of *Emote:D-R-A* complexes to provide a semantic context for motivator filters, and set the quantitative emotion filters. The intensity levels of these filters are set in response to the *Emote:D* mechanisms and the reflective component (*Emote:M*) of the emotion engine.

Computational experiments have used both sets of basic emotions and type-less emotion arousal models. Early experiments found that from any given state, a hive rapidly achieved a steady (continuous or oscillating) state. By changing the currently extant behavior set, or by communicating with another hive, transitions to the same or other steady states always occurred. Approximately 20,000 transition possibilities exist. Rules are used to select different hive dispositions and transitions. Similarly, through the modification of the internal state of a small number of cells, the emotion engine moves to a closely related but preferred state. This is analogous to the modal responses described in the Scherer model of emotion.

5 CRIBB and Emotion

5.1 The CRIBB Model

CRIBB (Children's Reasoning about Intentions, Beliefs and Behavior) is a computer model based upon a general sketch for belief-desire reasoning in children [2]. It simulates the knowledge and inference processes of a competent child solving false-belief tasks [27]. A simulation run in CRIBB starts by giving propositions containing facts and perceptions about some scenario in sequential steps according to the time interval in which the propositions arise. On the basis of the given propositions and the inferences drawn, CRIBB answers test questions about the cover story. The questions can be

about its own beliefs or about the intentions, beliefs and behavior of another person in the scenario.

CRIBB represents propositions about physical states of a given situation and the intentions, beliefs, perceptions and behavior of others. Its knowledge base consists of four types of practical syllogisms and three other inference schemata, which represent the relations between these propositions. Practical Syllogisms denote knowledge about the relations between intentions, behavior and beliefs of another person. The three other classes of inference schemata relate perception-belief, belief-time and fact-time. These are split into primary and secondary representations. Primary representations are the system's own beliefs about the situation and the behavior of other people. Fact-time inferences, propositions about facts along a time scale, are classed as primary representations. Belief-time and perception-belief inference schemata are both types of secondary representation as they contain beliefs about the system's own and others' beliefs. A further element of CRIBB is a consistency mechanism that detects and resolves contradictions in belief sets. This is invoked each time a new proposition is added, in order to ensure the consistency of its knowledge base.

5.2 Extending CRIBB with Emotions

Bartsch and Wellman's model [2] for belief-desire reasoning includes an emotion element that CRIBB does not implement. Consequently, CRIBB can be extended to perform some experiments with different models of emotion. Certain theories of emotion are more suitable for implementation in CRIBB. Both the appraisal and the goal-oriented approach cite cognition as the core of emotions. The scenarios used in CRIBB are based around a goal-oriented structure. The existence of intentions in CRIBB is comparable to a goal state. Therefore, implementing a goal base and using tenets of the goal-oriented approach to emotion is a suitable foundation on which to base a model of emotions.

Gibson's theory of direct perception [14] can be used to extend CRIBB's perception-belief mechanism to incorporate emotional capabilities. Gibson describes how sensory data when perceived is given affordances and valences. An affordance is something that refers to both the environment and the perceiver in a way that no existing term does. They are properties taken with reference to the observer. Affordances of the environment are what it offers, what it provides, either for good or bad. For example, if a surface is horizontal, nearly flat and sufficiently extended and if its substance is rigid then the surface affords support. Affordances can also be valenced. The theory of affordances can be extended to allow emotion to exhibit an effect on perception of the environment according to the importance of needs, goals and plans to the individual. The following extension to CRIBB does this.

When CRIBB is given a proposition, a belief is inferred from this. The consistency of this belief is checked with the existing set of beliefs. If no contradiction is found then the new proposition is added to the belief set. If

there is a contradiction then this is resolved and the most certain belief is added to the belief set. For example:

$$P := \{r, s, q, p\}$$

$$B := \{\neg p\}$$

$$P \otimes B \rightarrow B'$$

$$B' := \{r, s, q, p\}$$

B is the existing belief set and P is perception set. The new set B' contains the system's new belief set with all possible contradictions resolved (p is preferred to $\neg p$). In this scenario each perception of the world has equality.

Rather than attempt to completely and accurately model the agent's world, emotion can be used to guide attention so an agent is drawn to aspects of the environment deemed to be of importance. Assigning an emotional affordance will enable a process by which perceptions can be filtered according to their importance. Hence:

$$P := \{r, s, q, p\}$$

$$E := \{\text{importance}(\text{high}, p), \text{importance}(\text{low}, r)\}$$

$$B := \{\neg p\}$$

$$E \otimes P \rightarrow EP$$

$$EP := \{p, s, q, r\}$$

$$EP \otimes B \rightarrow B'$$

$$B' := \{p, s, q, r\}$$

The perception set, P, contains the same perceptions as before. However, the order in which the perceptions are processed can be changed according to the emotional affordance, E, attached to each one. The new belief set, B', contains the perceptions which have been processed in the order that accords with their emotional significance to that individual.

Emotion can be used to extend the belief and perception mechanism of CRIBB further. Consider a perception received from one source and a further perception, from a different source, that contradicts this. If, through the contradiction mechanism in CRIBB, the first perception is found to be false then this may affect the truth value of any beliefs and perceptions from that particular source. In other words CRIBB will now be less inclined to believe information received from this source. Or conversely, the information from some source may now be considered more reliable than before. This situation can be represented in CRIBB by creating an emotional correspondence for each possible source. This would give an indication of the likelihood of information from this source being either true or false.

Ongoing work on this model is using various agent test-beds to gather metrics to inform our research. One particular experiment makes use of the fungus eater scenario [26, 28]. Early results suggest that the addition of emotion to CRIBB results in a more effective use of resources to achieve tasks, with a more efficient resolution of goal conflicts. For example in an environment evenly populated with fungus and ore, both CRIBB and ECRIBB agents achieve their task goals (the collection of ore). The ECRIBB agents however make more effective use of available energy sources (the

fungus). In terms of the earlier arguments about goal conflicts, the emotion model augments the agent's autonomy and facilitates the resolution of goal conflicts. Experimentation continues to determine how an agent can adapt to environment changes through the modification of the emotional valences associated with perceptual affordance and goal importance.

6 Future Directions and Discussion

The research described here reflects on two perspectives to the integration of emotion into cognitive agents. The architecture of Figure 1 has limited reasoning capabilities (more limited than the implementations described in [9]), but makes use of a coherent emotion-based control language. CRIBB on the other hand is a serial deliberative (BDI) model that does not try to provide a coherent story for all the types of control states identified in [23] and [9]. A complete architecture would subsume the use of emotion as a control language, and indeed the entire BDI reasoning processes of CRIBB. The current separation enables complimentary work to progress independently. As this research develops the two complementary architectures can be integrated.

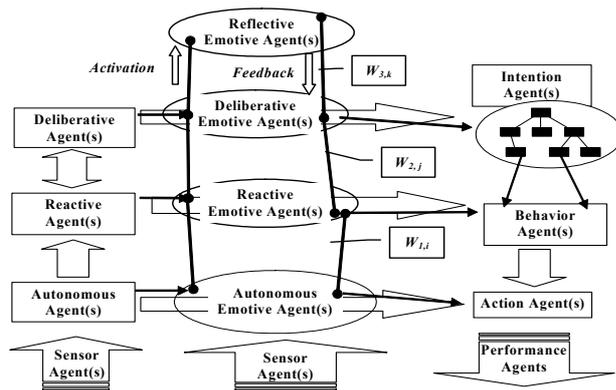


Figure 2. A distributed model that draws together the emotion engine of Figure 1, CRIBB and earlier work.

Duffy's theory that emotions occur in a continuum or a number of continua, which includes both 'emotional' and 'non-emotional' behavior, views emotion as a more integrated part of behavior rather than a separate element. This view is also supported by Scherer who argues the pattern of all synchronised changes in the different components over time constitutes an emotion. Both of these theories can be viewed as 'distributed' models. Using a distributed emotion model is problematic in CRIBB as CRIBB's serial reasoning model is not amenable to the asynchronous re-evaluation of plans and information processing that takes place within a distributed system. No such problem exists with the four layer architecture – it is designed to be asynchronous and distributed.

Oatley and Johnson-Laird propose that each goal and plan has a monitoring mechanism that evaluates events relevant to it. This mechanism broadcasts to the whole cognitive system, allowing it to respond to change as it occurs. For a distributed model of emotion, the

monitoring system would need not only to communicate goals and plans but also respond to each sub-system. CRIBB can be readily extended with a central monitoring system without jeopardising its reasoning model. This module already exists in the architecture of Figure 1 as the deliberative Motivator processes.

As a development of the work described here, distributed versions of the four-layer architecture are being investigated. The extended model includes those aspects described earlier in this paper and other work [9], and is being implemented as a multi-agent society (see Figure 2). In this architecture CRIBB is modeled as a deliberative perception agent and a separate deliberative reasoning agent. Changes to an agent's beliefs are possible through external influence, as in the Castelfranchi model of autonomy [7], using the mechanisms inherent in extended CRIBB (set E of affordances) mediated by ongoing emotional valences of the emotion engine. Exploratory implementations have made use of a simplified version of the motivator structures used in earlier work. Current work looks to formalise the control language based on a computational model of emotion that draws on the Oatley, Frijda and Sherer theories, i.e. a goal-based theory of emotion with modal responses and no basic emotions.

References

- [1] Alterman, R., (2000) Rethinking autonomy, *Minds and Machines*, 10(1):15-30.
- [2] Bartsch, K. & Wellman, H. (1989) Young children's attribution of action to beliefs and desires. *Child Development*, 60, 946-964.
- [3] Buck, R. (1986) The Psychology of Emotion. In J. LeDoux, W. Hirst (Ed.) *Mind and Brain: Dialogues in Cognitive Neuroscience*, Cambridge University Press. Cambridge.
- [4] Brooks, R. A. (1991) Intelligence without representation, *Artificial Intelligence*, 47:139-159.
- [5] Bower, G. (1994) Some Relations Between Emotions and Memory, In P. Ekman and R. Davidson (Ed.) *Nature of Emotion*, Oxford University Press. New York, 303-306.
- [6] Bratman, M.E., Israel, D.J. & Pollack, M.E., (1988) Plans and resource-bounded reasoning, *Computational Intelligence*, 4, 349-355.
- [7] Castelfranchi, C. (1995) Guarantees for autonomy in cognitive agent architectures. In: Wooldridge, M. and N. R. Jennings (Eds), *Intelligent Agents*. Springer-Verlag: 56-70
- [8] Damasio, A. (1994) *Descartes Error: Emotion, Reason and the Human Brain*, New York. Avon Books.
- [9] Davis, D. N. (2001) Control States and Complete Agent Architectures, *Computational Intelligence*, 17(4).
- [10] Duffy, E. (1941) An explanation of 'Emotional' phenomena without the use of the concept 'Emotion', *The Journal of General Psychology*, 25, 283-293
- [11] Elliott, C. (1992) *The Affective Reasoner: A Process Model of Emotions in a Multi-agent System*. PhD thesis, Northwestern University.
- [12] Ferber, J. (1999) *Multi-Agent Systems*, Addison-Wesley.
- [13] Frijda, N. (1987) *The Emotions*, Cambridge University Press, Cambridge

- [14] Gibson, J. (1986) *The Ecological Approach to Visual Perception* Lawrence Erlbaum Associates, New Jersey
- [15] Goleman, D. (1998) *Working with Emotional Intelligence* Bloomsbury, London.
- [16] Izard, C. (1993) Four systems for emotion activation, *Psychological Review*, 100(1), 68-90.
- [17] Matari, M. J. (1997) Studying the role of embodiment in cognition, *Cybernetics and Systems*, 28(6), 457-470.
- [18] Merleau-Ponty, M. (1965) *The Structure of Behaviour*, Methuan, London.
- [19] Newell, A., (1990), *Unified Theories of Cognition*, Harvard University Press.
- [20] Oatley, K. (1992), *Best Laid Schemes*, Cambridge University Press, Cambridge.
- [21] Ortony, A., Clore, G. & Collins, A. (1988) *The Cognitive Structure of Emotions* Cambridge University Press, Cambridge.
- [22] Scherer, K. (1994) Toward a Concept of 'Modal Emotions', In *Nature of Emotion*, P. Ekman, R. Davidson (Eds.) Oxford University Press, New York.
- [23] Simon, H. A. (1979) Motivational and emotional controls of cognition, In: *Models of Thought*, Yale University Press.
- [24] Sloman, A. & Croucher, M. (1987). Why Robots will have emotions. *Proceedings of IJCAI-87*, 197-202.
- [25] Sloman, A. (1999) Architectural requirements for human-like agents both natural and artificial, In *Human Cognition and Social Agent Technology*, K. Dautenhahn (ed.), Benjamins.
- [26] Toda, M. (1962) The Design of a Fungus Eater, *Behavioural Science*, 7, 164-183.
- [27] Wahl, S., Spada, H. (2000) Children's Reasoning about intentions, beliefs and behavior, *Cognitive Science Quarterly*, 1, 5-34.
- [28] Wehrle, T. (1994) New Fungus Eater Experiments. In P. Gaussier & J. D. Nicoud (Eds), *From Perception to Action (400-403)*. Los Alamitos. IEEE Computer Society Press

Emotional Learning as a New Tool for Development of Agent-based Systems

Mehrdad Fatourechi
 Department of Electrical and Computer Engineering,
 University of Tehran, Tehran, Iran
mehrdadf@ece.ubc.ca

Caro Lucas
 Center for Excellence and Intelligent Processing, Department of Control and Electrical Engineering
 University of Tehran, Tehran, Iran
lucas@ipm.ir

Ali Khaki Sedigh
 Department of Electrical Engineering, K.N.Toosi
 University of Tehran, Tehran, Iran
sedigh@eetd.kntu.ac.ir

Keywords: Intelligent control, multivariable systems, emotional learning, neurofuzzy control, agents.

Received: October 21, 2002

A new approach for the control of dynamical systems is presented based on the agent concept. The control system consists of a set of neurofuzzy controllers whose weights are adapted according to emotional signals provided by blocks called emotional critics. Simulation results are provided for the control of dynamical systems with various complexities in order to show the effectiveness of the proposed method.

1 Introduction

It is widely believed that decision making, even in the case of human agents, should be based on full rationality and emotional cues should be suppressed in order to not influence the logic of arriving at proper decisions. The assumption of full rationality, however, has sometimes been abandoned in favor of satisficing or bounded rationality models [1], and in recent years, the positive and important role of emotions have been emphasized not only in psychology, but also in AI and robotics ([2]-[4]). Very briefly, emotional cues can provide an approximate method for selecting good actions when uncertainties and limitations of computational resources render fully rational decision-making based on Bellman-Jacobi recursions impractical. In past researches ([5-9]), a very simple cognitive/emotional state designated as *stress* has been successfully utilized in various control applications. This approach is actually a special case of the popular reinforcement learning technique. However, in this case it is believed that since the continual assessment of the present situation in terms of overall success or failure is no longer simple behaviorist type of conditioning but it is closer to the definition of cognitive state modification and adaptation learning, the designation of emotional learning seems more appropriate. We should emphasize that here emotion merely refers to *stress* cue, and the

use of other, and perhaps higher emotional cues are left for future research.

On the other hand, in recent years, fuzzy logic has been extensively employed in the design of industrial control systems because Fuzzy controllers can work fine in conditions such as severe nonlinearities, time varying parameters or plant uncertainties as supervisory controllers. Also in the last decade, the intelligent control community has paid great attention to the topic of *neurofuzzy* control, combining the decision-making property of fuzzy controllers and learning ability of neural networks. Hence we have chosen a neurofuzzy system as the controller in our methodology.

In the present paper, the idea of applying emotional learning [8] to the dynamic control systems using the agent concepts [10] is addressed. This paper can be considered as the general framework for the previous Single-Input Single Output (SISO) works ([6]-[9]) and NSISO systems ([5]). In general, control scheme consists of a set of agents whose tasks are to provide appropriate control signals for their corresponding system's input. Each agent consists of a neurofuzzy controller and a number of critics, which evaluate the outputs' behavior of the plant and provide the appropriate signals for the tuning of the controllers. Simulation results for the control of the Vander Pol system (*single-agent single-critic approach*), a strongly coupled plant with uncertainty (*multi-agent multi-critic*

approach) and the famous inverted pendulum benchmark (*single-agent multi-critic approach*) are provided to show the effectiveness of the proposed methodology.

The main contribution of the current paper is the introduction of an easily implementable framework that could lead to a controller design with little tuning effort. We have adopted an agent-oriented approach to encapsulate separate concerns in multiobjective and multivariable controller design.

The organization of this paper is as follows: The focus of Section 2 is on the emotional learning and how it can be applied in the control scheme. A brief review of agent concepts and how they could be used in control applications is brought up in section 3. The structure of the proposed controller and its adaptation law are developed in section 4 and in section 5, simulation results are provided to clarify the matter further with the final conclusions to be addressed in section 6.

2 Emotional learning

According to psychological theories, some of the main factors of human beings' learning are emotional elements such as satisfaction and stress. Emotions can be defined as states elicited by instrumental reinforcing stimuli, which if their occurrence, termination or omission is made contingent upon the making of a response, alter the course of future emission of that response [11].

Emotions can be accounted for, as a result of the operation of a number of factors, including the following [11]:

1. The reinforcement contingency (e.g. whether reward or punishment is given, or withheld).
2. The intensity of reinforcement
3. Any environmental stimuli might have a number of different reinforcement associations.
4. Emotions elicited by stimuli associated with different reinforcers will be different.

It should also be mentioned that in this paper, emotion merely refers to *stress* cue and other (and perhaps higher) emotions are not considered here. In our proposed approach, which in a way is a cognitive restatement of reinforcement learning in a more complex continual case (where reinforcement is also no longer a binary signal), there exists an element in the control system called *emotional critic* whose task is to assess the present situation which has resulted from the applied control action in terms of satisfactory achievement of the control goals and to provide the so called emotional signal (the stress). The controller should modify its characteristics so that the critic's stress is decreased. This is the primary goal of the proposed control scheme, which is similar to the learning process in the real world because in the real world, we also search for a way to lower our stress with respect to our environment ([12-13]).

As seen, emotional learning is very close to reinforcement learning, but the main difference between

them is that in the former the reinforcement signal is an analog emotional cue that represents the cognitive assessment of future costs given the present state. So here the system does not wait for a total failure to occur before it starts learning. Instead, it continues its learning process at the same time as it applies its control action. The resulting analog reinforcement signal constitutes the stress cue, which has been interpreted as cognitive/emotional state.

In the next section, we'll discuss the concept of agent-based systems that will be used as the framework of our proposed control system.

3 Agent Concept and Multi-Agent Systems

The main problem of dealing with multivariable control systems is dealing with cross-coupled components between different inputs and outputs. In other words, changing an input not only makes some changes in the corresponding output, but also influences other outputs as well. As it will be discussed in section 4, emotional learning provides a simple useful tool in dealing with such unwanted effects. The concept of this method can be easily developed within the framework of multi-agent systems. In order to do that, in this section we briefly address *agents* and *multi-agent* systems.

Here we define an agent as referring to a component of software/hardware, which is capable of accomplishing tasks on behalf of its user. By reviewing Jennings and Wooldridge's work [14], we define an agent to be any kind of object or process that exhibits autonomy, is either reactive or deliberative, has social ability, and can reason, plan, learn, and- or adapt its behavior in response to new situations.

Multi-agent systems (MASs) are systems where there is no central control: the agents receive their inputs from the system (and possibly from other agents as well) and use these inputs to apply the appropriate actions. The global behavior of MAS depends on the local behavior of each agent and the interactions between them [15]. The most important reason to use MAS when designing a system is that some domains require it. Other aspects include: Parallelism, Robustness, Scalability and Simple Design.

Based on these concepts, we have proposed an emotion-based approach for the control of dynamic systems, which will be discussed in the next section

4. An Emotion-based Approach to the Control of Dynamic Systems using Agent Concept

In this section we design an intelligent controller based on the concepts considered in the previous sections. Fig. 1 shows the proposed agent's components and their relation with each other based on the idea presented in [16]. As it can be seen, the agent is

composed of four components. It perceives the states of the system through its *sensors* and also receives some information provided by other agents, then influences the system by providing a control signal through its *actuator*. The *critics* assess the behavior of the control system (i.e. criticize it) and provide the emotional signals for the controller. According to these emotional signals, the *controller* produces the control signal with the help of the *Learning element*, which is *adaptive emotional learning*. Inputs of this learning element are the emotional signals provided by both the agent's critics and other critics as well.

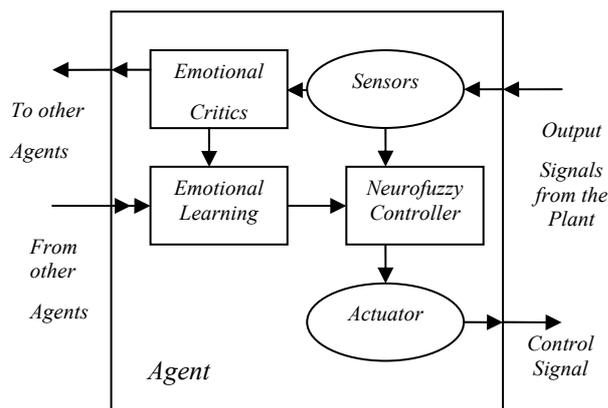


Fig 1. Structure of an agent in the proposed methodology

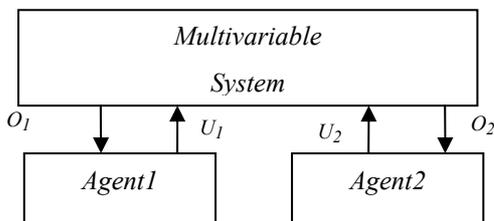


Fig 2. Multi-agent based approach to multivariable control

The number of the agents assigned here is determined based on the number of the inputs of the system. The number of the outputs of the system is effective in determining the number/structure of the system's critics, which their role is to assess the status of the outputs. (See Fig.2 for the schematic of the presented approach when applied to a two input – two output control system where U_1 and U_2 denote the control signals and O_1 and O_2 are the outputs of the system).

We now develop the controller structure for the multivariable systems, in general. From these calculations, derivation of the special case of SISO systems is straightforward.

In the general case of multivariable systems, each agent consists of a neurofuzzy controller. All of the neurofuzzy controllers have identical structures; each one has four layers. The first layer's task is the assignment of inputs' scaling factors in order to map them to the range of $[-1, +1]$ (the inputs are chosen as the error and the change of the error in the response of the corresponding output). In the Second layer, the fuzzification is performed assigning five labels for each input. For decision-making, max-product law is used in layer 3. Finally, in the last layer, the crisp

output is calculated using Takagi- Sugeno formula [17],

$$y_i = \frac{\sum_{l=1}^p u_{il} (a_{il}x_{i1} + b_{il}x_{i2} + c_{il})}{\sum_{l=1}^p u_{il}} \quad (\text{for } i = 1, 2, \dots, n)$$

(1)

Where x_{i1} and x_{i2} are inputs to the controller (the error and the change of error of the corresponding output), $i, n, u_{il}, p,$ and y_i are the index of the controller, number of controllers, l 'th input of the last layer, number of rules in the third layer and output of the controller, respectively and a_{il} 's, b_{il} 's and c_{il} 's are parameters to be determined via learning.

For each output, a critic is assigned whose task is to assess the control situation of the outputs and to provide the appropriate emotional signal. The role of the critics is very crucial here because eliminating unwanted cross-coupled effects of the multivariable control systems is very much dependent on the correct operation of these critics. Here, all the critics have the same structure as of a PD fuzzy controller with five labels for each input and seven labels for the output. Inputs of each critic are error of the corresponding output and its derivative and the output is the corresponding emotional signal. Deduction is performed by max-product law and for defuzzification, the centroid law is used.

The emotional signals provided by these critics contribute collaboratively for updating output layer's learning parameters of each controller, thus the cross-coupled nature of multivariable systems is considered in the critic and not in the controller itself. The aim of the control system is to minimize of the sum of squared emotional signals. Accordingly, first we describe the error function E as follows,

$$E = \sum_{j=1}^m K_j (\frac{1}{2} r_j^2)$$

(2)

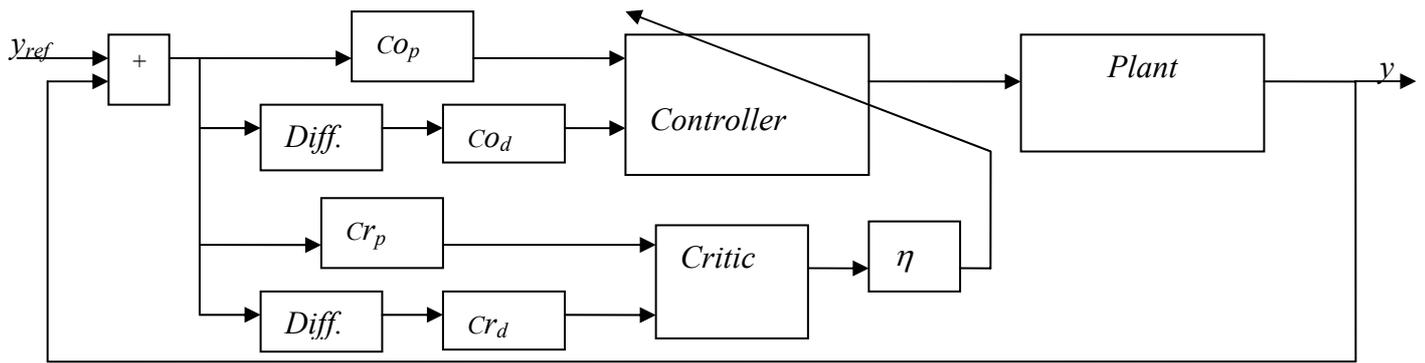


Fig.3. The control loop in the case of SISO systems

Where r_j is the emotional signal produced as the output of j 's critic, K_j is the corresponding output weight and m is the total number of outputs (for the special case of SISO systems, $K_j=1$ and $m=1$)

For the adjustment of controllers' weights the steepest descent method is used,

$$\Delta\omega_i = -\eta_i \frac{\partial E}{\partial \omega_i} \quad (i=1,2,\dots,n)$$

(3)

Where η_i is the *learning rate* of the corresponding neurofuzzy controller and n is the total number of controllers.



In order to calculate the RHS of (3), the chain rule is used,

$$\frac{\partial E}{\partial \omega_i} = \sum_{j=1}^m \frac{\partial E}{\partial r_j} \cdot \frac{\partial r_j}{\partial y_j} \cdot \frac{\partial y_j}{\partial u_i} \cdot \frac{\partial u_i}{\partial \omega_i}$$

$$(i=1\dots n \quad \text{and} \quad j=1, 2\dots m)$$

(4)

From (2), we have,

$$\frac{\partial E}{\partial r_j} = K_j \cdot r_j \quad (j=1,2,\dots,m)$$

(5)

Also,

$$\frac{\partial y_j}{\partial u_i} = J_{ji} \quad (i=1,2,\dots,n \quad \text{and} \quad j=1,2,\dots,m)$$

(6)

Where J_{ji} is the element located at the i th column and j th row of the Jacobean matrix.

Taking

$$e_j = y_{refj} - y_j \quad j=1,2,\dots, m$$

(7)

Where e_j is the error produced in the tracking of j th output and y_{refj} is the reference input (in case number of outputs is greater than the number of inputs, some of y_{refj} s are taken as zero as it will be cleared in the next section by the inverted pendulum example). Now we have,

$$\frac{\partial r_j}{\partial y_j} = -\frac{\partial r_j}{\partial e_j}$$

(8)

Since with the incrimination of error, r (the stress of the critic) will also be incremented and on the other

hand, on-line calculation of $\frac{\partial r_j}{\partial e_j}$ is accompanied with

measurement errors, thus producing unreliable results, only the sign of it (+1) is used in our calculations.

From (2) to (8), $\Delta\omega_i$ will be calculated as follows,

$$\Delta\omega_i = \eta_i \sum_{j=1}^m K_j \cdot r_j \cdot J_{ji} \cdot \frac{\partial u_i}{\partial \omega_i}$$

$$Co_p = 1, Co_d = 2, Cr_p = 1, Cr_d = 2, \eta = 20$$

$$(i=1,2,\dots,n \quad \text{and} \quad j=1,2,\dots,m)$$

(9)

Equation (9) is used for updating the learning parameters a_{il} 's, b_{il} 's and c_{il} 's in (1), which is straightforward.

In the next section, we'll apply the proposed method to several SISO and NSISO plants with different properties in order to see the performance of the proposed control methodology in practice.

5. Simulation Results

In this section, the proposed method is applied to control three dynamical systems. The first one is the highly nonlinear SISO Vander Pol system where a single-agent approach is used. In the second one, the controller is applied to a multivariable linear control plant with different conditions so that its robustness in the presence of parameter uncertainties is shown. This example is concerned with systems with equal number of inputs and outputs. In the last example, we apply our controller to the famous inverted pendulum benchmark, which is a SIMO (single-input multi-output) nonlinear non-minimum phase system.

Example 1: Vander Pol Equation

Our first example discusses the control of the Vander Pol system, which is considered as a highly nonlinear SISO system. We use a single-agent single-critic approach here. The equations governing this system are as follows:

$$\begin{aligned} \ddot{x} + (1 - x^2)\dot{x} + x &= u \\ y &= x \end{aligned} \tag{10}$$

In (10), u is the input, x is the single state equation and y is the output if the system.

The block diagram of the control system is shown in Fig.3. The Input scaling factors and the learning rate of the control system are chosen as follows:

(11)

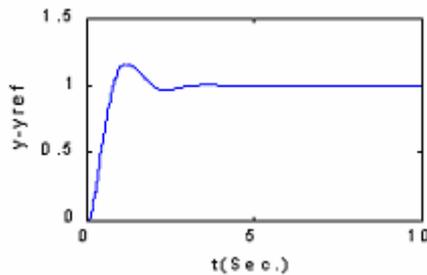


Fig 4. Step response of the Example 1

Step response of the control system is shown in Fig.4. The result shows the power of the proposed algorithm in the control of this nonlinear SISO system.

Example 2: Control of a plant with different conditions

In our second example, the problem of handling a multivariable plant with uncertainties is investigated. The plant has the following transfer function [18]:

$$P(s) = \begin{bmatrix} \frac{k_{11}}{1 + sA_{11}} & \frac{k_{12}}{1 + sA_{12}} \\ \frac{k_{21}}{1 + sA_{21}} & \frac{k_{22}}{1 + sA_{22}} \end{bmatrix} \tag{12}$$

It has a total of nine plant conditions as given in Table 1.

Our goal is to achieve the desired step response while output decomposition is maintained. A major problem encountered here is the tuning of control system’s coefficients in order to provide an acceptable step response. It’s a time consuming task and there exists the possibility that the desired step response may never be achieved.

Our Experience with this control structure shows that when the change made in the input of the system is *smooth* (i.e. there are no sudden changes like applying a step response in the input but instead smoother inputs

like sinusoids are applied) the control system acts very well. The reason is obvious: it takes much more time for the neural networks’ weights to adjust when the input of the system changes suddenly (let’s call it harsh input) compared to a situation where a much smoother input is applied. This problem is more evident in the case of multivariable systems when more than one controller’s weights should be adjusted. Hence, when applying a harsh input to a system, we’ll change it to a smooth one by pre-filtering it and obtaining a *smooth* (filtered) input for the system instead of *harsh* (unfiltered) one. The pre-filters’ specifications are determined by *the properties of the desired step response*. The results of the simulations here show that this approach, although it’s simple, is very efficient in different control situations.

In this example, suppose that it is desired that both

Table 1: Nine plant conditions of example 2

Plant Condition	K ₁₁	K ₂₂	K ₁₂	K ₂₁	A ₁₁	A ₂₂	A ₁₂	A ₂₁
1	1	2	0.5	1	1	2	2	3
2	1	2	0.5	1	0.5	1	1	2
3	1	2	0.5	1	0.2	0.4	0.5	1
4	4	5	1	2	1	2	2	3
5	4	5	1	2	0.5	1	1	2
6	5	5	1	2	0.2	0.4	0.5	1
7	10	8	2	4	1	2	2	3
8	10	8	2	4	0.5	1	1	2
9	10	8	2	4	0.2	0.4	0.5	1

Table 2: Additional plant conditions for the multivariable system in example 2

Plant Condition	K ₁₁	K ₂₂	K ₁₂	K ₂₁	A ₁₁	A ₂₂	A ₁₂	A ₂₁
10	1	2	1	1	1	2	2	3
11	1	1	1	1	1	2	2	3
12	1	1	0.5	1	1	2	2	3
13	1	2	2.5	1	1	2	2	3
14	10	8	10	8	0.2	0.4	0.5	1
15	10	8	10	8	0.7	0.4	0.5	1

outputs have no overshoot and a rise time not more than 1 second. Accordingly, based on a rough measure the transfer functions of pre-filters are the same and are chosen as follows (note that achieving more complicated inputs requires more complicated pre-filter design technics which is not the topic of our discussion here):

$$H_1(s) = \frac{16}{s^2 + 8s + 16} \tag{13}$$

Results of simulations are shown in Fig.5 for a step response at $t=0$ at the first input and another step response at $t=3$ at the second input (Since all the conditions nearly produced the same results, the results of simulations for three selected conditions are chosen

to be plotted). As it is clearly obvious, the change of plant conditions has little or almost no effect in the step responses of the system, i.e., system shows great robustness in the presence of uncertainties. Comparing the results with those obtained by classical methods such as the one in [18], shows the superiority of the proposed algorithm.

Although we've achieved good step responses and great robustness, but we should take another important aspect into notice and that's the interaction in this system, which is not high. Interaction is the major drawback in the design of multivariable systems because it introduces unwanted effects from different inputs in the outputs of the system. The more is the interaction in the systems, the more complex the control approach will be.

In order to show that our proposed controller can tolerate bigger parameter changes, which yield situations with high interaction, we added 6 more conditions to the previous ones (Table 2). The results of applying the controller are shown in Fig.6 for two selected conditions. As we can see, our method also shows great robustness to parameter uncertainties in the presence of high interactions.

Example 3: An Inverted Pendulum:

The problem of balancing an inverted pendulum on a moving cart is a good example of a challenging multivariable situation, due to its highly nonlinear equations, non-minimum phase characteristics and the problem of handling two outputs with only one control input [19] (the position of cart is sometimes ignored by the researchers [20]). Here, the dynamics of the inverted pendulum are characterized by four variables: θ (angle of the pole with respect to the vertical axis), $\dot{\theta}$ (angular velocity of the pole), z (position of the cart on the track), and \dot{z} (velocity of the cart). The behavior of these state variables is governed by the following two second-order differential equations [17]:

$$\ddot{\theta} = \frac{g * \sin\theta + \cos\theta * \left(\frac{-F - m * l * \dot{\theta}^2 * \sin\theta}{m_c + m} \right)}{l * \left(\frac{4}{3} - \frac{m * \cos^2\theta}{m_c + m} \right)} \quad (14)$$

$$\ddot{z} = \frac{F + m * l * (\dot{\theta}^2 * \sin\theta - \ddot{\theta} * \cos\theta)}{m_c + m} \quad (15)$$

Where g (acceleration due to gravity) is $9.8 \frac{m}{s^2}$, m_c (mass of cart) is 1.0 kg, l (half-length of pole) is 0.5 m, and F is the applied force in Newton. Our control goal is to balance the cart, yet keep the z not further than 2.5 meters from its original position. We use a single agent here, which provides the force F to the system and applies two emotional critics to assess the output. The first one criticizes the situation of the pole and the second one does the same for the cart's velocity. Both

critics are satisfied when inputs to them are zero (i.e. the pendulum is balanced and the cart has no velocity). The results of simulation for initial condition $\theta_0 = 10$ deg. are presented in Fig. 7. They show that after nearly six seconds the pole is balanced and the cart is stopped successfully around 1.4 meters from the original position.

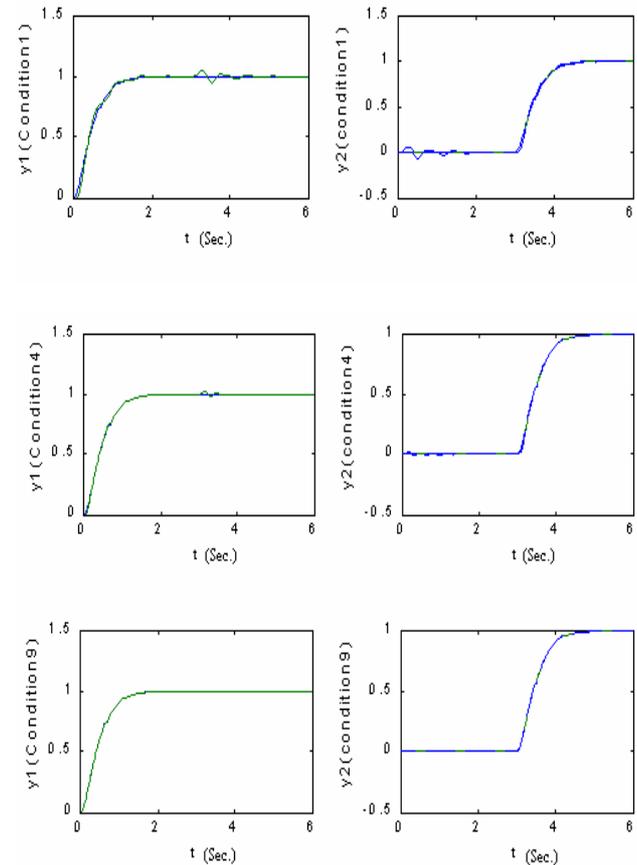


Fig.5. Simulation results of example 2. 5(a) condition 1; 5(b) condition (4); 5(c) condition (9)

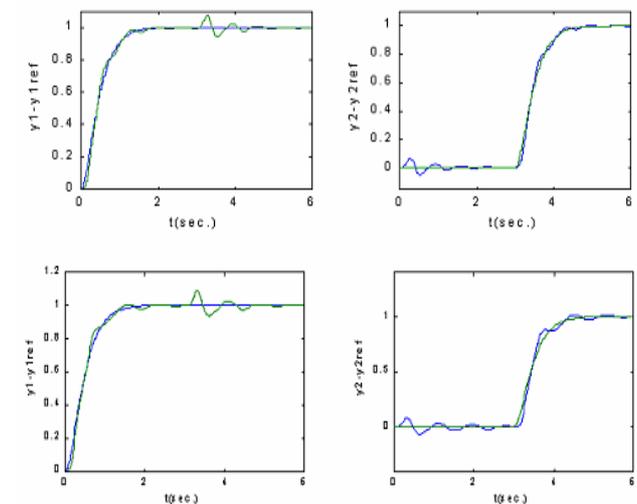


Fig.6. Simulation results of example 2. 6(a) condition 10; 6(b) condition (12).

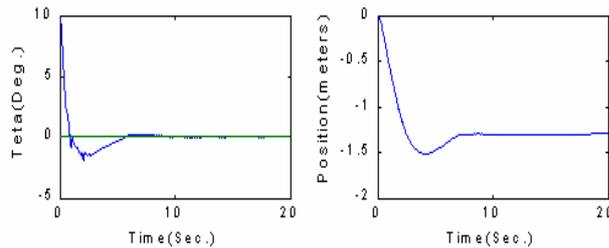


Fig. 7. Responses of variables of example 1 (from left to right: pole's angle and cart's position)

6. Discussions and Conclusions

In this section we discuss the general properties of the proposed framework and we'll summarize the work that has been done in this paper.

6.1. The role of emotional signals in the proposed control scheme

The proposed methodology is based on continuous emotional (stress) signals, which can be considered as performance measures of particular parts of the control system, which are of interest. In this paper, the parts that we're interested in, are the outputs of the control system and the cross-coupled components in the multivariable systems. In each part, the nearer we got to our predefined target, the less is the corresponding emotional signal and vice versa. With this simple approach, we can easily include any parts of the plant on which we want to have a control on it, in our framework. For example, for excluding the effect of cross-coupled components in multivariable systems, we'll assign a critic for each component. This critic would judge whether the control system has counterbalanced the cross-coupled effects or not. Based on the success of the controller in dealing with interaction, emotional signals are produced by the critics who in their turn would tune the parameters of the neurofuzzy controller so that the stress of the critics would be decreased.

The same situation also holds for the inverted pendulum example in the previous section. The main variable that is of interest is the position of the pendulum regards to the vertical axis, but at the same time the position of the cart is also of interest here as the secondary control variable. Hence the inputs of the neurofuzzy controller are the error and the derivative of the angle of the pole with respect to the vertical axis but the weights of the controller are tuned based on the outputs of two critics; the first one criticizes the position of the pole and the second one does the same for the position of the cart. Both critics produce continuous signals until their inputs are zero, i.e. when the predefined targets are achieved.

Next we'll discuss how our framework is related to agents and multi-agent systems and then we'll discuss

the advantages and the shortcomings of the current method briefly which will be followed by description of future works.

6.2 Relationship between the Proposed Framework and Agent-Based Systems

In this paper a major consideration has been distributing control concerns via agents. Each agent is used for representation of a control concern. In this paper we have used the notion of agency only in conceptual sense and no effort has been made towards utilization of agent-oriented technologies like ACL's, platforms, wrappers, etc. However, those technologies can be of benefit in future, more complex applications. The main agent property in our paper is autonomy. Our agents can be both interpreted as deliberative as well as reactive (since emotion is a mental state, but is also very close to the concept of reinforcement), and learning, reasoning, and adaptation is central to our proposed controller. Other benefits of agent orientation can also be seen to be applicable.

6.3 Conclusions and Future Works

In this paper, the emotional learning based intelligent control scheme was applied to dynamic plants. Also the performance of the proposed algorithm was investigated by several benchmark examples. The main contribution of the proposed generalization is to provide the easy to implement emotional learning technique for dealing with dynamic (especially multivariable) control systems where the use of other control methodologies (specially intelligent control methods) are sometimes problematic ([21]). Simplicity and tolerance for uncertainties and nonlinearities is what is gained by its use. This is shown in various contributions for SISO and NSISO systems in this paper.

On the negative side, it should be pointed out that only a very simple learning algorithm has been used throughout this paper. Although this stresses the simplicity and generality of the proposed technique, more complex learning algorithms involving time credit assignments [22] and temporal difference [23] or similar methods might be called for when processes involve unknown delays. Also as the number of the inputs and outputs of the system grows, the tuning of the control parameters becomes a tiresome task. A continuous genetic algorithm based optimization method is under development to find the optimal selection of the tuning parameters of the overall control system automatically. Having this done, generalization to systems with multiple numbers of inputs and outputs (more than two) can be realized efficiently.

Other works include the application of multiple critics in a SISO plant in order to achieve multiple objectives ([6-7]). For example in [6], two objectives (good tracking and low control costs) are considered simultaneously. This reference shows the difference between our approach and supervised learning in a

clearer manner, because it shows that our proposed methodology can perform well not only perform well in the case of cheap control, but also where control action also involves costs. Also implementing such control system for a switched reluctance motor (as a practical system) is under investigation. Again, agent orientation can underline the fact that each objective can be considered a separate concern delegated to an agent

Our future works include changing the structure of the controller so that it could be applied to processes with unknown delays, considering more emotions in our control structure, optimizing the structure of the controller (for example using genetic algorithm for optimum selection of the membership functions of both the controllers and the critics), and finally considering more complex cues in our learning process.

Acknowledgement

The authors would like to thank the two anonymous referees for their valuable comments.

References

- [1] A. H. Simon and Associates (1987), Decision making and problem solving, *Interfaces*, no.17.
- [2] C. Balkenius and J. Moren (2000), A Computational Model of Context Processing, *6th International Conference on Simulation of Adaptive Behavior*, Cambridge.
- [3] M.El-Nasr, T. Loerger, and J.Yen (1999), Peteei: A Pet with Evolving Emotional Intelligence, *Autonomous Agents99*, pp.9-15.
- [4] J. Velasquez (1998), a Computational Framework for Emotion-Based Control, *the Grounding Emotions in Adaptive Systems Workshop SAB'98*, and Zurich, Switzerland.
- [5] M. Fatourechchi, C. Lucas, and A. Khaki Sedigh (2001), An Agent-based Approach to Multivariable Control, *IASTED International Conference on Artificial Intelligence and Applications*, Marbella, Spain, pp.376-381.
- [6] M. Fatourechchi, C. Lucas and A. Khaki Sedigh (2001), Reducing Control Effort by means of Emotional Learning, *9th Iranian Conference on Electrical Engineering (ICEE2001)*, Tehran, Iran, pp.41-1 to 41-8.
- [7] M. Fatourechchi, C. Lucas and A. Khaki Sedigh (2001), Reduction of Maximum Overshoot by means of Emotional Learning, *6th Annual CSI Computer Conference*, Isfahan, Iran, pp.460-467.
- [8] C. Lucas and S.A. Jazbi (1998), Intelligent motion control of electric motors with evaluative feedback, *Cigre98*, Cigre, France, 11-104, pp.1-6.
- [9] C. Lucas, S.A. Jazbi, M. Fatourechchi, M. Farshad (2000), Cognitive action selection with neurocontrollers, *Third Iran-Armenia Workshop on Neural Networks*, Yerevan, Armenia.
- [10] P. Maes (ed.) (1991), Designing autonomous agents: theory and practice from biology to engineering and back, *The MIT press*, London.
- [11] E. D. Rolls (1998), *the Brain and Emotion*, Oxford University Press.
- [12] K.M. Galloti (1999), *Cognitive psychology in and out of laboratory (2nd ed.)*, Brooks/Cole, Pacific Grove, CA.
- [13] R.W. Kentridge and J.P. Aggleton (1990), Emotion: Sensory representations, reinforcement and the temporal lobe, *Cognition and Emotion 4*, pp.191-208.
- [14] M. Wooldridge and N. Jennings (1995), Intelligent Agents: Theory and Practice, *The Knowledge Engineering Review*, 10 (2), pp.115-152.
- [15] M. Wooldridge (1999), Intelligent agents, in G. Weiss (Ed.), *Multi agent Systems: A modern approach to Distributed Artificial Intelligence*, MIT Press, London, pp.27-77.
- [16] S. Russel and P. Norwig (1995), *a modern approach to artificial intelligence*, Prentice-Hall, Englewood Cliffs.
- [17] T. Takagi and M. Sugeno (1983), Derivation of fuzzy control rules from human operator's control actions, *IFAC Symp.on Fuzzy Information, Knowledge Representation and Decision Analysis*, pp.55-60.
- [18] C.C. Cheng, Y.K. Liao, T.S. Wanq (1997), Quantitative design of uncertain multivariable control system with an inner-feedback loop, *IEE Proceedings on Control Theory Applications*, no.144, pp.195-201.
- [19] R. H. Cannon (1967), *Dynamics of Physical Systems*, McGraw-Hill, New York.
- [20] J. S. Jang (1992), Self-learning fuzzy controllers based on temporal back propagation, *IEEE Transactions on Neural Networks*, 3(5), pp. 714-723.
- [21] P.G. Lee, K.K. Lee and G.J. Jeon (1995), An index of applicability for the decomposition method of multivariable fuzzy systems, *IEEE Transactions on Fuzzy Systems*, no. 3, pp. 364-369.
- [22] R. S. Sutton, A. G. Barto (1987), A Temporal – Difference Model of Classical Conditioning, *9th Annual Conference on Cognitive Science*, New Jersey, pp.355-378.
- [23] R. S. Sutton (1988), Learning to Predict by the Method of Temporal Differences, *Machine Learning*, no.3, pp.9-44.

Learning Behavior-selection in a Multi-goal Robot Task

Sandra Clara Gadanho and Luis Custódio
 Institute of Systems and Robotics, IST, Lisbon, Portugal
 sandra@isr.ist.utl.pt <http://www.isr.ist.utl.pt/~sandra/>
 lmmc@isr.ist.utl.pt <http://islab.isr.ist.utl.pt/>

Keywords: learning, emotions, autonomous robots

Received: October 30, 2002

The purpose of the work reported here is the development of an autonomous robot controller which learns to perform a multi-goal and multi-step task when faced with real world problems such as continuous time and space, noisy sensors and unreliable actuators.

In order to make the learning task feasible, the agent does not have to learn its action abilities from scratch, but relies on a small set of simple hand-designed behaviors. Experience has shown that these low-level behaviors can be either easily designed or learned but that the coordination of these behaviors is not trivial. To solve the problem at hand, a dual-system architecture is proposed in which a traditional reinforcement learning adaptive system is complemented with a goal system responsible for both reinforcement and behavior switching.

This goal system is inspired by emotions, which take a functional role on this work, and are evaluated in terms of their engineering benefits, i.e. in terms of their competitiveness when compared with alternative approaches. Experiments reported carefully evaluate the goal system and determine its requirements.

1 Introduction

In order to master a task, a robot controller may use reinforcement-learning techniques (e.g., 26; 14, for surveys on RL) to learn the appropriate selection of simple actions. For more complex tasks, skill decomposition is usually advisable as it can significantly reduce the learning time, or even make the task feasible. Skill decomposition usually consists of learning some predefined behaviors in a first phase and then finding the high-level coordination of these behaviors. Although the behaviors themselves are often learned successfully, behavior coordination is much more difficult and is usually hard-wired to some extent in other robotics applications (17; 15; 19).

While learning the low-level behaviors consists of deciding on a simple reactive action on a step-by-step basis, when learning behavior selection apart from deciding which behavior to select, the controller must also decide when to switch and reinforce behaviors. There are various reasons why a behavior may need to be interrupted: it has reached its goal; it has become inappropriate, due to changes in the environment; or it is not able to succeed in its goal.

In practice, the duration of a behavior must be long enough to allow it to manifest itself, and short enough so that it does not become inappropriate due to changing circumstances.

The problem of deciding when to change behavior is not an issue in traditional reinforcement learning problems, because these usually consist of grid worlds divided in cells which represent states. In those worlds, the execution of

a single discrete action is responsible for a state transition since it moves the agent to one of the cells in the neighborhood of the cell where the agent is located. In a continuous world, the determination of a state transition is not clear. In robotics, agent states change asynchronously in response to internal and external events, and actions take variable amounts of time to execute (19). As a solution to this problem, some researchers extend the duration of the current action according to some domain-dependent conditions of goal achievement or applicability of the action. Others will interrupt the action when there is a change in the input state (22; 2). However, this may not be a very straightforward solution when the robot is equipped with multiple continuous sensors that are vulnerable to noise. (18) go a step further, and auto-regulate the degree of discrimination of new events by attempting to maintain a constant attentional effort.

Inspired by literature on emotions, previous work has shown that reinforcement and behavior-switching can easily be addressed together by an emotion model (13; 12). The justification for the use of emotions is that, in nature, emotions are usually associated with either pleasant or unpleasant feelings that can act as reinforcement (e.g., 27; 1; 4) and frequently pointed to as a source of interruption of behavior (25; 24).

The task used in the current work has been solved with success by that emotional system as the goal system. The goal system proposed here represents an abstraction of that system which has similar performance. The current goal system does not model emotions explicitly although it is inspired on them, but instead tries to identify which are

the properties the goal system must have in order to work correctly. This goal system is based on a set of homeostatic variables which it attempts to maintain within certain bounds.

The goal system's required properties are identified within a complex task with multiple goals. Apart from dealing with real-world problems, the task developed has several features which pose extra difficulties to the learning algorithm:

- it has multiple goals which may conflict with each other;
- there are situations in which the agent needs to temporarily overlook one goal in order to successfully accomplish another;
- the agent has short-term and long-term goals;
- a sequence of behaviors may be required to accomplish a certain goal;
- the behaviors are unreliable: they may fail their goal or they may lead the agent to undesirable situations;
- the behaviors' appropriate durations are undetermined, they depend on the environment and on their success.

In the next section, this task is described in detail. This will be followed by a description of the proposed architecture in terms of its goal system and adaptive system. Finally, the experiments made are described, the proposed architecture is compared with related work and conclusions reached are presented. To conclude future work on the architecture is discussed.

2 The Robot Task

The experiments reported here evaluated controllers in a survival task that consists of maintaining adequate energy levels in a simulated environment with obstacles and energy sources which are associated with lights the agent can sense when nearby. The agent has basically three goals: to maintain its energy, avoid collisions and move around in its environment.

To gain energy from an energy source, the robot has to bump into it. This will make energy available for a short period of time. It is important that the agent is able to discriminate the existence of available energy, because the agent can only get energy during this period. This energy is obtained by receiving high values of light in its rear light sensors, which means that the robot must quickly turn its back to the energy source as soon as it senses that energy is available. To receive further energy, the robot has to restart the whole process by hitting the light again so that a new time window of released energy is started.

An energy source can only release energy a few times before it is exhausted. In time, the energy source will recover its ability to provide energy again, but meanwhile the robot is forced to search for other sources of energy in order to survive. The robot cannot be successful by relying on a single energy source, *i.e.* the time it takes for new energy to be available in a single energy source is longer than the time it takes for the robot to waste that energy. When an energy source has no energy, the light associated with it is turned off and it becomes a simple obstacle for the robot.

The extraction of energy was complicated, as described above, in order to make the learning task harder by requiring the agent to learn sequences of behaviors. Moreover, it requires the agent to temporarily suppress its goal of avoiding obstacles in the process of acquiring energy.

3 The Robot Controller

The proposed architecture — see Figure 1 — is composed by two major systems: the goal system and the adaptive system. The goal system evaluates the performance of the adaptive system in terms of the state of its homeostatic variables and determines when a behavior should be interrupted. The adaptive system learns which behavior to select using reinforcement-learning techniques which rely on neural-networks to store the utility values. The two systems are described in detail in the following.

3.1 Goal System

In an autonomous agent, the goal system complements a traditional reinforcement-learning adaptive system in that it determines how good the adaptive system is doing, or more specifically, the reinforcement it is entitled to at each step. In the current work the goal system is also responsible for determining when behavior switching should occur.

Previous work (13) addressed the problem of the goal system by using an emotional model. A mixture of perceptual values and internal values were used in the calculation of a single multi-dimensional emotional state. This state in turn was used to determine the reinforcement at each time step and significant differences in its value were considered to be relevant events used to trigger the behavior selection mechanism.

In the current work, this system has been modified to emphasize the multiple goal nature of the problem at hand and identify and isolate the different aspects of the agent-environment interaction that need to be taken into consideration when assessing the agent's overall goal state. The goals are explicitly identified and associated with homeostatic variables. These homeostatic variables are associated with three different states: target, recovery and danger. The state of each variable depends on its continuous value which is grouped into four qualitative categories: optimal, acceptable, deficient and dangerous. See details of state transition in Figure 2 and an example of the categorization of the continuous values of an homeostatic variable

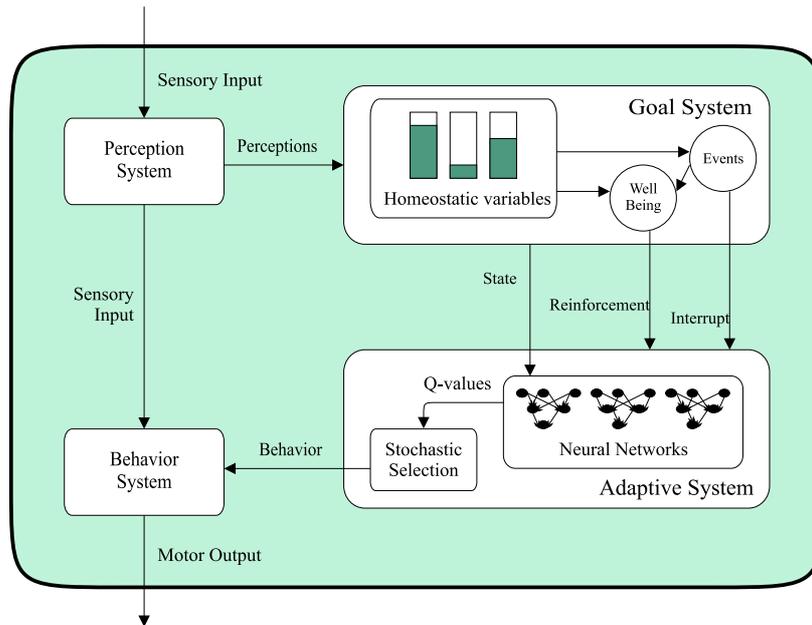


Figure 1: The robot controller.

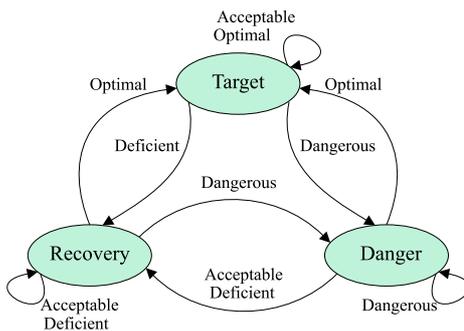


Figure 2: The state transitions of an homeostatic variable dependent on its value.

in Figure 3. The variable remains in its target state as long as its values are optimal or acceptable, but it only returns to its target state once its values are optimal again. This state transition is akin to that of a thermostat in that a greater deviation from the target values is required to change from a target state into a recovery state than the inverse transition. The danger state is associated with dangerous values and can be related with urgency of recovery.

To reflect the current hedonic state of the agent a well-being value was constructed from the above. This value depends primarily on the values of the homeostatic variables. When a variable is in the target state it has a positive influence on the well-being, otherwise it has a negative influence which is proportional to its deviation from target values.

In order to have the system working correctly two other influences on well-being were also required:

State change — when a homeostatic variable changes from a state to another the well-being is influenced positively if the change is towards a better state and negatively otherwise;

Prediction of state change — when some perceptual cue predicts the state change of a homeostatic variable, the influence is similar to the above, but lower in value and dependent on the accuracy of the prediction and on how soon the state change is expected.

In particular, if a transition to the target state involves a sequence of steps then a positive prediction may be made any time a step is accomplished. The intensity of the prediction increases as the number of steps to finish the sequence is reduced.

Predictions are always associated with individual homeostatic variables and are only made if the corresponding variable value is not optimal.

The two goal events just described were modeled after emotions, in the sense that they result from the detection of significant changes in the agent’s internal state or predictions of such changes.

In the same way that emotions are associated with feelings of ‘pleasure’ or ‘suffering’ depending on whether this change is for the better or not, these goal events influence the well-being value such that the information of how good the event is is conveyed to the agent through the reinforcement. One may distinguish between the emotion of happiness when a goal is achieved (or predicted to be achieved) and the emotion of sadness when a goal state is lost (or about to be lost).

The primary influence of the homeostatic variables, on the other hand, is modeled after the natural background

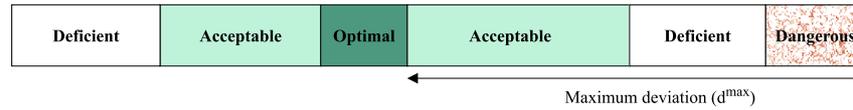


Figure 3: An example of the categorization of the possible continuous values of a homeostatic variable.

emotions which reflect the overall state of the agent in terms of maintaining homeostasis (7).

The goal events are also responsible for triggering the adaptive system for a new behavior selection, which is also often associated with emotions.

The calculation of the well-being value (wb) is presented in Equations 1, 2 and 3. This depends on the domain-dependent set of homeostatic variables (\mathcal{H}) in different ways: their state, their transitions and predictions. These different influences are weighted by their respective coefficients (c_s , $c_t(h)$ and c_p) presented in Table 1. The weights w_h are constants which denote the relative importance of each homeostatic variable h and their value should lie between -1 and 1. The value of the well-being is normalized by a constant value (wb^{\max}), calculated by Equation 3, so that it is never above 1.0 or below -1.0. This depends on the maximum absolute value (c_t^{\max}) of the transition coefficient which is 1.0 (see Table 1).

$$wb = \sum_{h \in \mathcal{H}} \frac{(c_s r_s(h) + c_t(h) + c_p r_p(h)) w_h}{wb^{\max}} \quad (1)$$

$$r_s(h) = \begin{cases} 1 & \text{if } h \text{ is in target} \\ -d(h)/d_h^{\max} & \text{otherwise} \end{cases} \quad (2)$$

$$wb^{\max} = (c_s + c_t^{\max} + c_p) \sum_{h \in \mathcal{H}} w_h \quad (3)$$

The influence of the state of an homeostatic variable on well-being is expressed by $r_s(h)$ described in Equation 2. This value is 1 if the homeostatic variable is in its target state. Otherwise, it depends on the normalized deviation from optimal values, *i.e.* the shortest distance ($d(h)$) of the current value to a optimal value normalized by the maximum possible distance (d_h^{\max} , see example in Figure 3) of any value of this homeostatic variable to a target value. This ensures that the normalized deviation is always between 0 and 1.

The values of predictions ($r_p(h)$) depends on the strength of the current prediction and vary between -1 (for predictions of no desirable changes in the homeostatic variable h) and 1 (for predictions of desirable changes). If there is no prediction then $r_p(h) = 0$.

The values of both w_h and $r_p(h)$ are domain-dependent and are presented later.

For the task at hand, three homeostatic variables were identified:

Energy — is the battery energy level of the agent and reflects the goal of maintaining its energy;

Welfare — maintains the goal of avoiding collisions — this variable is in its target state when the agent is not in a collision situation;

Activity — ensures that the agent keeps moving — if the robot keeps still its value slowly decreases until eventually its target state is not maintained.

These variables are directly associated with the robot goals mentioned previously. Their associated weights (w_h) was 0.5 for Energy, 0.3 for Welfare and 0.2 for Activity. These weights translate the relative importance of each one of the goals. The most important goal is for the agent to maintain its energy, obstacles should be avoided when possible and the activity goal is secondary. The homeostatic-variable values were categorized according to Table 2.

State change predictions were only considered for the Energy and the Activity variables. In the Energy case, two predictions are made. A small value prediction is made whenever the light detected by the sensors is above a certain threshold (0.4) and its value has just changed significantly¹. Another, higher-valued, prediction is made whenever the agent detects significant changes in energy available to re-charge. The actual values of the predictions are:

- $p(I_a)$ as expressed in Equation 4, with I_a being the energy availability, when the agent detects significant changes in energy availability; or
- $p(I_l)/2$ with I_l equal to light intensity, if there is solely a detection of a light change.

$$p(I) = \begin{cases} I & \text{if } I \text{ has increased} \\ -0.5(1 - I) & \text{if } I \text{ has decreased} \end{cases} \quad (4)$$

The Activity prediction is a sort of no-progress indicator given at regular time intervals when the activity of the robot is low for long periods of time. This is in fact a negative prediction (value of -1), because it predicts future failure in restoring Activity to its target state if the current behavior is maintained since it has failed to do it in a reasonable amount of time. It is important that the agent's behavior selection is triggered in these situations, otherwise a non-moving agent will eventually run out of events.

¹A significant change is detected when its value is statistically different from the values recorded since a state transition was last made, *i.e.* if the difference between the new value and the mean of the previous values exceeds both a small tolerance threshold (set to 0.02) and ξ times the standard deviation of those previous values (the ξ constant was set to 2.5).

Coefficient	Definition	Value	
c_s	State coefficient	1.0	
$c_t(h)$	State transition coefficient		
	h changed from/to:	– Target	1.0
		– Danger	-1.0
		Target Recover	-1.0
		Danger Recover	1.0
h did not change state		0.0	
c_p	Prediction coefficient	0.5	

Table 1: Coefficient values used in the experiments.

Homeostatic variable	Optimal	Acceptable	Deficient	Dangerous
Energy	[1.0, 0.9]	(0.9, 0.6]	(0.6, 0.2]	(0.2, 0.0]
Welfare	[1.0, 0.9]	(0.9, 0.7]	–	(0.7, 0.0]
Activity	[1.0, 1.0]	(0.9, 0.8]	–	(0.8, 0.0]

Table 2: Value intervals of the different qualitative categories of the homeostatic variables.

3.2 Adaptive System

The adaptive system implemented is a well known reinforcement-learning algorithm which has given good results in the field of robotics: Q-learning (30). Through this algorithm the agent learns iteratively by trial and error the expected discounted cumulative reinforcement that it will receive after executing an action in response to a world state, *i.e.* the utility values.

Traditional Q-learning usually employs a table, which stores the utility value of each possible action for every possible world state. In a real environment, the use of this table requires some form of partition of the continuous values provided by sensors. An alternative to this method suggested by (15) is to use neural networks to learn by back-propagation the utility values of each action. This method has the advantage of profiting from generalization over the input space and being more resistant to noise, but on the other hand neural-networks on-line training may not be very accurate. The reason being that the neural networks have a tendency to be overwhelmed by the large quantity of consecutive similar training data and forget the rare relevant experiences. Using an asynchronous triggering mechanism as the one proposed by the current architecture can help with this problem by detecting and using only a few relevant examples for training.

The system uses one feed-forward neural network per behavior, with: 7 input units, six representing state information plus one bias; 10 hidden units; and 1 output unit that represents the expected outcome of the associated behavior. The state information fed to the neural-networks comprises the homeostatic variable values and three perceptual values: light intensity, obstacle density and energy availability². All these values vary between 0 and 1.

The developed controller tries to maximize the reinforce-

ment received by selecting between one of three possible hand-designed behaviors:

Avoid obstacles — Turn away from the nearest obstacle and move away from it. If the sensors cannot detect any obstacle nearby, then remain still.

Seek Light — Go in the direction of the nearest light. If no light can be seen, remain still.

Wall Following — If there is no wall in sight, move forwards at full speed. Once a wall is found, follow it. This behavior by itself is not very reliable in that the robot can crash, *i.e.* become immobilized against a wall. The avoid-obstacles behavior can easily help in these situations.

At each trigger step, the agent may select between performing the behavior that has proven to be better in the past and therefore has the best utility value so far, or selecting an arbitrary behavior to improve its information about the utility of that behavior. The selection function used is based on the Boltzmann-Gibbs distribution and consists of selecting a behavior with higher probability, the higher its utility value in the current state.

4 The Experimental Procedure

The evaluation of the controller's success in the task described is not straightforward. To start with, the agent must be successful in accomplishing each one of its goals, which does not allow for a direct single-dimension evaluation between different controllers.

Furthermore, knowing if the agent is learning its task is not trivial. On the one hand, it is possible that the agent may solve its task simply by taking advantage of implicit domain knowledge such as the information provided by the

²High if a nearby energy source is releasing energy.

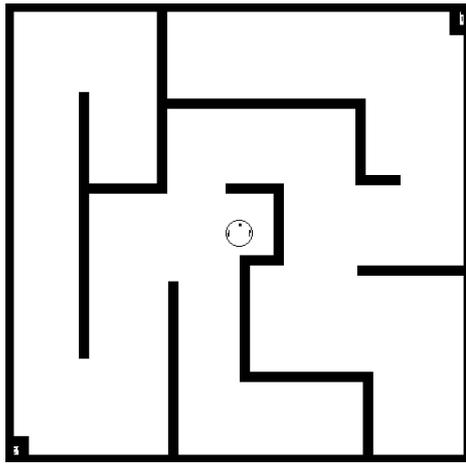


Figure 4: The simulated robot and its environment.

behavior switching mechanism and the knowledge already contained in the hand-designed behaviors. On the other hand, it is not clear how well a reasonably competent controller can manage all the different goals simultaneously. For those reason, it is important to compare the performance of the controller with those of a random behavior-selection controller and of an alternative controller which is competent at the task.

Secondly, although results may be artificially reproduced by simulation, robots in the real world are not expected to face exactly the same situations twice. Minor environmental changes or slightly different sensor or motor outcomes are likely to lead to very different experiences. This allied to the fact that the controller makes use of some random decisions in the learning exploration process make the results of a single trial test unreliable. For this reason, a rigorous evaluation of a controller requires several trials.

Each experiment consisted of having thirty different robot trials of three million learning steps. In each trial, a new fully recharged robot with all state values reset was placed at a randomly selected starting position. For evaluation purposes, the following statistics were taken:

Energy — mean energy level of the robot;

Distance — mean value of the Euclidean distance d , taken at one hundred steps intervals³, between the opposing points of the rectangular extent containing all the points the robot visited during the last interval, it is a measure of how much distance was covered by the robot;

Collisions — percentage of steps involving collisions;

All the experiments were carried out in a realistic simulator developed by (20) of a Khepera robot — a small robot with a left and a right wheel motor, and eight infrared sensors that allow it to detect object proximity and ambient

light. Six of the sensors are located in the front of the robot and two in the rear. The robot environment — Figure 4 — consisted of a closed environment with some walls and two lights surrounded by bricks on opposite corners.

5 Results

The proposed controller has empirically shown its competence by exhibiting a performance similar to the emotional controller discussed previously — see Table 3. Previous exhaustive experiments on the emotional controller have shown that it was quite competent and performed better than more traditional approaches (13; 12). In fact, previous experiments on learning behavior selection reported by the designer of the adaptive system selected (15) had to resort to severe simplifications of the behavior selection learning task. These simplifications included having behaviors associated with very specific pre-defined conditions of activation and only interrupting a behavior once it had reached its goal or an inapplicable behavior had become applicable.

The table also shows the results for a random controller which selects a behavior randomly at regular intervals⁴. These results show that both learning controllers significantly improve performance at all levels.

To assess the necessity of each of its properties, the controller had properties removed one at a time and was empirically compared against the complete controller. The experimental results obtained are presented in Table 3 and the conclusions reached are the following.

The behavior interruptions provided both by state transition and prediction of state transition proved essential to the performance of the task. The former are responsible for interrupting the behavior when a problem arises or has been solved. The latter allow the agent to take the necessary steps to accomplish its aims. In particular, a controller with no Energy predictions is not able to acquire energy and a controller with no Activity predictions will eventually stop moving.

In terms of reinforcement, all types of contributions were found valuable and the controller was fairly robust against changes in the relative weights of influences of homeostatic state, state transitions and predictions. The controller is able to learn successfully without the predictions influence on reinforcement, but the time to convergence is slower.

It was also found that, for the successful accomplishment of the task and in particular the achievement of their respective goals, all homeostatic variables should be taken into account in the reinforcement. Agents without activity reinforcement showed that it is more profitable for the agent to move as a last resort only when its energy is low and there is no light nearby. Avoiding moving helps to reduce the number of collisions.

The controller's success is quite sensitive to the correct adjustment of the relative weights of each homeostatic vari-

³The robot takes approximately this number of steps to efficiently move between corners of its environment.

⁴The interval selected was 35 steps based on previous results (11) which indicated this value as the most suited for the task.

Controller	Energy	Collisions	Distance
Proposed	0.53 ± 0.02	0.94 ± 0.15	1.87 ± 0.03
Emotional	0.54 ± 0.01	1.65 ± 0.48	1.96 ± 0.02
Random	0.02 ± 0.01	3.64 ± 0.27	0.83 ± 0.01
Proposed controller with behavior-switching not triggered by:			
State change	0.50 ± 0.02	2.07 ± 0.23	2.10 ± 0.02
Prediction	0.00 ± 0.00	2 bumping	3 moving
Energy	0.00 ± 0.00	0.00 ± 0.00	2.37 ± 0.00
Activity	0.00 ± 0.00	3 bumping	0.00 ± 0.00
Proposed controller with reinforcement not affected by:			
State ($c_s = 0$)	0.42 ± 0.04	0.57 ± 0.10	2.16 ± 0.01
State change ($c_t(h) = 0$)	0.24 ± 0.05	0.76 ± 0.27	2.03 ± 0.05
Prediction ($c_p = 0$)	0.48 ± 0.03	0.98 ± 0.22	1.84 ± 0.04
Energy ($w_{energy} = 0$)	0.10 ± 0.03	0.21 ± 0.05	1.97 ± 0.06
Welfare ($w_{welfare} = 0$)	0.58 ± 0.02	11.1 ± 2.10	1.80 ± 0.05
Activity ($w_{activity} = 0$)	0.48 ± 0.02	0.48 ± 0.13	0.92 ± 0.11

Table 3: Summary of the controllers' performance. It shows the means of the values obtained in the last three hundred thousand steps of each trial. Errors represent the mean 95% confidence intervals. Results are presented for the proposed controller, the reference emotional controller, a random controller and modified versions of the proposed controller. Modifications consisted of disregarding specific trigger events or selectively dropping influences on reinforcement. In the former case, and in particular if activity prediction events were ignored, the agent would eventually stop receiving triggering events altogether. This would usually happen with the agent stopped in an isolated position, but sometimes it would also happen to a moving agent or to an agent crashed into a wall. These exception cases are accounted for in the table.

able on reinforcement. This is a problem introduced by the proposed architecture which did not arise in the emotional controller. In fact, this is the only reason why this controller may be considered worse than the emotional controller: it required extra design effort.

6 Related Work

The idea of homeostatic values stems from neuro-physiological research on emotions (8; 7) and has been modeled previously by the DARE model (23; 16; 29). In the DARE model, which emphasizes the dual nature of decision making where both emotions and cognition take part, there is a body with target values which has a central role in the evaluation of situations.

There are other robot emotion-based architectures which rely on homeostatic variables. An example is the robot architecture developed by (10) which learns emotionally grounded symbol-object associations. In this case, there are a few internal variables which trigger drives when they are out of their target values. Drives have pre-defined associated behaviors and the robot only learns about differently-colored objects, namely how they may change its internal variables. Similarly to the current work there are innate emotions which are derived from monitoring the internal variables and associated emotions. However, the innate emotions only monitor changes of the internal variables values into/from their target values, and the associated emotions are not associated with behavior-state pairs

but with objects.

Another example, is the Kismet robot (5) whose drives have an acceptable bounds of operation named the homeostatic regime. If the drive's value is below these bounds then it is in the overwhelmed regime, if above then it is in the under-stimulated regime. Drives, along with somatically-marked releasing mechanisms, influence the affective state by contributing to the valence and arousal measures. If the drive is in the homeostatic regime, then there is a positive contribution to the valence, otherwise the contribution is negative. The contribution to arousal decreases with the drive value. Only the currently active drive, *i.e.* the one whose pre-defined associated behavior has been selected, influences the emotion state. Arousal, valence and stance are the three dimensions of the affective state. Emotions are defined as points in these space and are expressed by Kismet's face in its interaction with a caregiver. This approach shares the use of homeostatic variables but has a very different model of emotions based on a three-dimensional continuous space instead of processes⁵ and the task of the agent is quite different.

In other architectures (*e.g.*, 3; 6; 28), homeostatic variables are monitored to produce drives and do not have any direct relation with emotions⁶.

⁵There are adepts of the two types of emotion models, although arguments against defining emotions in terms of a few continuous dimensions seem stronger (9).

⁶Note that specific domain-dependent dependencies may be hand-coded by the designer when defining the activation conditions of the emotions.

All these architectures are quite different in that there is a hard-coded relationship between the drives and the produced behavior, while in the proposed architecture the agents learn how to satisfy their goals or each goal to satisfy at any one point by choosing among available behaviors.

7 Conclusions

The current work proposes a new architecture for learning behavior coordination which is inspired by emotions. Its goal system, in particular, is based on homeostatic processes which bare similarities with foreground and background emotions. In fact, (8; 7) refers homeostasis as central to emotional processes. Furthermore, the associations made by the adaptive system are akin to somatic-markers suggested by (8). Both provide a long-term indication of the “goodness” of the several options available to the agent in a certain situation, based on previous experiences. Emotions as used in this architecture, provide a low-level processing of internal homeostatic state and relevant perceptions which is used both in evaluation of the situation (or more specifically, reinforcement to the learning system) and interruption of behavior. These are two processes which have also been strongly associated with emotions by other researchers. Another distinctive feature of the proposed architecture is that only changes in perception which are relevant in terms of the agent current internal state are brought to its attention.

In this work, an engineering approach (31) is taken towards emotions. This means that there is not so much an emphasis on attempting to have a replica of human emotions, as there is on having a competent architecture.

For this reason, this architecture was subject to rigorous experiments which thoroughly evaluate the different aspects of the proposed architecture and compared it with other alternatives. Although the experiments were done in simulation, the robot faced a demanding task which keeps the essential problems of a real-world environment.

Experiments demonstrated the validity of the architecture, by showing that it is very competent in accomplishing the task it was designed for. Furthermore, this architecture clearly specifies how the learning process should be controlled, namely what the reinforcement should be and when the behaviors should be interrupted, once the domain-dependent goals of a task are identified.

8 Future Work

In the architecture presented, the goal system must be tailored to the task at hand so that it reflects its aims, whereas the adaptive system is more flexible and may solve different tasks when associated with different goal systems. However, the goal system does not need to be totally hand-designed. One may envisage an adaptive goal system

where subgoals are found or new perceptual cues for prediction of internal state changes are uncovered. This way the goal system would model some of the emotional associations animals and humans create around specific events or situations. This would be in-line with the theory that during learning stimuli are primarily associated with emotions which then drive the behavior associations (21).

One of the most difficult problems was to determine the relative weights of importance of the different homeostatic variables. This suggests that the homeostatic variables may have to be associated with different adaptive systems to be combined in a later stage for final behavior selection. This way the information required to pursue each goal can be kept separate.

Acknowledgement

The first author is a post-doctoral fellow sponsored by the Portuguese Foundation for Science and Technology. This work was partially supported by the FCT *Programa Operacional Sociedade de Informação* (POSI) in the frame of QCA III.

References

- [1] James S. Albus. The role of world modeling and value judgment in perception. In A. Meystel, J. Herath, and S. Gray, editors, *Proceedings of the 5th IEEE International Symposium on Intelligent Control*. Los Alamitos, CA: IEEE Computer Society Press, 1990.
- [2] Minoru Asada. An agent and an environment: A view on body scheme. In Jun Tani and Minoru Asada, editors, *Proceedings of the 1996 IROS Workshop on Towards real autonomy*, pages 19–24, Senri Life Science Center, Osaka, Japan, 1996.
- [3] Bruce Blumberg. *Old Tricks, New Dogs: Ethology and interactive creatures*. PhD thesis, MIT, 1996.
- [4] Stevo Bozinovski. A self-learning system using secondary reinforcement. In R. Trappl, editor, *Cybernetics and Systems*, pages 397–402. Elsevier Science Publishers, North Holland, 1982.
- [5] Cynthia Breazeal. Robot in society: Friend or appliance? In *Agents'99 workshop on emotion-based agent architectures*, pages 18–26, Seattle, WA, 1999.
- [6] Dolores Cañamero. Modeling motivations and emotions as a basis for intelligent behavior. In *Proceedings of the First International Symposium on Autonomous Agents, AA'97*, Marina del Rey, CA, February 1997. The ACM Press.
- [7] Antonio Damasio. *The feeling of what happens*. Harcourt Brace & Company, New York, 1999.
- [8] Antonio R. Damasio. *Descartes' error — Emotion, reason and human brain*. Picador, London, 1994.

- [9] Paul Ekman. An argument for basic emotions. *Cognition and Emotion*, 6(3/4):169–200, 1992.
- [10] Masahiro Fujita, Rika Hasegawa, Gabriel Costa, Tsuyoshi Takagi, Jun Yokono, and Hideki Shimomura. Physically and emotionally grounded symbol acquisition for autonomous robots. In Lola Cañero, editor, *AAAI Fall Symposium on Emotional and Intelligent II: The tangled knot of social cognition*, pages 43–48. Menlo Park, California: AAAI Press, 2001. Technical report FS-01-02.
- [11] Sandra Clara Gadanho. *Reinforcement Learning in Autonomous Robots: An Empirical Investigation of the Role of Emotions*. PhD thesis, University of Edinburgh, 1999.
- [12] Sandra Clara Gadanho and John Hallam. Emotion-triggered learning in autonomous robot control. *Cybernetics and Systems — Special Issue: Grounding emotions in adaptive systems*, 32(5):531–559, July 2001.
- [13] Sandra Clara Gadanho and John Hallam. Robot learning driven by emotions. *Adaptive Behavior*, 9(1), 2001.
- [14] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
- [15] Long-Ji Lin. *Reinforcement learning for robots using neural networks*. PhD thesis, Carnegie Mellon University, 1993. Technical report CMU-CS-93-103.
- [16] Márcia Maças, Paulo Couto, Carlos Pinto-Ferreira, Luis Custódio, and Rodrigo Ventura. Experiments with an emotion-based agent using the DARE architecture. In *Proceedings of the AISB'01 Symposium on Emotion, Cognition and Affective Computing*, pages 105–112, University of York, U. K., March 2001.
- [17] Sridhar Mahadevan and Jonathan Connell. Automatic programming of behavior-based robots using reinforcement learning. *Artificial intelligence*, 55:311–365, 1992.
- [18] Yuval Marom and Gillian Hayes. Maintaining attentional capacity in a social robot. In R. Trappl, editor, *Cybernetics and Systems 2000: Proceedings of the 15th European Meeting on Cybernetics and Systems Research. Symposium on Autonomy Control — Lessons from the emotional*, volume 1, pages 693–698, Vienna, Austria, April 2000.
- [19] Maja J. Mataric. Reward functions for accelerated learning. In William W. Cohen and Haym Hirsh, editors, *Machine Learning: Proceedings of the Eleventh International Conference*, pages 181–189. San Francisco, CA: Morgan Kaufmann Publishers, 1994.
- [20] Olivier Michel. *Khepera Simulator* package version 2.0: Freeware mobile robot simulator written at the University of Nice Sophia-Antipolis, March 1996. Downloadable from the World Wide Web at <http://diwww.epfl.ch/lami/team/michel/khep-sim/>.
- [21] O. Hobart Mowrer. *Learning theory and behavior*. John Wiley & Sons, Inc., New York, 1960.
- [22] Miguel Rodriguez and Jean-Pierre Muller. Towards autonomous cognitive animats. In F. Morán, A. Moreno, J.J. Merelo, and P. Chacón, editors, *Advances in artificial life — Proceedings of the Third European Conference on Artificial Life*, Lecture Notes in Artificial Intelligence Volume 929, Berlin, Germany, 1995. Springer-Verlag.
- [23] Rui Sadio, Gonçalo Tavares, Rodrigo Ventura, and Luis Custódio. An emotion-based agent architecture application with real robots. In Lola Cañero, editor, *AAAI Fall Symposium on Emotional and Intelligent II: The tangled knot of social cognition*, pages 117–122. Menlo Park, California: AAAI Press, 2001. Technical report FS-01-02.
- [24] H. A. Simon. Motivational and emotional controls of cognition. *Psychological Review*, 74:29–39, 1967.
- [25] Aaron Sloman and Monica Croucher. Why robots will have emotions. In *IJCAI'81 — Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 2369–71, 1981. Also available as Cognitive Science Research Paper 176, Sussex University.
- [26] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning*. The MIT Press, 1998.
- [27] Silvan S. Tomkins. Affect theory. In Klaus R. Scherer and Paul Ekman, editors, *Approaches to Emotion*. Lawrence Erlbaum, London, 1984.
- [28] Juan D. Velásquez. A computational framework for emotion-based control. In *SAB'98 Workshop on Grounding Emotions in Adaptive Systems*, pages 62–67, Zurich, Switzerland, 1998.
- [29] Rodrigo Ventura and Carlos Pinto-Ferreira. Emotion-based agents: Three approaches to implementation (preliminary report). In Juan D. Velásquez, editor, *Workshop on Emotion-Based Agent Architectures*, Seattle, U. S. A., 1999. Workshop of the Third International Conference on Autonomous Agents.
- [30] C. Watkins. *Learning from delayed rewards*. PhD thesis, King's College, Cambridge, 1989.
- [31] Thomas Wehrle. Motivations behind modeling emotions agents: Whose emotions does your robot have? In *SAB'98 Workshop on Grounding Emotions in Adaptive Systems*, pages 71–76, Zurich, Switzerland, 1998.

Multiple Emotion-Based Agents using an Extension of *DARE* Architecture

Márcia Maças, Luis Custódio

Institute for Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal

{marcia,lmcc}@isr.ist.utl.pt, <http://www.isr.ist.utl.pt/~islab>

Keywords: agent architecture, emotions, society of agents

Received: October 30, 2002

The role of emotions in human intelligence and social behaviours has been considered very important in the past years. The DARE architecture, an emotion-based agent architecture, aims at the modelling of this contribution for building autonomous agents. In this paper the results of its application to a multiple agents environment are presented. Emotions are used at an individual decision level, through the modelling of the somatic marker hypothesis, and are also used on decisions that involve others, using the same hypothesis and adding the notion of sympathy. The representation of other agents external expression allows to predict their internal state. This process is based on the assumption that similar agents express their internal state in similar way, being a mean of implicit communication. Sympathy allows more informed individual decisions, specially when these depend on others. On the other hand it makes agents learn, not only based on their own experience, but also with others experience. Besides implicit communication, it is also used explicit communication, through messages exchanging. In the symbolic layer, a new layer added to the DARE architecture, interactions between agents are represented and used to improve individual and social behaviours.

1 Introduction

Recent research findings on the neurophysiology of human emotions suggest that human decision-making efficiency depends deeply on the emotions machinery. In particular, the neuroscientist António Damásio (3) claims that alternative courses of action in a decision-making problem are (somatically) marked as good or bad, based on an emotional evaluation. Only the positive ones (a smaller set) are used for further reasoning and decision purposes. This constitutes the essence of the Damásio's somatic marker hypothesis. In another study about emotions, conducted by the neuroscientist Joseph LeDoux (8), it is recognized the existence of two levels in the sensorial processing, one quicker and urgent, and another slower but more informed.

*DARE*¹ architecture for emotion-based agents is essentially grounded on these theories about emotions neurological configuration and application. In previous work the application of this architecture focused on decision-making at the agent's individual level (15; 16; 9; 14; 11).

In the somatic marker hypothesis, the link between emotions and decision-making is suggested as particularly strong for the personal and social aspects of human life. Other emotion theories, mainly in psychology, focus on the social aspects of emotion processes. The work presented here tries to explore these notions and the importance of emotional physical expression on social interactions, as well as the sympathy that may occur in those in-

teractions.

In what concerns emotion expression, it has been claimed that there is not another human process with such a distinct mean of physical communication, and more interesting it is unintentional (7). Some theories point out that emotions are a form of basic communication and are important in social interaction (Rivera, Oatley and Johnson-Laird in (13)). Others propose that physical expression of emotion is the body preparation to act (4), where emotional response can be seen as a built-in action tendency aroused under pre-defined circumstances. This can also be a form of communicating to others what will be the next action. If the physical message is understood it may defuse emotions in others, establishing an interactive loop with or without actions in the middle (Dantzer cited in (2)). The AI research concerning multi-agent systems relies mainly on rational, social and communication theories. However, the role of emotions in this field has been considered important by an increased number of researchers (10; 12; 2; 1).

Linked to expressing emotions is the notion of sympathy defined as the human capability to recognize others' emotions (5). This capability is acquired by having consciousness of our own emotions. Humans can use it to evaluate others' behaviours and predict their reactions, through a mental model learned by self-experience or by observation that relates physical expression with feelings and intentions. Sympathy provides an implicit communication mean, sometimes unintentional, that favours social interactions.

In this paper the results of the application of the *DARE* architecture to a multi-agent environment are presented.

¹*DARE* stands for **E**motion-based **R**obotic **A**gent **D**evelopment (in reverse order). This work has been developed under the framework of a research project funded by the Portuguese Foundation for Science and Technology (project PRAXIS/P/EEI/12184/98).

In this architecture, emotions play a role on individual decision-making based on the somatic marker and the stimuli double processing hypotheses. These concepts are extended for decision-making involving other agents.

Agents represent others' external expression in order to predict their internal state, assuming that similar agents express the internal state in the same way (a kind of implicit communication). Sympathy is grounded on this form of communication, allowing more informed individual decisions, specially when these depend on others. On the other hand, it allows the agent to learn, not only by its own experience, but also by the observation of others' experience. The architecture also allows the modelling of explicit communication through the incorporation of a new layer, the symbolic layer, where relations between agents are represented and processed.

2 DARE Architecture

The *DARE* architecture was applied to an environment that simulates a simple market involving: producer agents, that own products all the time; supplier agents, that must fetch products from producers or other suppliers either for its own consumption or for selling to consumers; and consumer agents, that must acquire products from suppliers for its own consumption. Agents are free to move around the world, interact and communicate with others. Their main goal is to survive by eating the necessary products and, additionally, maximize money by selling products.

Figure 1 shows a global view of the *DARE* architecture. Stimuli received from the environment are processed in parallel on three layers: perceptual, cognitive and symbolic. Several stimuli are received simultaneously, and they can be gathered from any type of sensor.

In the case of the market experiment, agents receive both visual and auditive stimuli. Auditive stimuli are strings with messages exchanged between agents or broadcasted. Visual stimuli consists of all the objects (agents and respective products) inside the agent's vision angle.

For each stimulus, three internal representations, which will be evaluated on the corresponding layer, resulting on the selection of an action to be executed by the agent, are created. Meanwhile, the world may change due to its dynamics or as a consequence of other agents actions.

In the market application agents actions consist of movements, picking products, eating them, and exchanging messages. Every action executed has an effect on the agent's internal state, which when added to the next perceived stimuli is used to update memory and feature meanings.

The perceptual analysis evaluates stimuli based on i) a pre-defined set of relevant features, ii) their meanings and iii) the current internal state. This analysis results on a fast action selection which will be executed if the global situation (stimuli and internal state) is considered urgent, in which case the upper layers will be inhibited. The action selected in the perceptual layer will also be executed when-

ever the upper layers are not able to select an action due to lack of information. Cognitive and symbolic analysis use memory to evaluate stimuli and predict action effects based on similar actions executed or observed in the past. The internal state and its changes are crucial to the evaluation and anticipation processes on these layers.

2.1 Perceptual Layer

2.1.1 Feature Extraction and Built-in Information

Figure 2 presents the perceptual layer in more detail. When stimuli are acquired by the agent's sensors a set (\mathcal{RF}) of pre-defined, simple and relevant features are quickly extracted. This extraction provides a basic and simple internal representation of each stimulus, called the perceptual image (I_p). The I_p assembles the amount of each relevant feature found in the stimulus. Since several stimuli can be sensed at the same time the set of all perceptual images at instant t is defined as \mathcal{IP}^t . All perceptual images are evaluated based on the agent's internal state at instant t and pre-defined associations between features and meanings.

The agent's internal state, \mathcal{IS} , is a vector with a pre-defined number of components specified for each agent. The contents of this vector varies as consequence of actions execution. The ideal contents of \mathcal{IS} is pre-defined by the homeostatic vector \mathcal{HV} . Both \mathcal{IS} and \mathcal{HV} are crucial for agent behaviour, since the main goal of an agent implemented with this architecture is to approach its internal state to the ideal one.

Every evaluation takes into account the unbalance of the internal state, δ^t , *i.e.*, the difference between both vectors,

$$\delta^t = \varphi(\Delta(is_1, hv_1), \Delta(is_2, hv_2), \dots, \Delta(is_p, hv_p))$$

where is_i and hv_i are components of \mathcal{IS} and \mathcal{HV} , respectively; Δ is a function that computes their difference; and φ is a function that processes all the differences in order to qualify them. This function may be the processing of thresholds or an application-dependent function which analyse/process specific patterns in its arguments. The unbalance is reflected on the agent's external expression.

At the market experiment, the agents internal state consists of a set of nutrients,

$$\mathcal{IS}^t = [glycides^t, proteins^t, fatty^t, sugar^t]$$

which are decremented in every movement action, proportionally to the power used in it, and are also changed whenever the agent eats a product. Different products mean different changes on the nutrients, some might be increased, others decreased, depending on colours present in the product image. Initially, the agents are in perfect balance ($\mathcal{IS} = \mathcal{HV}$). The unbalance is a vector with the difference between each nutrient current value and its ideal,

$$\Delta(is_i, hv_i) = is_i - hv_i$$

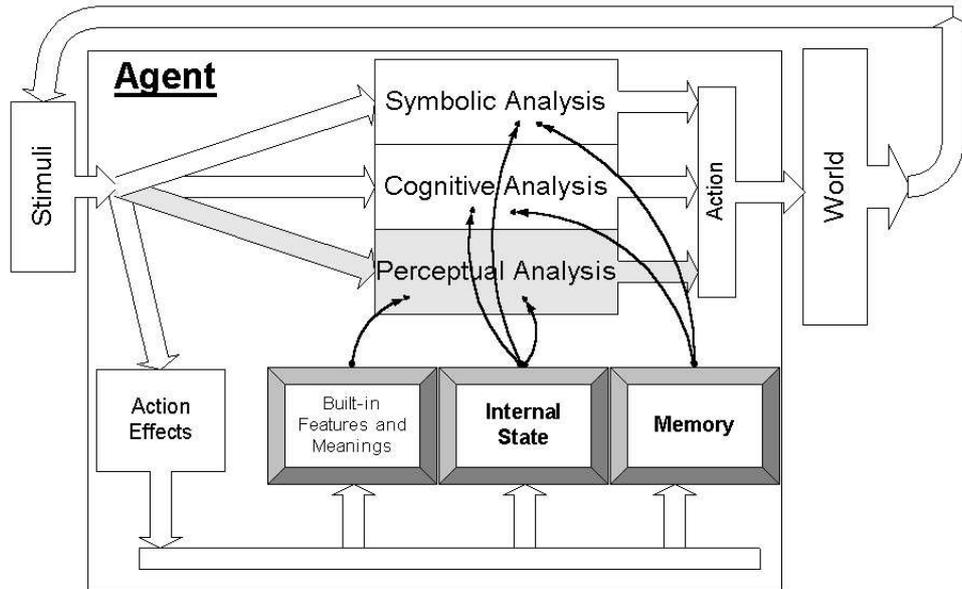


Figure 1: Global view of DARE architecture.

where the nutrient with maximum absolute difference defines the agent’s current need.

$$\varphi(\Delta(is_1, hv_1), \dots, \Delta(is_p, hv_p)) = \begin{cases} \arg \max_j \Delta(is_j, hv_j) & \text{If } \Delta(is_j, hv_j) < \epsilon_{min} < 0 \\ \arg \min_j \Delta(is_j, hv_j) & \text{If } \Delta(is_j, hv_j) > \epsilon_{max} > 0 \end{cases}$$

This internal unbalance is mapped into images that represent the external expression of the agent.

2.1.2 Perceptual Evaluation

The perceptual evaluation tries to qualify the presence of relevant features in the I_p . This qualification is based on the mapping between the features and their pre-defined meanings, taking into account the current internal state. The result is a perceptual desirability vector, DV_p , which represents a basic, simple and fast assessment of a stimulus.

In the case of the market implementation, visual stimuli are bitmaps and the relevant features are some colours in agents and products images. The relevant colours for agent images are red, yellow and green, whereas for product images are dark red, dark green, dark yellow, dark magenta, dark gray, red, green, yellow and magenta. The extraction of relevant features is simply the counting of pixels for relevant colours in the bitmap, *i.e.* the perceptual image is the set of the number of pixels for each relevant colour.

The meaning–feature association, represented by pre-defined weights w_{nf} , establishes the goodness or badness of each colour, where positive weights mean good colours and negative ones mean bad colours. In this implementation, all relevant colours are initially considered positive.

So DV_p is the result of processing w_{nf} and I_p components.

$$DV_p = \sum_f w_{nf} I_{pf}$$

where w_{nf} represents how good feature (colour) f is for nutrient n .

After the perceptual evaluation of all stimuli detected at instant t , the stimulus found as being more desirable is selected as the incentive for action, and its I_p and DV_p are used to select the action.

2.1.3 Action Selection and Evaluation

There is a pre-defined set of simple actions that can be selected at the perceptual layer (e.g., approach, avoid, wander, pick, and eat). At this layer the action selection is based on reactive rules designed to cover urgent situations where the agent must survive. When the I_p and DV_p of the incentive stimulus satisfy the pre-conditions of a action rule, this action will be selected.

For instance, a very hungry agent near a product will immediately select the eat action; if it is not near enough it will select a movement action to approach it.

After an action being executed, the action, all the I_p ’s and DV_p ’s of the processed stimuli, and the action effect on the agent internal state are associated and stored in memory. In future similar circumstances, upper layers may anticipate action effects and decide accordingly.

At the perceptual layer the action effects are evaluated in order to adjust the weights that represent the meaning of relevant features. If the internal state changes over a threshold after an action being executed, and this change

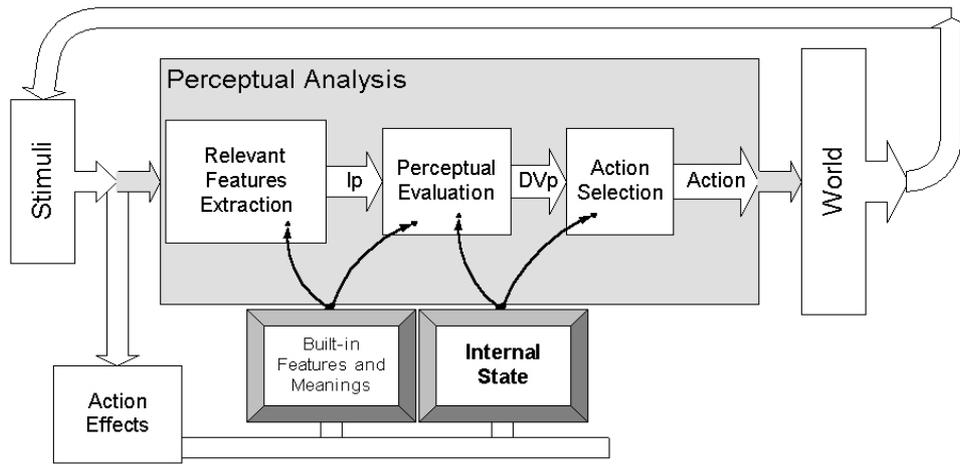


Figure 2: DARE architecture – Perceptual Layer.

means a strong approach or deviation to/from balance, the weights that lead to its selection are adapted in order to reflect this knowledge,

$$\text{If } \exists_n, \text{ that } |\Delta(\mathcal{I}S_n^t, \mathcal{I}S_n^{t+1})| > \eta_n, \text{ then}$$

$$w_{nf}^{t+1} = \begin{cases} w_{nf}^t + \tau & \text{If } \delta^{t+1} < \delta^t \\ w_{nf}^t & \text{If } \delta^{t+1} = \delta^t \\ w_{nf}^t - \tau & \text{If } \delta^{t+1} > \delta^t \end{cases}$$

where f is the predominant relevant feature of the incentive stimulus, η_n is the threshold that defines a major change in the component n , and τ is the adaptation value.

Since this evaluation of effects is based on stimuli and current internal state, this adaptation is only temporary and gradually the weights will return to their initial values. So, at each instant t ,

$$w_{nf}^{t+1} = \begin{cases} w_{nf}^t - 1 & \text{If } w_{nf}^t > w_{nf}^0 \\ w_{nf}^t + 1 & \text{If } w_{nf}^t < w_{nf}^0 \end{cases}$$

If the effect repeats often the adaptation ends up to be persistent. This process aims at giving some degree of adaptation and flexibility to the perceptual layer.

In the market implementation, when an agent eats a product that instead of increasing an unbalanced nutrient decreases it, the weight associated to the predominant colour of the product is decreased. In following decisions, the DV_p for this product will have less desirability than other products without this colour. The gradual return to initial values allows the agent to re-select this product later, because it could be desirable for a different nutrient.

In order to determine the urgency of a situation, thresholds for the desirability vector components are defined. Whenever the DV_p is detected as urgent (above or below the threshold) the upper layers are inhibited and the action selected at the perceptual level is executed immediately.

The contents of DV_p are influenced both by the stimuli and the internal state. If the internal state is very unbalanced the DV_p must reflect that situation in terms of the urgency to handle it. So the urgency of a situation is defined by a threshold w.r.t. the current unbalance,

$$|\Delta^t(j)| > \sigma_p$$

where j is the most unbalanced component (nutrient) of the internal state, determining urgency if its absolute difference to the ideal value is above σ_p .

2.2 Cognitive Layer

The perceptual adaptation is limited to simple stimulus features and a rough evaluation of the action effects on the internal state. Figure 3 presents the cognitive layer. From the sensed stimuli all the possible features are extracted, defined in the set \mathcal{F} , which satisfies the condition $\mathcal{R}\mathcal{F} \subset \mathcal{F}$. This extraction is computationally more heavy than the perceptual one, but will supply more information to distinguish stimuli. It is not only a quantification of the feature presence on stimulus, but also a processing that results on the stimulus full characterization, allowing identification. The result of this extraction is the cognitive image, I_c , of a stimulus. Since several stimuli can be sensed at the same time it is defined the set of all cognitive images at instant t as $\mathcal{I}\mathcal{P}^t$. In the market experiment, where relevant features are some colours present in the bitmap, the cognitive image is instead the full bitmap.

The purpose of this layer is to generate adequate individual behaviours² through learning by experience.

Cognitive evaluation and action selection is conditioned by the urgency found by the perceptual layer evaluation. Nevertheless, the $\mathcal{I}\mathcal{C}^t$ is always created and stored in mem-

²Social behaviours will be the purpose of the symbolic layer.

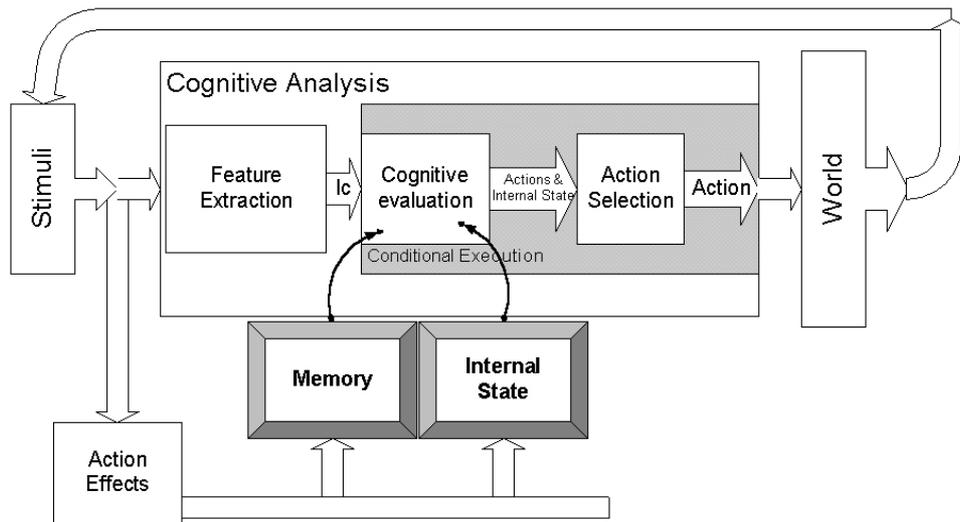


Figure 3: DARE architecture – Cognitive Layer.

ory, associated with the corresponding items stored by the perceptual layer.

2.2.1 Cognitive Evaluation and Action Selection

Mainly, the cognitive evaluation consists of a search in memory for situations similar to the current one. This process uses the current \mathcal{IP}^t to reduce search space, assuming that similar I_c 's have similar I_p 's. The structures in memory that have an incentive stimulus with an I_p similar to the one of the currently sensed stimulus are selected and the corresponding I_c in memory is compared with the current I_c^t . If they are similar the structure in memory is selected for further search.

Memory is usually structured as sequences of stimuli–actions associations that end whenever a significant change in the internal state is detected. Each element of a memory sequence must have i) the I_p^t and I_c^t of the incentive stimulus; ii) the \mathcal{IP}^t and \mathcal{IC}^t sets; iii) the executed action, and iv) its effect on the internal state. Each sequence must also have the overall change on the internal state, in order to determine the desirability of that course of action. Once sequences with a stimulus similar to one of the current set are found, the associated internal state changes are applied to the current internal state and the sequence that anticipates a more balanced internal state is chosen to be executed. Consider \mathcal{SEL} the set with all the matching sequences in memory; \mathcal{F}^m an element of a sequence; a^m the action represented in \mathcal{F}^m ; and Δ_{a^m} the change in the internal state caused by a^m execution. The action selected at the cognitive layer, a_c^t , is determined by

$$\forall a^m \in \mathcal{F}^m \in \mathcal{SEL}, a_c^t = \arg \min_{a^m} (\delta^t + \Delta_{a^m})$$

For instance, if a consumer agent has eaten in the past a product that made one nutrient increase by a certain value,

and now its internal state needs that nutrient, the agent will anticipate the increase and, if there is not other product with a better anticipation, will choose to approach and eat that product, executing the same action sequence retrieved from memory.

If the prediction made by the cognitive evaluation reveals a degree of urgency, *i.e.*, if it predicts, given an unbalanced internal state,

$$|\Delta^t(j)| > \sigma_c,$$

a strong and positive change,

$$|\Delta^t(j) + \Delta_{a_c^t}(j)| < \sigma_c,$$

the action is executed and the symbolic layer is inhibited, otherwise the symbolic layer is processed and may or not confirm the execution of the cognitively selected action. The threshold σ_c defines the urgency of an internal state component at the cognitive layer and must satisfy $\sigma_c < \sigma_p$.

After an action being executed, the internal state is changed and memory is updated. Memory management is very important in this architecture, because the environment is dynamic and continuous. A simple strategy consisting of storing all executed sequences is not an adequate approach. Eliminating repeated sequences could be done by abstracting some details but it is not enough, because as a sequence of stimuli-action associations seldom repeat themselves and internal state is also varying, few will be eliminated. A better strategy is the elimination based on irrelevance, eliminating those sequences that have smaller internal state changes.

2.3 Symbolic layer

The symbolic layer was introduced aiming at the capture of concepts involved in communication and sympathy. This

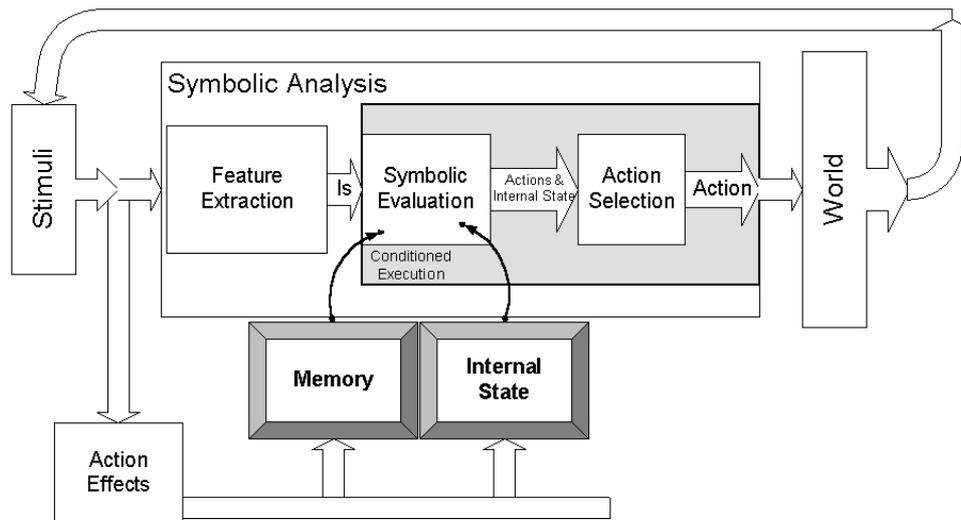


Figure 4: *DARE* architecture – Symbolic Layer.

layer has the same conceptual foundation of the cognitive layer in what concerns the modelling of the somatic marker hypothesis, allowing the same kind of adaptation and learning. In the symbolic layer those concepts are applied to more abstract information extracted from stimuli in order to i) establish communication between agents, ii) represent explicitly their goals and interactions and iii) trigger reasoning. Figure 4 presents the symbolic layer.

In this layer memory is used to determine a course of action like in the cognitive layer, but this notion is extended from the individual past experience to other agents' past experience, which have been observed by the deciding agent. This extension leads eventually to emergence of imitation behaviours. Because it is used a symbolic representation there is also place for logical inference allowing planning and reasoning.

2.3.1 Feature Extraction

The features that are extracted at this layer result on a set of symbols, the symbolic image I_s . Its construction relies not only on the extracted features but also on the agent ontology. Each agent has an initial pre-defined ontology \mathcal{O}_0 that maps the features present in a stimulus into symbols that describe it. There are two kinds of symbols, those which describe a certain property of the stimulus (descriptive symbols) and those which identify the stimulus, associating a name with a specific set of descriptive symbols. The ontology is updated³ based on experience, \mathcal{O}_t , through a new interaction with an already known stimulus, by communication with other agents or by the contact with a new stimulus.

Agents may not share the same identification symbols

³By adding new descriptive symbols and/or new associations between an identification symbol and sets of descriptive symbols.

for the same set of descriptive symbols, *i.e.*, the same stimulus may have different names across agents. In order to bootstrap communication and acquisition of new symbols, it is necessary that agents extract the same symbolic features, have the same set of initial descriptive symbols and share the meanings of the descriptive symbols defined on the initial ontology.

In the market implementation, the symbolic layer is the only layer that processes both the visual and the auditive stimuli. Symbols like round, sharp, symmetric, smile (for agents only), defined on the initial ontology, are extracted from the visual stimuli⁴. From the auditive stimuli the structure and elements of the messages are extracted, mapping the message contents with respect to recent visual stimuli. For instance, messages involved on a negotiation for a product between a supplier and a consumer are represented by the messages structure, the ontology meanings, and the recently sensed visual stimuli of the intervenients and of the subject of negotiation (product).

2.3.2 Extended Internal Representation

The agent internal state at the symbolic layer is extended in order to have components not directly related to survival. Examples of these components, on the market implementation, are the products possessed by the agent; the explicit representation of goals determined by negotiation involving product orders; and the money variable. The latter is processed in a similar way as nutrients, but represents quality of life instead of pure survival. For instance, when at lower layers the agent picks or eats a product without negotiation (by stealing) the agent's money is strongly decreased as a mean of punishment that is only processed at

⁴These features should ideally be extracted by image processing of the cognitive image. However, in this implementation, because this processing is computationally hard they were pre-defined for each image.

the symbolic layer, because in the lower layers money is not considered.

Therefore, it is defined the vector ELS as the extension of IS , where

$$IS \subset ELS$$

and the extended homeostatic vector EHV , where

$$HV \subset EHV$$

that defines the ideal values of the extended internal state. In the market implementation, the ELS was defined as follows,

$$ELS^t = [glycides^t, proteins^t, fatty^t, sugar^t, money^t]$$

At the symbolic layer new features in the external expression corresponding to the extended internal state of the agent are recognized. For example, there is a mapping between the money state and the image of the agent, that is only recognized at the symbolic layer and ignored at lower layers.

In the symbolic layer, it is maintained and used an internal representation of the agent itself. The internal state is processed by constructing an internal stimulus as if the agent was sensing itself in the world. All the internal images, I_p , I_c and I_s , of this internal stimulus are created in the same fashion as for external stimuli, and are considered for decision at all layers. This allows that an agent may eat the product it possesses and may compare its state with others' state.

2.3.3 Symbolic Evaluation

Although, the symbolic evaluation is conditioned by the result of the cognitive evaluation, the construction of the symbolic images and their storing in memory, associated with the corresponding I_p 's and I_c 's, is always performed.

If the symbolic evaluation is performed, memory is searched for similar situations based on the symbolic point of view. I_p and I_c are used to reduce space search, since it is assumed that similar I_s must have similar I_p 's and I_c 's. Memory at symbolic layer has different elements in order to represent symbolic images, new actions and the extended internal state effects.

The symbolic evaluation shares the same functional concepts of the cognitive one, except that these are extended to social representations. The current stimuli symbolic structure is compared to those in memory, which may be structured into sequences that terminate whenever a major change on an agent expression of its internal state (the agent's own expression or others' expressions observed by it) is found. When a sequence terminates due to a change on the expression of the agent itself, this sequence corresponds to sequences in the cognitive layer, focusing on the individual welfare, except when the change is caused by an element that exists only on the extended internal state (like money, for instance). When a sequence ends because

of another agent change of expression, it is evaluated as if it was the own agent. This is the process of gathering and storing others' experiences. Search is generalized to all agents experience, not only the self experience. However, there are some conditions to this generalization: first, the agent will only capture others' experiences that have been observed by it, and second, only considers experiences of agents with expression images similar to its own, in order to map the change of expression into a change on its internal state. The symbolic layer is specially useful on communication actions, by listening dialogs between other agents and observing their changes of expression, in order to imitate or not the observed course of action.

Besides possible imitation behaviours, this mechanism of observing expressions and actions of other agents also allows the agent to anticipate other agents' actions. Nevertheless, there is the possibility for the agent to make mistakes on the assessment of others' internal state. Depending on the application, an expression may not be directly mapped to a specific internal state but only to a set of internal states. Moreover, some changes of expression may not be a direct effect of the last action executed.

2.3.4 Action Selection

At the symbolic layer there is a new set of actions that can not be selected in any other layer due to symbolic representation requirements, mainly communication actions. The action selection at the symbolic layer is similar to the cognitive one but it takes into account the extended internal state. So, sequences with a structure (\mathcal{F}_{simb}^m) that has an incentive stimulus similar to one of those currently sensed, and that end up with a better internal state or with a positive change of expression, are selected (\mathcal{SEL}_{simb}) and the best determines the action a_s^t to be executed,

$$\forall a^m \in \mathcal{F}_{simb}^m \in \mathcal{SEL}_{simb}, \\ a_s^t = \arg \min_{a^m} (E\delta^t + E\Delta_{a^m})$$

Due to the symbolic representation, this action selection process may be complemented with logical reasoning and planning applied to the contents of memory. In this way it is possible to generate goals and anticipate the best way to achieve them in a long term perspective.

The sympathy and communication can still be used by this reasoning process, assessing other agents' internal state through their external expression, and allowing prediction of their actions. On the other hand, a long term prediction may reveal the need for a certain agent to be in a particular internal state in order to the deciding agent achieve a specific goal that relies on other agent state. Therefore, the agent generates a sub-goal that results on the other agent achieving that internal state; if that state is positive for the other agent, this process may be seen as cooperation, otherwise it will be considered competition.

At the market implementation, dialogs are initiated in order to negotiate the price of a product. These dialogs

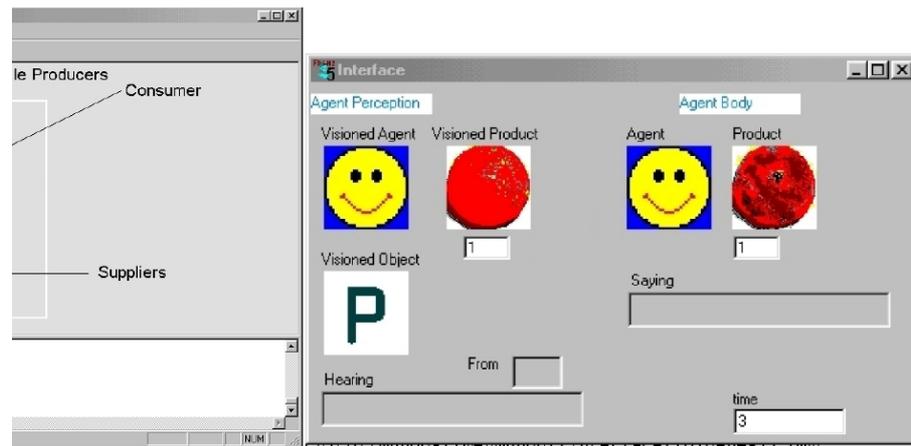


Figure 5: Initial configuration of the environment and Agent interface.

intend to avoid the punishment for stealing and to satisfy the agent needs (for nutrients in the consumer and money in the supplier). When the supplier does not have the product needed by the consumer, but there is a producer who has it, the consumer places an order for the supplier to get the product given an arranged price. To pick that product will be a goal for the supplier in order to satisfy the consumer and its own need for money (cooperation).

The symbolic selected action is never negative for the individual survival (simple internal state) and should improve individual quality of life (extended internal state). After an action being executed, the internal state and memory are updated allowing the progress of learning and adaptation.

3 Results

The tests to the market implementation were made in order to evaluate the contribution of each component of the architecture for the overall performance of the agents in the environment. The first set of tests were made only to the cognitive and perceptual layer. The goal was to determine the degree of survival related to the nutrients in the internal state. The initial configuration of the environment⁵ is presented in figure 5 (right) by an image of the application interface, on the left is an image of the agent interface with the user. The results on the next plots refer to 20 simulations of 500 server cycles of 5 seconds, in each cycle agents receive new visual stimuli and several auditive stimuli and they can execute more than one action, these factors depend on the server performance at each time. The architecture parameters, referred in the previous section, were set to $\sigma_p = 150$, $\sigma_c = 100$, $\tau = 20$ and $\eta_n = 5$.

Using only the perceptual layer, without any adaptation, the results showed that agents were unable to distinguish good products from bad products, ending up to eat everything, which lead to poor survival capability. Figure 6

⁵The environment used is an adaptation of RoboCup soccer server code (6).

presents the plot of the mean of maximum unbalance obtained in the simulations using only the perceptual layer with and without adaptation. The ideal plot would be one with lines near 0 and an acceptable one would have lines below 150 (given the perceptual urgency threshold σ_p).

The perceptual layer with adaptation reveals an improvement on survival capability (figure 6) and the observed behaviours showed that agents select more often products with the same predominant colours of products that recently had a positive effect on the internal state balance. Due to the limited information provided by predominant colour, the agents often select products that show different effects when compared with the one recently experimented. This lead either to the selection of an apparently good product which in fact is not or to losing the chance of experimenting a product potentially good.

The results obtained by adding the cognitive layer to the adaptive perceptual layer are shown in figure 7. The unbalance reduces to an acceptable level. Initially, agents select actions using the perceptual layer, but as soon as memory starts to be filled up the actions selected become more adequate given previous experiences. The cognitively selected products are the ones that achieve best internal state. The perceptual layer acts effectively whenever urgency is detected, allowing agents to return to an internal state where the cognitive layer is no longer inhibited. Since both layers, perceptual and cognitive, do not process the money element agents always steal to satisfy their needs.

A second set of tests were performed in order to test the full architecture, *i.e.*, the perceptual (adaptive), cognitive and symbolic layers. These tests had different configurations so that the analysis of results could be focused more on the behaviours produced by the symbolic layer and less in the survival capabilities of lower layers. These configurations had less agents and more products and the results described next are not numerically based but rather behaviour based.

The tests with the symbolic layer revealed that dialog actions of negotiation were selected whenever the cogni-

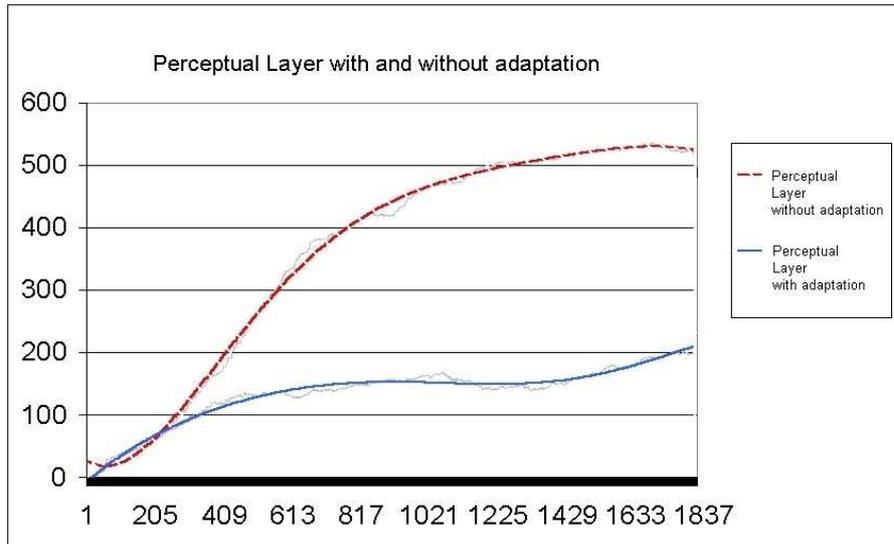


Figure 6: Mean of the maximum unbalance obtained using only the Perceptual layer, with and without adaptation. Horizontal axis represents each time an action is executed and the internal state is changed.

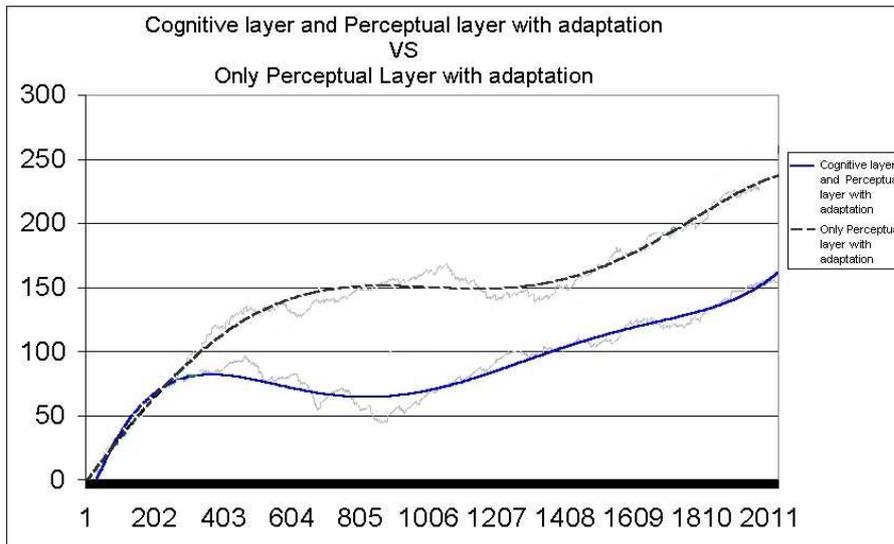


Figure 7: Mean of the maximum unbalance obtained using only the Perceptual layer with adaptation compared to the addition of Cognitive layer. Horizontal axis represents each time an action is executed and the internal state is changed.

tive layer selects to eat a product. The symbolic layer retrieves from memory the money punishment and starts the negotiation. Initially the dialogs are built-in, as requests and answers, but after the first success they are retrieved from memory and adapted to the new situation (new product, supplier and price). The retrieved dialogs may have been experienced by the agent itself or by others observed and listened by it. Dialogs are initiated if they are marked in memory by a positive change of the agent expression, considering money and nutrients. When a dialog is recalled, the agent chooses the actions executed by a similar agent which had a positive change of expression.

The symbolic layer revealed the capability to balance the extended internal state, not only the money element but also the nutrients. For instance, when a consumer agent has no more money (this agent has no means to earn money, only suppliers have it), the negotiation never succeeds, leading to the selection of an eat action (stealing) at the symbolic layer, as it does at the cognitive layer. This means that the agent chooses to be punished because the expression change is positive in the nutrient feature and is equally negative on the money element. This behaviour is considered adequate because there is no better option at that given situation.

The tests also showed the adequacy of consumer orders. A consumer requests a product to a supplier. When the supplier has the product, it might sell it, otherwise the supplier might accept an order to get it from a producer or another supplier. It is assumed in this implementation that consumer agents can not interact with producers. The consumer and supplier negotiate the price, and once they reach an agreement, the supplier generates a subgoal in order to find and pick that specific product. What happens is that the supplier recalls in memory an internal state that was satisfied by that product, and assumes it as a virtual current internal state, deciding as if it was needing it. Once the product is picked by the supplier, the consumer approaches and eats it, paying the amount of money previously arranged. If during the order satisfying process, the supplier has a need for nutrients the process is suspended until it is no longer unbalanced. This interruption derives from the cognitive urgency inhibition of the symbolic layer. After satisfying its needs, the supplier resumes the process of getting the product ordered.

4 Conclusion

The tests performed to the market implementation showed that the adaptive perceptual layer alone allowed the agents to distinguish good products from bad products based on recent experience of eating them. The perceptual adaptation is based only on the relevant features, which are not very informative, resulting on frequent mistakes. With the cognitive layer it was obtained adequate behaviour given the environment, *i.e.*, the existing products in the world. Agents choose, by recalling past experiences, the best prod-

uct for its current internal state. Along with the increase of experience the decisions increase in adequacy, in what nutrients concern. The symbolic layer uses the notion of sympathy for decision-making purposes, introduces communication, learning from observation, and explicit goals, maintaining the survival capability.

Overall, the extension of the *DARE* architecture presented here revealed an interesting performance in the dynamic multi-agent environment where it was tested, showing similar capabilities on individual decision-making, flexibility and learning as its previous version, and new abilities to model the social role of emotions. In what concerns future work, this architecture could be improved on two aspects: i) allow agents to anticipate action effects on a long range basis, and ii) incorporate rational (logical) inference mechanisms.

References

- [1] M. Aubé. A commitment theory of emotions. In D. Cañamero, editor, *Emotional and Intelligent: the Tangled Knot of Cognition. 1998 AAAI Fall Symposium Technical Report FS-98-03.*, pages 13–18. AAAI, 1998.
- [2] D. Cañamero and Walter Van de Velde. Socially emotional: using emotions to ground social interaction. In Kerstin Dautenhahn, editor, *Human Cognition and Social Agent Technology*. John Benjamins Publishing Company, 1997.
- [3] António Damásio. *Descartes' Error: Emotion, Reason and the Human Brain*. Picador, 1994.
- [4] N. Frijda. *The Emotions*. Cambridge University Press, 1986.
- [5] D. Goleman. *Emotional Intelligence*. Bloomsbury, 1996.
- [6] Noda Itsuki, Peter Stone, Emiel Corten, and et al. Soccerserver manual - ver.5 rev.00 beta. Technical report, July 1999.
- [7] William James. What is an emotion?. first published in mind, 9, 188-205. 1884.
- [8] Joseph LeDoux. *The Emotional Brain*. Simon and Schuster, 1996.
- [9] Márcia Maças, Rodrigo Ventura, Luis Custódio, and Carlos Pinto-Ferreira. Experiments with an emotion-based agent using the dare architecture. In *Proceedings of the AISB'01 Symposium on Emotion, cognition, and affective computing*, pages 105–112, March 2001.
- [10] Paolo Petta. Principled generation of expressive behavior in an interactive exhibit. In J.D. Velasquez, editor, *Workshop: Emotion-Based Agent Architectures*

- (EBAA'99), *Third International Conference on Autonomous Agents (Agents '99)*, pages 94–98. Seattle, WA, 1999.
- [11] Rui Sadio, Gonçalo Tavares, Rodrigo Ventura, and Luis Custódio. An emotion-based agent architecture application with real robots. In *Emotional and Intelligent II: The Tangled Knot of Social Cognition – AAI Fall Symposium*, 2001.
- [12] Alexander Staller and Paolo Petta. Introducing emotions into the computational study of social norms. In *Proceedings of the 2000 Convention of the Society for the Study of Artificial Intelligence and the Simulation of Behaviour (AISB'00)*, pages 17–20. 2000.
- [13] Kenneth T. Strongman. *The Psychology of Emotion – Fourth Edition*. John Wiley and Sons, 1996.
- [14] Pedro Vale and Luis Custódio. Learning individual basic skills using an emotion-based architecture. In *Proceedings of the AISB'01 Symposium on Emotion, cognition, and affective computing*, pages 105–112, March 2001.
- [15] Rodrigo Ventura, Luís Custódio, and Carlos Pinto-Ferreira. Artificial emotions - goodbye mr. spock! In *Proceedings of the 2nd Int. Conf. on Cognitive Science*, pages 938–941, 1998.
- [16] Rodrigo Ventura, Luís Custódio, and Carlos Pinto-Ferreira. Emotions - the missing link? In D. Camero, editor, *Emotional and Intelligent: the Tangled Knot of Cognition. 1998 AAI Fall Symposium Technical Report FS-98-03.*, pages 170–175. AAI, 1998.

Timidity: A Useful Emotional Mechanism for Robot Control?

Mark Neal
 Department of Computer Science
 University of Wales, Aberystwyth. UK.
 mjn@aber.ac.uk

Jon Timmis
 Computing Laboratory
 University of Kent, Canterbury. UK.
 J.Timmis@kent.ac.uk

Keywords: endocrine system, artificial endocrine systems, neural networks, robot behaviour, emotion, perception, homeostasis

Received: October 28, 2002

Responses labelled as emotional in the higher animals are frequently portrayed as incidental to the generation of reasonable behavior. Clearly this view is incompatible with the reality of animal behavior as observed in nature, emotion plays a significant role in the generation of useful behaviour. Homeostasis is the product of the interaction of the nervous, endocrine and immune systems. This work views emotional responses as part of an integrated approach to the generation of behavior in artificial organisms via mechanisms inspired by homeostasis. The mechanism presented here employs the concept of a novel Artificial Endocrine System which interacts with an Artificial Neural Network to generate behaviour which could be classified as emotive.

1 Introduction

The quest for more effective techniques in the implementation of intelligent systems has lead to the consideration of many mechanisms of both human and non-human cognition as potential models for behaviour generation. Artificial systems which attempt to capitalise on the use of emotional state information in the generation of behaviour have usually concentrated on symbolic representations and manipulation at the level of beliefs, desires and actions (12). Much sub-symbolic work has assumed that the emotive state of a system is an emergent property rather than a controlling factor (2). This work proposes a sub-symbolic mechanism for the explicit representation of emotive states as hormone concentrations which modify the behaviour of an artificial neural network (ANN). This mechanism is viewed as part of a more ambitious and general approach to the generation of intelligent behaviour centred around the biological concept of homeostasis (16).

The generation of intelligent behaviour in robotic systems has been a goal for many years. Some major advances have been made by borrowing ideas from biological organisms, but the generation of systems which can remain independent from human intervention for long periods of time is still largely unfulfilled. The applications for such systems grow more demanding with time and cry out for novel approaches to long term control of autonomous behaviour.

2 Biological Background

It is necessary to spend a little time examining the biological motivation for this work. To this end, the paper will explore (at a high level) the biological systems that help to maintain homeostasis within an organism. From there, it will be possible to move into the artificial domain, adopting useful metaphors from the biological system.

2.1 Mechanisms for Biological Homeostasis

Homeostasis is the ability of an organism to achieve a steady state of internal body function in a varying environment. Homeostasis is achieved via complex interactions between a number of processes and systems within organisms, namely the nervous system; the endocrine system and the immune system. By examining these systems and their interactions, we hope to emulate aspects of this behavior in artificial systems.

2.1.1 The Nervous System

The nervous system (NS) is central to an organism's ability to process and act upon stimuli that it receives from an external source. Organisms ranging from slugs to humans are endowed with a nervous system which ranges in size, ability and function. This system will then develop and improve over the lifetime of the organism, via processes such as learning and memory (although not exclusively these).

An organism will be exposed to a vast number of stimuli, to which it must react. Simply put, the NS will take sensory input and generate effector output. The sensory parts of the NS take input from vision, taste etc., which are stimuli for effector elements such as muscles. The processes seen in the NS have inspired artificial systems (artificial neural networks)(7) which form an integral part of the work proposed in this paper. The interaction between the NS and the endocrine system is complex and incompletely understood, but for the purposes of this work a simplistic model of neural stimulation and inhibition by hormones is employed.

2.1.2 The Endocrine System

Within an organism, chemicals known as hormones implement a regulatory mechanism acting directly at an individual cell level. This system, the endocrine system, is responsible for the production and storage of these chemicals (15). Hormones are also produced by neurons and immune cells such as T-cells, but for the current purposes these mechanisms will be ignored. These hormones have a great deal of influence over a large number of bodily functions and are key actors in the maintenance of homeostasis. Hormones have many functions which affect behavior, assist growth, drive reproduction and so on. Typically, production of a hormone is in response to a change in state of the organism. Such changes are detected via the nervous system, immune system or by changes in other hormone or metabolite levels. Hormones are released into the blood or lymph system and are able to reach virtually all the tissues within the organism. It is quite possible (and normal) that there will be a number of different hormones present in the blood or lymph at any one time. However, not all cells will react to all hormones, as the response to hormones is highly specific: only certain cells are capable of responding to certain hormones. When a hormone locates its particular target cell, a binding takes place through specific receptors on the cells. Receptors on the target cell are usually located in one of two sites: within the cell nucleus (steroid hormone receptors) or in the plasma membrane (non-steroid hormone receptors, e.g., proteins, amines, and peptides). Non-steroid hormones decay and are ultimately removed from the organism at various rates. Built into the system is a mechanism by which hormones such as these will decay. This decay rate may well be a few minutes, but could potentially be a number of days. When a hormone binds with a receptor on the cell membrane, it stimulates internal signals to the appropriate sites within the cell, which in turn alter the cell's activity. For this work the only mechanism for hormone production which is considered is the change of external environment inducing the production of hormone analogous to short-lived non-steroidal substances which affect neuro-transmission.

2.1.3 The Immune System

The immune system is a remarkable, but complex, natural defence mechanism, which responds to foreign in-

vaders called pathogens. Organisms typically have two lines of immunity, innate (inherited at birth) and adaptive (also known as acquired) which develops over the lifetime of the organism (14). Pathogens are first attacked by the innate immune system, and if this defence by the innate immune system fails, then the pathogen is passed over to the adaptive immune system. The adaptive immune system primarily consists of B- and T-lymphocytes (cells). Through receptors on the cell, they are capable of binding with pathogenic material (antigens). Whilst the immune system is integral to the achievement of homeostasis, for the purpose of work presented in this paper, discussion of immune operations will not be considered further, as no attempt has been made at this point to integrate an AIS into the device.

2.2 Interactions between Biological Systems

So far, attention has been given to three systems within an organism: the nervous system, endocrine system and immune system. These systems do not act independently but as one large and complex system.

Work in (3) examined the mechanisms by which these three systems interact and can be summarised as follows (for a more in-depth analysis's see (5)): Immune, neural and endocrine cells can express receptors for each other and products from immune, neural and endocrine systems can exist in lymphoid, endocrine and neural tissues. This allows for interaction and communication between cells and molecules all three ways. The action of various endocrine products on the neural system is accepted to be an important stimulus of a wide variety of behaviors. These range from behaviours such as flight and sexual activity to sleeping and eating.

3 A Framework for Artificial Homeostasis

The concept of a *framework* is often employed when attempting to construct complex systems such as these. A framework could be said to consist of building blocks, which when combined, form a complete system, and indeed work in (5) proposed a potential framework for Artificial Immune Systems (AIS), and made allusions to the fact that Artificial Neural Networks could also be thought of in such a way. The authors argued that a framework would consist of (1) a representation of the components of the system (2) mechanisms by which to evaluate interactions of these components and (3) procedures for adaptation. Under such a conceptualisation, it is easier to discuss how such systems may be combined to form a more complex system. Table 1 captures the salient features of this argument, with the addition of the Artificial Endocrine System (AES). It is proposed that combinations of these components, will be useful in the construction of systems capable of artificial

	ANN	AES	AIS
(1)	Neuron	Endocrine gland	Lymphocyte
(2)	Network topology	Hormone interactions	Affinity measures
(3)	Learning algorithms	Hormone structure update	Immune algorithms

Table 1: ANN, AES and AIS in a simple framework, see text for definition of (1), (2) and (3)

homeostasis. Work in this paper is restricted to the use of a combination of ANN and AES.

3.1 Artificial Counterparts of the Biological Systems

Significant work has been done in extracting useful metaphors from the nervous system for the creation of artificial neural networks (7). Work is now emerging in the field of AIS (5), but little has been done on AES. This section will discuss ANN and AIS and postulate that through the combination of these approaches and an AES it may be possible to create an artificially homeostatic system. Work in (5) describes some of these ideas, and the reader is directed to there for further detail on interactions of both the biological and artificial systems.

3.1.1 Neural Networks

A substantial body of research has been undertaken in extracting useful metaphors from the neural systems. Artificial Neural Networks are parallel distributed processing systems that are constructed via the connection of simple processing known as artificial neurons (7). ANN have been applied to a vast array of problem areas such as machine vision (11) and robot control (10). Figure 1 is a graphical depiction of a simple artificial neuron. In order to be of any practical use, individual neurons are connected together to form artificial neural networks. These networks are *trained* in order to be able to classify input patterns (x) through the constant adjusting of the weights (w_i) until the ANN can recognise the pattern. The weights are adjusted via a number of possible *learning algorithms* e.g. backpropagation. An artificial neuron can be represented mathematically as shown in Equation 1. Once the summing of the inputs has taken place, the neuron will fire, depending on the activation function $f(u)$, in this case of work in this paper a standard sigmoidal activation function has been employed, as shown in equation 2.

$$u = \sum_{i=0}^{n \cdot x} w_i \cdot x_i \quad (1)$$

$$f(u) = \frac{1}{1 + e^{-u}} \quad (2)$$

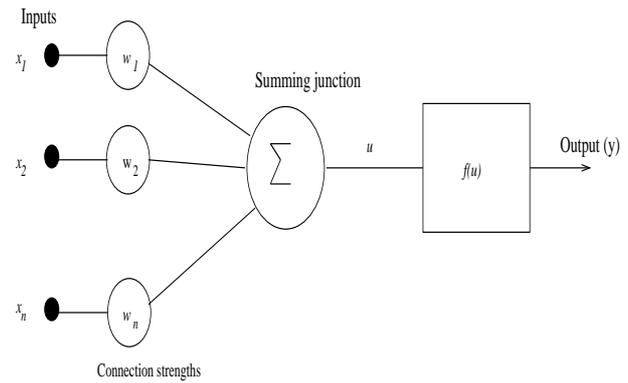


Figure 1: A simple artificial neuron

Work in this paper proposes to augment this basic artificial neuron, with interactions from an artificial endocrine system. For the purposes of this work, the weights within the ANN are constant, although for future work, this will not be the case.

3.1.2 Artificial Endocrine Systems

This paper proposes a new biologically inspired technique known as an Artificial Endocrine System. The role of the AES is to provide a long term regulatory control mechanism for the behaviour of the system. The AES proposed consists of *gland cells* which secrete *hormones* in response to external stimuli, to the value r_g for one gland g . This is shown in equation 3 where α_g is the rate at which hormones are released for a particular gland g .

$$r_g = \alpha_g \sum_{i=0}^{n \cdot x} x_i \quad (3)$$

The level of hormone is subject to geometric decay, as shown in equation 4 where $c(t)_g$ is the hormone concentration at a time t for a gland g and β is the decay constant.

$$c(t+1)_g = (c(t)_g \cdot \beta) + r_g \quad (4)$$

Membrane receptors located on artificial neurons are sensitive to hormones, thus providing a mechanism for the regulation of the ANN by the AES. Gland cells secrete and record the concentration of hormones present in the system. Each gland cell secretes a specific hormone, represented by a simple string of bits. Within the integrated AES-ANN the hormone sensitive membranes of neurons simply have a list of hormone receptors (again, represented as bit patterns) to which hormones are matched and a neuron-specific action associated with each receptor. At present, perfect matches of hormone to receptor are considered (though this is not necessarily required: imperfect matches should generate lesser reactions). In the natural endocrine system, hormones are transported throughout the body: the same effect is achieved in the AES through the matching of each hormone secreted to the receptors on each cell's membrane

in turn. A record of the current concentration of a hormone is maintained in the gland cell which secretes the hormone, and is then used to moderate the strength of reaction.

True to the analogy with the biological endocrine system, different cells types react to particular hormones, in different ways. The actions which are triggered in individual cells can vary according to four factors: the hormone which is detected, its concentration, the type of receiving cell and the individual cell's make-up. The former two of these factors are explained above, but the latter require further explanation. The type of cell receiving the hormone signal will clearly dictate what actions it is capable of performing. For example, a neural cell may lower (or raise) its threshold value or increase (or decrease) its sensitivity to one or many of its inputs; and a gland cell may increase (or decrease) secretion rate of a hormone. The precise make-up of cells is fixed when they are added to the system. This may include variations in membrane characteristics (abilities to receive hormone signals), the effects that those signals have within the cell and other cell-type-specific characteristics such as connectivity pattern of a neuron etc.

In order to allow for the AES-ANN interactions, the hormone levels have to be able to affect the input weights in the ANN. Figure 2 provides a simple graphical representation of how this is achieved. Here the recorded hormone level affects each input weigh on a particular neuron. It is easier to see this when these interactions are described mathematically, as in equation 5, where in this case x_i and w_i are the same as equation 1 and ng is the number of glands in the system, c is the concentration of hormone, S is the sensitivity of the connection for receptor i to hormone j and M is the match between the receptor i and hormone j and is defined in equation 6, where dis is a distance measure function.

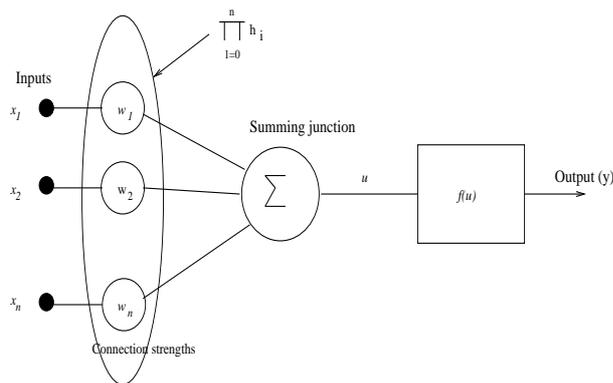


Figure 2: The affect of endocrine interaction on the artificial neuron

$$u = \sum_{i=0}^{n.x} w_i \cdot x_i \cdot \prod_{j=0}^{n.g} c_j \cdot S_{ij} \cdot M_{ij} \quad (5)$$

$$M = \frac{1}{1 + dis(i, j)} \quad (6)$$

It is now possible to compare equation 1 with equation 5. It should be noted that the new equation for the AES-ANN interaction is a simple augmentation of the original equation, with the iterative application of hormone levels applied to each input weight in the neuron. Whilst adjusting the activation function for ANNs has been explored before, adding hormonal interactions as described above is novel and fundamentally different. The decay constant of the hormone which mediates neuron activity over short/medium timescales means that the ANN-AES is no longer limited to immediately reactive responses. It should also be noted that this new AES Neuron bares a passing resemblance to the *Sigma Pi Neurons* (9), it is fundamentally different upon further examination.

3.1.3 Artificial Immune Systems

AIS is very much an emerging area of biologically inspired computation. This insight into the immune system has led to an ever increasing body of research in a wide variety of domains such as machine learning (13), immunised fault tolerance (4) and computer security (6) to name a few. Recently, an attempt has been made to bring together what at times seemed a disparate area of research, in a general AIS framework which describes basic AIS components, interactions and algorithms (5). Here the authors argued that AIS could be seen as a novel soft computing paradigm that has great potential to be hybridised with a variety of other soft computing approaches and computational intelligence paradigms.

It is anticipated that in future work, an AIS will be integrated into the mechanism, described here and will interact with both the ANN and AES to maintain homeostasis.

4 Target Application for ANN-AES Interaction

In order to test some of the mechanisms proposed here, a simple artificial neural system and artificial endocrine system were implemented. The chosen application was a controller for a mobile robot in an office environment. The robot (a Pioneer 2DX¹) is equipped with 16 sonar sensors arranged around its perimeter, which are capable of detecting objects reliably up to about 5 metres away. A simple neural network was initially set up manually to link these sensors to the motors which drive the wheels. The network generates simple object avoidance behaviour which works effectively in a static environment. Network weights were chosen manually and adjusted after experimentation to maintain reasonable clearance when approaching and avoiding objects. The network was supplied with a bias node which was used to ensure that there was sufficient activity in the output nodes to generate forward motion when there was no stimulation of any of the sonar sensors. The resultant behaviour observed was “wandering” whilst

¹<http://robots.activmedia.com/>

avoiding walls and other objects. Typically when a wall was approached, the robot would gradually turn by an angle close to 180 degrees and move off in a straight line until it encountered the next obstacle.

This behaviour became inadequate when dealing with a highly populated environment. The robot became very close to some objects and upon occasions collided with them. A “more cautious” set of weights for the neural network would have eliminated the problem, but would then have made the behaviour unnecessarily cautious in less cluttered environments. For some tasks such as automated floor cleaning or map building this would be undesirable as it would leave larger than necessary regions of the environment unexplored. Thus, the alteration of the distance to which the robot would approach obstacles became a candidate for the application of hormonal control.

4.1 Mechanisms for Implementation

The ANN and AES employed in the implementation of the robot controller are as described in sections 3.1.1 and 3.1.2 respectively. In this implementation, which was designed to investigate the functionality of the AES, there was no weight update mechanism (learning) employed in the neural network.

The controller was compiled and run on a PC104 embedded personal computer (running RedHat Linux) physically mounted inside the robot itself. The PC104 board uses a serial connection to communicate with a low-level microcontroller which drives the motors and services the sonar sensors employed. Communication with the embedded PC104 is via a radio-ethernet link.

Figure 3 provides a graphical illustration of the implemented system. The ANN is fully connected and no weight adaptation mechanism is employed at this stage. Similarly, hormones and receptors are fixed in length and content. Artificial neurons receive input from sensors, as does the artificial endocrine gland. When a stimulus is encountered, the artificial endocrine gland excites or inhibits each synapse in the ANN (as illustrated by the shaded area) via the hormone release mechanism, see equation 5. The hormone (which might be seen as analogous to a hormone such as adrenalin) was excitatory to all synapses (or weights). The activation level of the two output neurons is then used to drive the motors directly.

4.1.1 Experimental Setup

The robot was run in the Intelligent Systems Laboratory at Aberystwyth in a relatively constant, but uncontrolled environment. The robot was placed in the same start position and orientation for each run. The start point was at the centre of an approximately square region of floor (approximately 6 metres across) bounded on two sides by desks and chairs, on one side by a wall and on the fourth side by a wall with narrow openings at each end which lead into short cul-de-sac corridors.

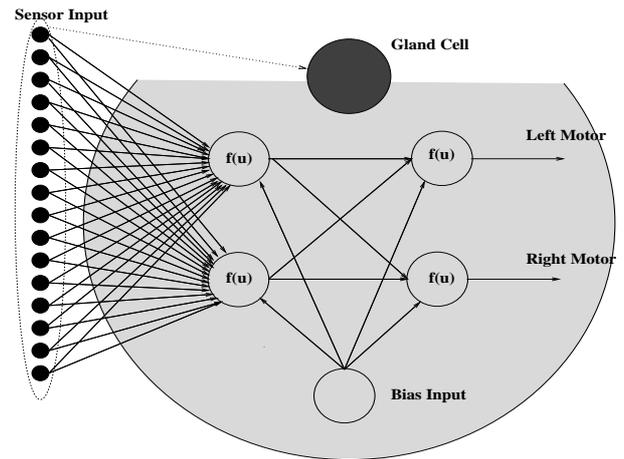


Figure 3: The Artificial Neural Network augmented with the Artificial Endocrine Gland. There are 16 sensor inputs, 2 hidden layer nodes and 2 output nodes. Each output node controls either the left motor or right motor control. The shaded area indicates that the hormonal gland influences the neurons in the network

4.1.2 Experimental Method and Aims

In order to examine the effect that the endocrine interactions were having on the behaviour of the robot, a series of experiments were performed. The robot was run for 22 minutes at a time after which results were downloaded and the robot returned to the start position.

Three different versions of the controller were used. The first was the pure ANN implementation in which the hormone was not allowed to interact with synapses at all. This is intended as a control. A second version was given a fixed level of hormone throughout the experiment and was used to ensure that the hormone was having the intended effect on the behaviour of the robot. The third version was given a complete implementation of the AES in which the endocrine cell was connected to the sensors and released more hormone when the sensors were more stimulated (ie. obstacles were closer). Each experiment was run 5 times in order to investigate the variability of performance and to lend statistical weight to the results.

Thus the aim of these experiments was to show that:

1. Higher hormone levels result in a more expeditious retreat from obstacles
2. The variable hormone release mechanism is effective in allowing both close approaches and expeditious retreats

4.1.3 Results

Table 4.1.3 shows results from all fifteen experiments performed. The activity of a sensor is calculated as follows:

$$sa = 1 - dist \quad (7)$$

where sa is the sensor activity as presented to the ANN and AES, and $dist$ is the distance (in metres) to the nearest obstacle as detected by that sensor. Due to the small size of the space in which the robot was operating any objects more than one metre away were ignored.

The table shows the summed activity of the most active sensor over complete runs. It is clear from the values in the table that when equipped with the controller with no hormone present the robot spends much more time in close proximity to obstacles than when the controller either has a fixed high level of hormone or a variable level of hormone. It is also clear that the values observed are (despite some variability between repetitions) significantly different under the different regimes.

Thus by examining the first two columns of the table we can confidently assert that higher hormone levels do result in a more expeditious retreat from obstacles.

<i>Integrals of sensor activities</i>			
Hormone Regime	Zero	Fixed	Varying
	8217	3153	4617
	7435	3083	5066
	9576	4495	4714
	14421	2514	5023
	10211	4439	4509
Average	9972	3537	4786
St. Dev.	2717	885	247

Table 2: Integrals of sensor activities under various hormone regimes

Figure 4 shows the activity of the most active sensor at the beginning of three runs (one with each version of the controller). The first approach to the wall, represented by the first increase in sensor activity, clearly shows the differences between the behaviour induced by the three controllers. The controller with no hormone present spends three times as long in range of the obstacle as the fixed hormone level controller, and a little more than twice as long within range than the variable hormone controller. Of most interest however, is that the controller with a variable hormone will approach obstacles to a similar distance, when compared to the controller without any hormone. It then will beat a hasty retreat. This seems to provide evidence that the second aim, as stated above, is also achieved.

Figure 5 shows the hormone level from a specimen run of the robot under the variable hormone regime. The release of hormone as the robot approaches obstacles can clearly be seen as marked “spikes” in the upper trace. The region marked as “negotiating corridor” marks a period of time in which the robot entered one of the cul-de-sac corridors, manoeuvred to the end of it turned around and moved out of it again. The close proximity of the walls whilst the robot is in the corridor maintains a raised level of hormone for a prolonged period during which the robot avoids approaching walls too closely. Figure 6 shows the path taken by the robot on this occasion (only the period

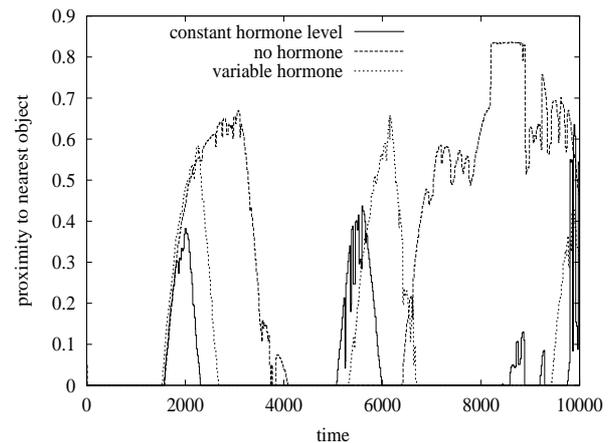


Figure 4: Example sensor readings under three different hormone regimes

between entering the corridor and leaving the corridor is shown). It can be seen that the trajectory taken by the robot under a variable hormone regime covers about 1 metre of the corridor’s width. Figure 7 shows a similar portion of the robot’s trajectory whilst under control of the fixed hormone level controller. On this occasion it can be seen that the trajectory covers only about 70cm of the width of the corridor. The skewed nature of the trajectory with respect to the walls as drawn (especially in figure 7) is due to wheel slippage allowing errors to build up in the dead-reckoning used to keep track of the robot position. Future work will be undertaken using a motion tracking system to eliminate such problems.

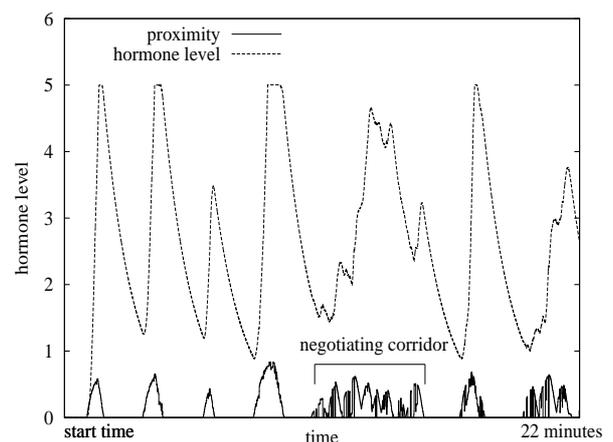


Figure 5: Recorded hormone level for an example robot run

4.2 “Emotional” Response

Informal experimentation with the robot (whilst running the variable hormone level controller) was carried out

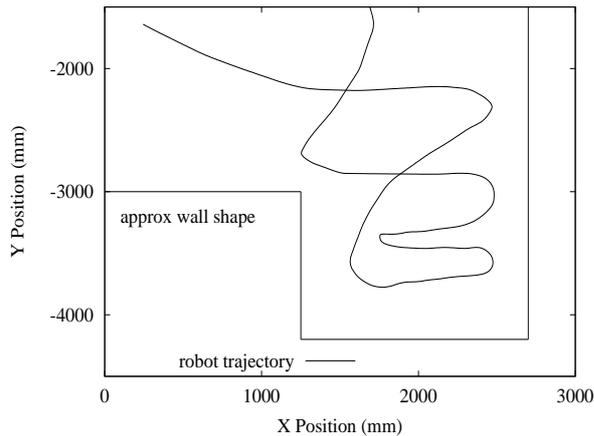


Figure 6: Negotiating a corridor under the variable hormone regime

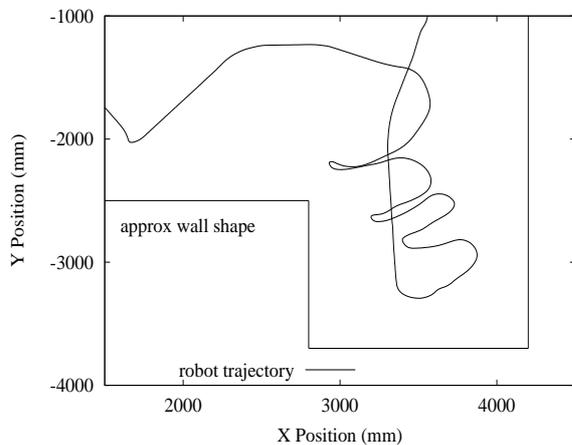


Figure 7: Negotiating a corridor under the fixed hormone regime

which involved "trapping" the robot between the arms by blocking the sensors at very close range. This initially caused the robot to stop moving forward and to turn around "looking for a way out". If the trapping was continued for a long period of time the robot gradually became more and more rapid in its movements and eventually it would spin at maximum speed on the spot. As soon as the blocking of the sensors stopped it would head away from the torturer on an erratic path into an open space. Pursuing the robot or obstructing its path whilst the hormone level was still very high resulted in very violent turns away from the obstruction. Over a period of a couple of minutes the robot would "calm down" as the hormone level reduced. Presentation of these results is extremely difficult in anything approaching a scientific manner, and until the authors have found a way to do so we will not attempt more than an informal description such as this.

It is worth noting however that people's reactions to this

"torturing" of the robot are interesting in their own right. Requests to "leave the poor thing alone" and other such comments are not uncommon. Indeed it is surprising how people are very willing to project emotions onto a small autonomous robot which exhibits even very rudimentary displays of "distress" and "fear". The notion of projecting emotional states onto the robot is one with which the authors are a little uncomfortable, but we feel it is our duty to put our philosophical/psychological stance on display. Of the various philosophical theories of emotion we (in broad agreement with (8)) identify three plausible candidates at a level appropriate for consideration with respect to this work. Versions of an epiphenomenal theory which are often attributed to Hume and Descartes seem to rely on self-awareness and the ability to report *feelings* and thus lack credibility in this case due to the level of intelligent reporting and introspection that they seem to require. Such theories also lack a causal route between the emotions and actions which would seem to be fundamental to the attribution of emotional state in mechanisms as simple as that embodied in our robot. A theory which is arguably more suited to the mechanisms present in our robot is one that was put forward as a part of the (now largely disregarded) behaviorist stance. This theory states that an emotion is "*an hereditary 'pattern-reaction' involving profound changes of the bodily mechanism as a whole, but particularly of the visceral and glandular systems*"(17). This biological level of description is appealing, but there are several problems with the repercussions of the *hereditary* qualifier which makes this a particularly weak candidate theory (see (8) for discussion). The third (and favoured) theory is a version of a cognitive theory originating from the Greeks (and especially Aristotle(1)), which proposes that emotions are evoked in response to beliefs about the state of the world and possible events which may be caused by that state in the future. Thus whilst we are not entirely comfortable about assigning the word *belief* to the state of the neurons in the robot when presented with obstacles in the world, this is the point at which we feel most comfortable about making a stand. Thus we should say that the control system as a whole (through the belief that an object close to the robot presents a risk of collision) achieves a level of fear via the response of the gland cell releasing the hormone. This modifies the behaviour of the robot in a way consistent with "fear" which in turn results in timid behaviour. Thus the "fearful" state of the robot is caused by and displayed by the internal state of the robot and its interactions with the environment. We do not propose that this is the only or even the best philosophical explanation of how the controller works, but believe that this explanation captures the spirit in which the controller was constructed.

The results as presented above show a potentially useful application of an artificial endocrine mechanism in moderating interactions with obstructions in the environment of a mobile robot. Ascribing an emotion such as "fear" or "timidity" to this mechanism seems like a reasonable approach to describing the behaviour that is generated. Al-

though the mechanism used is extremely simple, the behaviour generated is both functional and emotively appealing.

5 Conclusions

A mechanism for the integration of ANN and AES models has been presented. This is set in a wider context with respect to homeostatic mechanisms. An example implementation of a robot controller which embodies this mechanism is detailed and results demonstrating its behaviour and performance are presented. The behaviour shows traits which may be useful in exploratory behaviour, especially in varying environments. The behaviour is also appealing subjectively and elicits responses from onlookers who are willing to ascribe emotional states to the robot.

Future work will involve use of precision motion tracking software and more complex controllers involving more than one hormone. Research into more complete models of artificial homeostasis is also on-going.

References

- [1] Aristotle. *The Basic Works of Aristotle*, chapter Rhetoric. Random House, 1941.
- [2] R Aylett. Emotion in Behavioural Architectures. In *Workshop on Emotion-Based Agent Architectures*, Seattle, USA, 1999.
- [3] H O Besendovsky and A del Ray. Immune-Neuro-Endocrine Interactions: Facts and Hypotheses. *Nature*, 249:356–358, 1996.
- [4] D Bradley and A M Tyrell. A Hardware Immune System for Benchmark State Machine Error Detection. In *Congress on Evolutionary Computation. Part of the World Congress on Computational Intelligence*, pages 813–818, Honolulu, HI., 2002. IEEE.
- [5] L N de Castro and J Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag, 2002.
- [6] S Forrest, S Hofmeyr, and A Somayaji. Computer Immunology. *Communications of the ACM*, 40(10):88–96, 1997.
- [7] S Haykin. *Neural Networks - A Comprehensive Foundation*. Prentice-Hall, 1999.
- [8] W Lyons. *Emotion*, pages 1–52. Cambridge University Press, 1980.
- [9] D E Rumelhart J McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, Cambridge, Massachusetts, 1986.
- [10] E Oyama, N Y Chong, A Agah, T Maeda, and S Tachi. Inverse Kinematics Learning by Modular Architecture Neural Networks with Performance Prediction Networks. In *IEEE International Conference on Robotics and Automation*, pages 1006–1012, 2001.
- [11] G Sandini, F Bosero, F Bottino, and C Ceccherini. The Use of Anthropomorphic Visual Sensor for Motion Estimation and Object Tracking. Proc. OSA Topical Meeting on Image Understanding and Machine Vision. *IEEE Transactions on Neural Networks*, 1(1):28–43, 1989.
- [12] A Sloman. *The Philosophy of Artificial Intelligence*, chapter Motives, Mechanisms and Emotions, pages 231–247. Oxford, 1990.
- [13] J Timmis and M Neal. A Resource Limited Artificial Immune System. *Knowledge Based Systems*, 14(3/4):121–130, 2001.
- [14] I Tizzard. *Immunology : An Introduction*, chapter The Response of B Cells to Antigen, pages 199–223. Saunders College Publishing, 2nd edition, 1988.
- [15] A J Vander, J Sherman, and D Luciano. *Human Physiology: The Mechanisms of Body Function*. McGraw-Hill, 5th edition, 1990.
- [16] F J Varela. *Self Organising Systems*, chapter Autonomy and Autopoiesis, pages 14–23. New York Campus Press, 1981.
- [17] J B Watson. *Psychology from the Standpoint of a Behaviorist*. Lippincott, 1919.

Emotional Information Retrieval for a Dialogue Agent

Rafal Rzepka and Kenji Araki

Graduate School of Engineering, Hokkaido University

Kita-ku, Kita 13-jo Nishi 8-chome, 060-8628 Sapporo, Japan

{kabura, araki}@media.eng.hokudai.ac.jp http://sig.media.eng.hokudai.ac.jp/index_e.html

Koji Tochinai

Graduate School of Business Administration, Hokkai-Gakuen University

Toyohira-ku, Asahi-machi 4-1-40, 062-8605 Sapporo, Japan

tochinai@econ.hokkai-s-u.ac.jp

Keywords: natural language processing, spoken dialog agents, affective computing

Received: May 12, 2001

In our project (GENTA - GENeral belief reTrieving Agent), we are trying to realize a conversational agent, which would be able to talk in any domain by using web-mining techniques to retrieve information that is impossible to obtain in usually used corpora. In our research we try to simulate reasoning processes based on Internet textual resources including chat logs. Our goal is a dialogue system which learns the linguistic behaviour of an interlocutor concentrating on the role of emotion during analysing discourse. The system is not using any databases of commonsensical word descriptions, they are being automatically retrieved from the WWW. We describe two values called Positiveness and Usualness and explain their role in the Inductive Learning that is used for achieving emotion-based reasoning skills. As this is a new approach to knowledge acquisition for dialogue agents we concentrate on the theoretical part of our project. Finally we introduce the results of the preliminary experiments.

1 Introduction

WWW is an enormous database, which is being used widely these days. Unfortunately it is said to be difficult to handle as it is full of informational garbage that makes web mining or knowledge-base creation a hard task. However when we started to rummage through those personal home-pages with very similar contents that are seemingly useless for AI purposes, we imagined that human brain cells might look exactly the same. Not only are the stored pieces of semantic information important but also the number of how many times such similar data was stored. We assumed that the Internet is interesting material for retrieving the common sense, beliefs, opinions and emotional information for various types of agents. Without any sophisticated method our system is able to easily discover that in most cases “being cold” is not pleasant and cold beer almost always “sounds nice” or that one movie star is being loved and another hated. In this paper we introduce ideas for our project (GENTA - GENeral belief reTrieving Agent(1)) and the results of initial experiments with implementing a primitive method of retrieving basic feelings towards human user’s utterances and applying this emotional information whilst inductively learning the speech acts. The earliest ideas for our project began whilst observing foreign students’ linguistic behaviour while speaking the Japanese language, which was not their mother tongue. Although their language abilities and cultural background happened to be

very different, their conversations always seemed interesting, which agrees with intercultural will of communication theories(2). We noticed several dependencies, for example that the keywords triggering the conversations were mostly of two types - topics that clearly have a positive or negative emotional load as “a present” or “the red tape” and that the conversational flow concentrates on what the interlocutors have in common or conversely - how they differ. Even if it is quite obvious to human beings, these conditions for what we call interesting conversation are very difficult to be met by machines. First of all, computers have problems recognizing what would be interesting and what could bore its conversational partner. Secondly, they have difficulties with spoken language, especially when there are plenty of grammatical mistakes not to mention interlocutors who use broken language. Therefore we looked for a conversational environment where the computer system could face-up to these challenges and chose IRC (Internet Relay Chat) because of its international character and “world simplicity”(3). Although it is a multi-user environment we concentrated on one-on-one conversations, as the program does not handle multi-thread yet. This time we chose the English language as the initial implementation was done in Japanese(1).

2 Basic assumptions

From the very beginning of human-computer interaction, the purpose of communication was almost always clear - the machine had to understand an order from the human user and help him in some way. That approach is obviously being forced by the pressure that industry puts on the world of science. Even if the talk itself was a purpose of a program(6), it was always supposed to help the user somehow and to be socially useful. Pure “chat for chat’s sake” agents are not widely developed and scientifically neglected because of their low usability and problems with evaluating such systems. In our opinion, concentrating on problem-solving or helping agents makes HMC (Human-Machine Conversations) research unbalanced and we argue against the importance of usability while developing programs that can communicate with human beings. We have noticed that nowadays approaches become more and more sophisticated, machines learn how to analyse metaphors and to stochastically use very large corpora but there are still too few agents based on affective computing(7)(8)(9), that can react not only in a “store-clerk-human-way” but also a “human-way” if a user says “I’m sad” or just “I want to buy a dog”. Since Damasio(10) has underlined the meaning of emotions in reasoning, the number of AI or cognitive science researchers who search for the methods of implementing emotions into machines grows rapidly. Virtual and physical architectures are built and agents or robots are taught to analyse emotional state as the additional information for perceiving. In our approach we have assumed that addition is not enough and that this is crucial, since we understand human conversation as a cycle of exchanging emotion-based information. By that we do not mean that pure information exchange does not exist, we want to make a machine remember that its reaction depends also on the emotional load of an input. Hence our plan is to implement Pavlovian-like reactions into the computer’s conversation abilities. For the simplicity of our system we also simplified an idea on the utterance structure. We imagined a human language interaction as an electron jumping through the layers shown in Fig. 1. It goes up when making an utterance and down when receiving it. We assume that the emotional layer includes intentions that motivate to start, to continue, to change or to stop a conversation flow. We imagine that if anyone puts an electron to any area of semantic layer, for example by saying “a kingdom for a soda!” and an interlocutor who does not know what this cliché exactly means hears it, the electron jumps down to the emotional layer instead of wandering along the complicated semantic network. The interlocutor *feels* that “kingdom” and “soda” are not abusive, “soda” even sounds nice. The nicer it sounds to him, the greater the probability of choosing it for a conversation topic, or as we call it triggering keyword. An aggressive reaction becomes less possible in such case. Guided by the aforementioned assumption, we decided to realize a method that would allow us to simplify semantic layer processing by

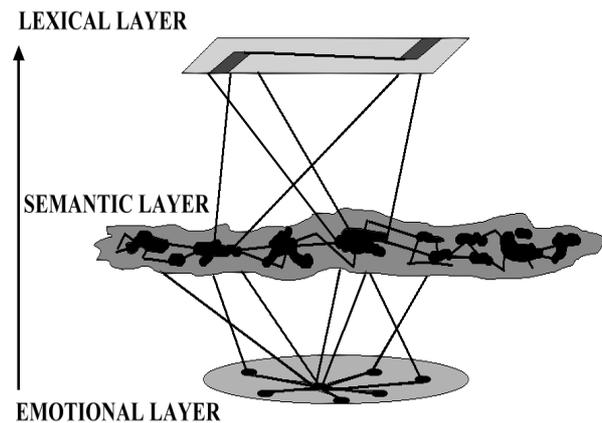


Figure 1: The idea of the utterance structure before simplifying.

limiting choices in lexical and emotional layers. We also made an assumption based on Fletcher division of blind people’s mental imagery(11) and we believe that machines could gain not deficient or inefficient but different imagination. Our idea is that the electron having fewer choices in the emotional and lexical layers may naturally decrease the possibilities of exploring the semantic layer as we illustrate in Fig. 2. Since we are also interested in the aspects of modelling an artificial imagination, we decided to base our system on pure written word-level conversation without audio-visual stimuli, although we used their simplified substitute, as they are necessary for obtaining the basic emotional information. This substitute is the usage of emoticons (facemarks) widely employed in keyboard-based chat, which suggest that a given utterance was, for instance, ironic or supposed to be a joke. For example, if the utterance from Fig. 2. is done with a smiley “:)” the electron should not hit the emotional layer at the fear spot because this would not be natural behaviour.

3 Other kinds of retrieval

In order to see what kind of information could be retrieved from Internet resources and used for general belief and emotion processing, we made several tests with different search engines. For retrieving common emotions we were observing the hit numbers of the results of searched frames as: “I am afraid of” [N] (for nouns usually causing fear) “I always wanted a” [N] and “I always wanted to” [V] (for what people may dream about) or “Usually people [V] at [N]” (for retrieving verbs for building a script of given noun). We discovered that it is possible to retrieve thousands of sentences that could be processed for what we call emotional common-sense statistics. The more specific given frame was, for instance “I always wanted to be” instead of “I always wanted to”, the less ambiguous sentences were retrieved and results as “I always wanted

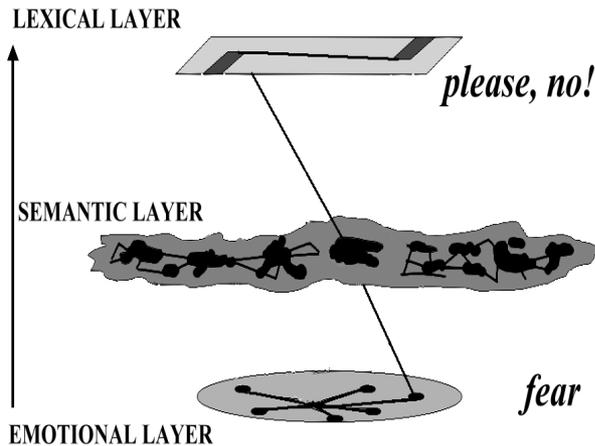


Figure 2: The idea of the utterance structure after simplifying.

to do it” were apparently decreasing. As we noticed in previous works(1)(4)(5), the Internet can also provide basic semantic tagging by counting frames including prefixes (or particles as in Japanese). For example if a parser receives an unknown word, let’s say “Sapporo”, it can easily declare that it is a place by comparing result numbers of given searching frames: “I talked to [Sapporo] and”, “I felt [Sapporo] then”, “I ate [Sapporo] with”, “I went to [Sapporo] by”, etc. Such frames must be chosen very carefully and one should use stop words as “and”, “then”, “with” or “by” to avoid counting expressions as “I talked to Sapporo Council” but treating such expressions as “animates” also produced some interesting effects. Search engines also provide misspell detection what can be useful in fast typing chat environments with non-native speakers. Unfortunately automation of such searching requires the usage of crawlers (web robots) and complicated filtering which still takes too much time to keep up with interactive conversation processing. Therefore we decided to limit the retrieved information only to the simple test of whether something is liked or disliked and how common it is.

4 Description of GENTA system

4.1 General Belief

By this expression we mean a mixture of common sense based on retrieved opinions and simplified environmental knowledge. The system’s knowledge of a user is assumed as almost none - GENTA does not know the nationality, age or sex of its conversation partner; he or she is not necessarily a native speaker of English. After exchanging greetings, the system waits for a user’s initial utterance and if there is none it starts a conversation using the learning data of “Conversation Keeper”, which will be described later. While detecting the speech act, which is also explained below, GENTA tries to guess the leading keyword(s) from the

first user’s utterance since the domain of conversation is still unknown. Next, GENTA searches the Internet for the whole utterance and its grammatically connected parts previously parsed by a parser(13) trying to establish what can be associated with given verbs, nouns, noun phrases, adjectives or conditional expressions concentrating on feelings-based opinions. For example when the input is, “Do you like playing soccer when it rains?” GENTA counts how many sentences “I like playing soccer when it rains”, “I love playing soccer”, “I hate playing soccer”, and “I love it when it rains” and “I hate it when it rains”, etc. appear on the Web. This lets our system achieve “own” opinion about playing soccer when it rains because this knowledge is assumed as “general” or “common”. Then, paraphrasing Shannon’s information theory(14), we assume that the keyword with less frequency is more interesting for interlocutors and GENTA chooses playing soccer as a leading topic (27268 vs 33985 hits). The system believes that the discourse should be continued in this “semantic direction”. But before that, “Conversation Keeper” must establish which linguistic behaviour (called “a dialogue act” here) will be proper for a reply, which is our current task.

4.2 Conversation Keeper

Taking last decade research results into consideration (15)(16)(17)(18)(19)(20)(21)(22) we decided to find precisely the combination of web-mining and learning methods that would help us to create a dialogue system that does not require a large amount of initial data prepared by hand and does not need sophisticated modules. As our first step, before creating the real dialogue manager, NLG module, etc., we decided to confirm that our system is able to learn from above-word level information. As already mentioned, we assumed that peculiarity and emotional load of given expressions could support intention recognition, which is one of the most important tasks of human discourse management. Therefore here we divided General Beliefs into two above word-level values that we call Positiveness and Usualness, which are also measured by counting the aforementioned string frequencies upon the WWW. Positiveness value is calculated with following formula:

$$Positiveness = \frac{C_{\alpha_1} + C_{\alpha_2} * \gamma}{C_{\beta_1} + C_{\beta_2} * \gamma}$$

$$\alpha_1 = disliked, \alpha_2 = hated$$

$$\beta_1 = liked, \beta_2 = loved, \gamma = 1.3$$

Where γ is to strengthen the “love” and “hate” opinions. We prepared dialogue act tags, as handing or demanding of information, opinion and reason; advising, warning, greeting and nodding. GENTA can automatically declare Usualness and Positiveness for utterances, as in the following example:

Do you like playing soccer when it rains?

becomes a *DAPU string* (Dialogue + Act + Positiveness + Usualness):

OD P5U4 cond P3U5

which means that it was a Demand of Opinion consisting of two positive and usual expressions connected by subordinate clause conjuncter (SCC) “cond” (conditional clause). What is characteristic for our method, even if the Positiveness of expression seems to be doubtful (that most Web page creators like it when it rains that does not necessarily mean that most human beings also do) it does not disturb the process since the opinion remains logical. Values of Usualness and Positiveness are calculated by comparing frequencies of (“I don’t like ...” / “I hate ...”) and (“I like...” / “I love ...”) searching frames. The frequency thresholds are different depending on the length of the searched string.

4.3 Inductive Learning

GENTA system has the ability to learn from spontaneous human conversations. We use the Inductive Learning method(16) to predict which utterance should be used and to make new rules while talking. The system represents dialogue discourse as connected *DAPU strings*

OH P5U2 : OD P5 U3 : ND : ...

(A1 B1 A2 B2 A3 B3 ...)

which are divided into double rules

(A1 B1) (B1 A2) (A2 B2) (B2 A3) (A3 B3) ...

stored in a Dictionary (Fig. 3,4). When new input is done, GENTA parses the utterance to *DAPU string* by recognizing a dialogue act determinator, which are words attributed to every dialogue act tag. For example, should determines advising tag. If subordinate clause conjuncter (SCC) is detected, both clauses are parsed into *DAPU strings* and they become an individual element for learning. When there is more than one sentence during one turn, GENTA confirms if they are of the same dialogue act. If not, the input is divided - the last rule is changed and a new one is created, for example:

A1: Do you care?

B1: Well, I don’t care. What about ya?

A2: Me neither, man!

creates (A1:B1a) (B1b: A2) instead of (A1:B1) (B1: A2).

Learning concentrates on dialogue acts tagging and conjuncters, and their coexistence with Positiveness and Usualness. For example, if an unknown dialogue act determinator appears, our system decides the most probable tag and unless a user cancels a computer’s output by using one of the cancelling expressions, such as “???” or “What are you talking about?” and so on, a new rule is created in the rule dictionary (Fig. 4.). If an emoticon is detected, the Positiveness value is decreased or increased depending on the kind of a facemark.

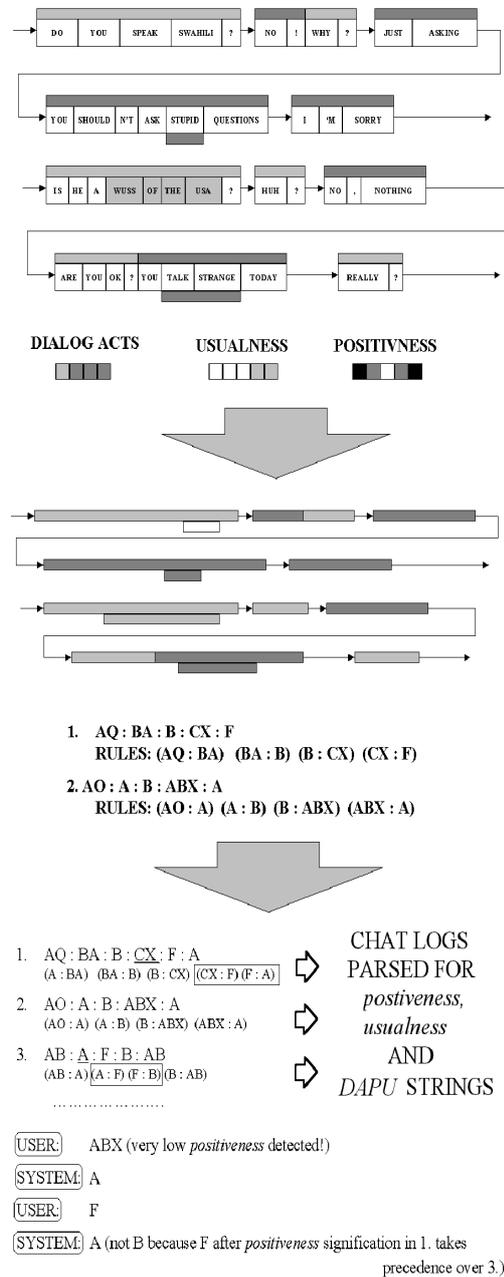


Figure 3: A simple example of the learning process from the utterances already parsed for Positiveness, Usualness and *DAPU strings*.

5 Experiment and evaluation

5.1 Method

Existing standards (23) in HMC evaluation, which concentrate on semantic quality of output, were not useful for evaluating spontaneous chat only with above-word information supported dialogue acts. Hence we had to prepare our own evaluation method:

- There are two human interlocutors A and B.
- They converse through IRC channel, which is monitored by our system (G).
- G listens to A's utterances and proposes its own answers (as *DAPU strings*).
- *DAPU strings* of B's utterances are compared with G's ones.
- Afterward the third person evaluates the naturalness of the strings when a system chose a different dialogue act, as there is more than one possibility.

5.2 Results

As we are not particular about perfect language, two non-native English speakers took part in our experiment. There was no particular topic of conversation. Subjects made 128 turns and they mostly talked about sports. GENTA's dictionaries were empty in the initial phase and we taught the system only one determinator for every dialogue act and only two to three basic conjunctors for every kind of subordinate clauses. We decided on empty dictionaries to see when a system starts to learn and because we want GENTA to retrieve what is needed from the Internet. The Web as a corpus that is constantly changing and creating dictionaries for Positiveness or Usualness would not reflect those changes. By comparing user B and GENTA's *DAPU strings* we understood that:

- The systems already started to use learned rules by the eighth turn, as the chat was mostly question-answer style but finally less than half (37.5%) of dialogue acts were chosen the same way by a user. Although 81.25% of those different ones were evaluated as natural by human being.
- Positiveness (On a five point scale: 1-negative, 2-slightly negative, 3-neutral, 4-slightly positive, 5-positive) of systems output that had the same dialogue act tag as a human was in 59.1% of cases the same as the user's. By this we mean there are three levels: positive, indifferent, negative.
- Usualness (On a five point scale: 5-very usual, 4-usual, 3-slightly peculiar, 2-peculiar, 1-very peculiar) of a system's output was only in 20.8% of cases the same as a human user's, because all parser errors due to misspells were detected as the most peculiar expressions.

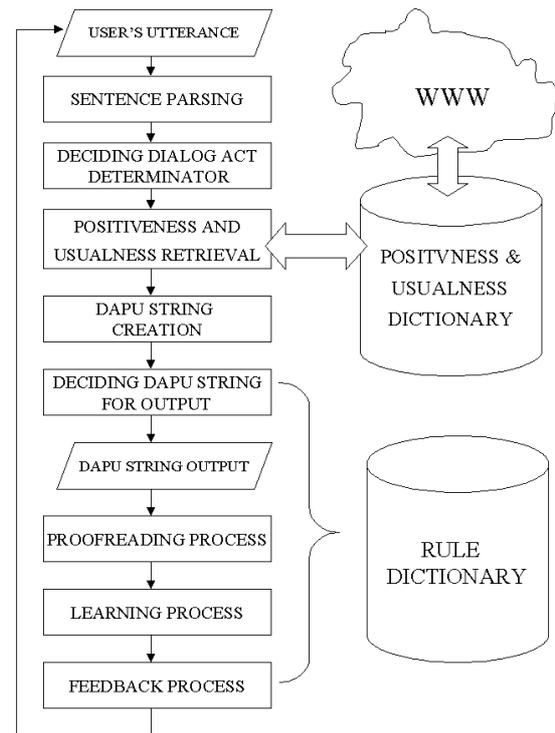


Figure 4: GENTA's Inductive Learning process.

Positiveness and Usualness were compared only in cases where dialogue acts were the same in human and machine outputs, as the dialogue acts choice significantly influences those two values. Because these two values were depending on the Internet connection speed (about 15-20 seconds for 1 calculation), the computer's propositions were given with growing time difference but it had no influence on our experiment's results.

6 Conclusion and discussion

We have described a new approach to WWW statistical information usage in a dialogue system, which is able to achieve information that is not obvious to the machine without using logic programming and other sophisticated methods. As we are in the preliminary stages of our project, we could only indirectly evaluate our ideas as grammatically built sentences were not output by the system. However, the results are convincing enough to continue walking on our chosen path - even if the system was not guessing an interlocutor's intentions properly, it proposed its own dialogue acts, which were not against the logical flow of conversation. In most cases too few turns made learning material inefficient but it is quite difficult to evaluate GENTA system before implementing further modules that will lead to generating more understandable output. The following is an example of this difficulty:

A: They prefer drinking beer. (IH P5U4)

G: P5U4 ex (that could be: “a beer!!!” or “I love beer!”)

B: C’mon, they’re watching games too. (IH P4U3)

It seems obvious that at this stage it is too early to evaluate our system as a talking system and it is rather impossible to see its abilities in context management. Although when we add the knowledge retrieval and representation modules we foresee that such outputs will be useful. We believe that tuning up the parsing methods and increasing data for learning will help to achieve better results in the future. We made a first step in the creation of an agent that should be able to chat about any topic with proper human-like reactions. A promising factor for future tasks is that our program was only based on automatically retrieved knowledge of common opinions and the peculiarity of user’s utterance, which could be used in many interesting ways, as manipulating GENTA’s “personality” for example by decreasing its Positiveness when, for instance, the weather is bad. Another idea is that one could use retrieved information as a model imagination of an interlocutor. We want GENTA to know what his partner probably thinks while for instance saying “I need a girlfriend”. Thus there is a need to experiment with different parsers and to create mechanisms which allow GENTA to learn other things from the Internet - the largest and most rapidly growing database in the world, and try to apply those methods to commonly explored areas as for example in qualitative spatial reasoning. It must also be able to answer “wh-” questions, so we plan to concentrate on implementing a substitution of imagination which should be an elastic plan retrieval mechanism supported by commonsensical libraries created through search frames as “I always (verb) when it rains” or “usually people buy (noun) when they want to (verb + noun)” and also on the automatic creation of such frames. Our method could also be interesting from a sociological point of view, since GENTA can become a “mirror personality” of an average wired English or Japanese speaker in his original version(1), which could make it a much more interesting conversation partner than its predecessors. By using searching frames as “computers never”, “computers can” we can also model a basic knowledge base for a machine that could also be conscious of its possibilities. We also understood that adopting GENTA to other languages would only be limited to translating the search frames. We strongly believe that with the constantly improving computer and network abilities, the Internet will become the main source of any kind of knowledge for future AI systems. We also predict that millions of private homepages sharing users’ feelings and opinions will be crucial information to helping machines to “understand” what we typically think and what average humans know.

Acknowledgement

This work is partially supported by the Grants from the Government subsidy for aiding scientific researches (No.14658097) of the Ministry of Education, Culture,

Sports, Science and Technology of Japan.

References

- [1] R. Rzepka, K. Araki, K. Tochinai (2001) Basic Idea of General Belief Retrieving Agent “GENTA” 2001 *IEEE Hokkaido Chapters Joint Convention Record*, IEEE Sapporo Section, pp. 414-415.
- [2] B. Malinowski (1922) *Argonauts of the Western Pacific*, Routledge & Kegan Paul.
- [3] R. Rzepka (1999) Communication methods in Japanese Internet chat - new semiotic changes of the natural language structure, *Master’s Thesis, Department of Linguistics*, Poznan University.
- [4] R. Rzepka, K. Araki, K. Tochinai (2002) Prediction of the User’s Reply Using Emotional Information Retrieved from Internet Resources 2002 *IEEE Hokkaido Chapters Joint Convention Record*, IEEE Sapporo Section, pp. 229-230.
- [5] R. Rzepka, K. Araki, K. Tochinai (2002) Is It Out There? The Perspectives of Emotional Information Retrieval from the Internet Resources, *Proceedings of the IASTED Artificial Intelligence and Applications Conference*, ACTA Press, Malaga, pp. 22-27.
- [6] J. Weizenbaum (1966) ELIZA - a computer program for the study of natural language communication between man and machine, *Communications of the Association for Computing Machinery* 9, ACM, pp. 36-44.
- [7] J. Bates (1994) The role of emotion in believable agents, *Communications of the Association for Computing Machinery* 37, ACM, pp. 122-125.
- [8] R.W. Picard, J. Klein (2002) Computers that Recognise and Respond to User Emotion: Theoretical and Practical Implications, *MIT Media Lab Tech Report 538*, to appear in *Interacting with Computers*.
- [9] M. Seif El-Nasr, T.R. Ioerger, J. Yen (1999) A Web of Emotions, *Proceedings of Workshop on Emotion-Based Agent Architectures part of Autonomous Agents ’99*.
- [10] A.R. Damasio, (1994) *Descartes’ error – emotion, reason, and the human brain*, Avon, New York.
- [11] J.F. Fletcher (1980) Spatial representation in blind children, Development compared to sighted children, *Journal of Visual Impairment and Blindness* 74, pp. 381-385.
- [12] J. Groenendijk, M. Stokhof, F. Veltman (1996) Coreference and modality, In S. Lappin, editor, *Handbook of Contemporary Semantic Theory*, Blackwell, Oxford, pp. 179-213.

- [13] D.D. Sleator, D. Temperley (1994) Parsing English with a link grammar, *Third International Workshop on Parsing Technologies*.
- [14] C.E. Shannon (1948) A mathematical theory of communication, *Bell System Technical Journal*, vol. 27 pp. 379-423 and 623-656.
- [15] J. Alexanderson (1996) Some ideas for the automatic acquisition of dialogue structure, *Vermobil Verbundvorhaben, Report 157*.
- [16] K. Araki, K. Tochinai (2001) Effectiveness of Natural Language Processing Method Using Inductive Learning, *Proceedings of the IASTED International Conference Artificial Intelligence and Soft Computing*, ACTA Press, Cancun.
- [17] D. Jurafsky, R. Bates, N. Coccaro, R. Martin, M. Meteer, K. Ries, E. Shriberg, A. Stolcke, P. Taylor, C. Van Ess-Dykema (1997) Automatic Detection of Discourse Structure for Speech Recognition and Understanding, *Proceedings of 1997 IEEE Workshop on Speech Recognition and Understanding*, IEEE.
- [18] J.R. Glass (1999) Challenges for spoken dialogue systems, *Proceedings of the 1999 IEEE ASRU Workshop*, IEEE.
- [19] J. Kreutel, C. Matheson (1999) Modelling questions and assertions in dialogue using obligations. *Proceedings of Amselogue 99, the 3rd Workshop on the Semantics and Pragmatics of Dialogue*, University of Amsterdam.
- [20] A. Stockle (1998) Dialog act modeling for conversational speech, *In Papers from the AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*.
- [21] L. Bell, J. Gustafson (1999) Utterance types in the August dialogues, *Proceedings of IDS'99, ESCA Workshop on Interactive Dialogue in Multi-Modal Systems*.
- [22] J. Gustafson, M. Lundeberg, J. Liljencrants (1999) Experiences from the development of August - a multi modal spoken dialogue system, *Proceedings of IDS'99, ESCA workshop on Interactive Dialogue in Multi-Modal Systems*.
- [23] W. Minker (1998) Evaluation Methodologies for Interactive Speech Systems, *Proceedings of LREC'98*, Granada.

Data Compression in a Pharmaceutical Drug Candidate Database

Zina Ben Miled¹, Huian Li², Omran Bukhres², Michael Bem³, Robert Jones³, Robert Oppelt³

¹ Electrical & Computer Engineering, Purdue School of Eng. & Tech., Indianapolis, IN 46202

² Computer & Information Science, Purdue School of Science, Indianapolis, IN 46202

³ Eli Lilly and Company, Indianapolis, IN 46202

Keywords: drug candidate database, compression, random access

Received: January 15, 2001

Pharmaceutical drug candidate databases have reached massive sizes in recent years due to the improvement of benchside high throughput screening tools used by scientists. This rapid increase has caused a shift in the bottleneck in discovery and product development from the benchside to the computational side, thus creating a need for new computational tools that can facilitate the access and interpretation of such massive data.

In this paper, a window-based compression technique that supports random database access is introduced. This technique improves random access to records in the database while maintaining high sequential throughput. The impact of the proposed compression technique is evaluated in the context of a non-indexed and an indexed database. The performance gain of the window-based compression technique is demonstrated using a drug candidate database which is used in the pharmaceutical drug discovery process.

1 Introduction

During the past decade, the amount of data generated in early drug discovery process exploded because of the introduction of lab automation and enhanced screening technologies. This data explosion created the need for analysis tools enabling a bench scientist to use the data generated to facilitate the discovery of new drugs. There are several challenges associated with the data management of these data because the associated databases are highly multidimensional, complex, and dynamic (3; 4).

In this paper, a window-based compression technique is introduced for pharmaceutical databases and its impact on improving data access in the databases is evaluated. Window-based compression has been successfully used for images such as JPEG (34). Compression can reduce the response time for data access in large size databases by transferring smaller size data blocks from disk and decompressing the data in memory. The proposed window-based approach is applicable to databases that mostly contain numerical data. The database under consideration needs to be sorted with respect to the values of the attributes. The sorted database is then partitioned into a set of records called windows. The data in all the windows is encoded and indexed. This approach is scalable with respect to the size of the data in the database while allowing random access to the data.

In this paper, a drug candidate database is used to show the performance gain associated with the proposed compression technique. This driving application is an important step in the drug discovery process. An overview of the application is presented in Section 2. Section 3 presents related research work. The window-based compression is

described in Section 4. Section 5 analyzes the impact of indexing and window-based compression on data throughput in the drug candidate database. Section 6 describes the buffer techniques used to facilitate updates in the compressed databases. Conclusion and future work are included in Section 7.

2 Application

Given a target disease, such as cancer, the drug discovery process consists of determining a pharmaceutical therapy for this target disease. This involves finding a strategic macromolecule that is pivotal to the disease state. Knowledge of the receptor site on the macromolecule allows the scientist to find new smaller molecules that will fit it, much like a key fits a lock. These molecules constitute lead drug candidates. The more accurate the fit, the more efficient the drug candidate is in binding to the receptor site, and thus a more potent drug candidate is realized. This study focuses on the computational process involved in extracting lead drug candidates from a database containing millions of drug candidates. Once a set of lead drug candidates is extracted for a given target disease, extensive structural modifications of the candidate's molecular structure are made and biological tests are conducted to identify at least one drug candidate for advanced studies. The objective is to select a suitable drug candidate that can be proven effective in the disease and thus may be sold as a pharmaceutical therapy for the target disease.

The database consists of the descriptor values for the drug candidates, which are computed by using several descriptor programs. Descriptors refer to numerical representation of a molecular attribute of the drug candidates. Each

descriptor program computes a specific set of descriptors for a compound. These descriptors quantify the characteristics of the molecule that may be important in fitting the structure to the receptor site. Example descriptor programs include Fingerprints (37), Jurs (30), Topological Indices (1), etc... For each descriptor program, the relation between the drug candidates, the descriptors, and the descriptor values is shown as a relational database table in Figure 2. The first column (attribute) of this table is the identification number for the drug candidate (Drug ID). The second column (attribute) identifies the descriptor for which the corresponding drug candidate has a value. The third column (attribute) contains the value of the corresponding descriptor (column 2) for the given drug candidate (column 1). These descriptor values are actual numerical values representing different characteristics of the molecule such as the number of sequences of a given set of atoms that are present in the molecule (drug candidate).

Drug candidates may not have values for all the descriptors in a given descriptor program. For example, if the descriptor represents the presence of a given sequence of atoms in the drug candidate, that sequence may not be present in all the drug candidates. The “missing value” is used to refer to the drug candidate-descriptor tuple for which a descriptor value does not exist. For example, if the descriptor represents the presence of a given sequence of atoms in the drug candidate (molecule), that sequence may not be present in all the drug candidates. Instead of representing this information as a record of type “drug candidate number, descriptor number, descriptor value = 0”, the entire record is omitted.

desc. prog.	1	2	3	4	5	6
No. of Desc.	8	149	49	15,265	208	2,048
desc. prog.	7	8	9	10	11	12
No. of Desc.	88	2	102	36	557	30

Table 1: Number of descriptors in each descriptor program

Drug Candidate	Descriptor	Value
1	100	54
2	100	55
2	101	56
2	102	1.5
...
6555	100	32
6555	102	2.4

Table 2: Physical data representations for each descriptor program

A drug candidate database may include several million drug candidates and thousands of descriptors. Without loss of generality, we limit the study presented in this paper to a subset of the large database containing 300,000 drug candidates. There are 12 descriptor programs in this drug candidate database, each containing a varying number of unique

descriptors (Table 1). Each descriptor program in the drug candidate database is stored as a flat file on disk using the physical data representation shown in Table 2.

During the drug discovery processes, the database is screened in order to extract lead drug candidates on the basis of a computational model. In the simplest form this model is the weighted sum of several descriptor values. For example, given a target disease D , the scientist may establish that the computational model for a lead drug candidate is $0.5 \times d_1^5(x) + 0.3 \times d_3^7(x)$, where $d_1^5(x)$ and $d_3^7(x)$ are the values of descriptor 1 from descriptor program 5 and descriptor 3 from descriptor program 7, respectively, for the drug candidate x . The higher the value of this weighted sum the greater the likelihood that the drug candidate will be an adequate pharmaceutical therapy for the disease D . Of course, the actual computational models are more complex than the above example. A typical computational model may include 100 descriptors.

When screening the database, scientists may give a customized drug candidate input list or require the entire database to be searched. In the first case, the drug candidate IDs are randomly selected from the database and the data access to the database is random. In the second case, the drug candidate IDs in the drug candidate input list are sequential numbers and the database access is consecutive. Using the drug candidate list given by the scientists and the descriptor list from the model, the corresponding descriptor values are retrieved from the database and used to apply the model to find the lead drug candidates.

3 Related work

Database compression is different from traditional data compression because it should support conventional database operations, such as retrieval, update, insertion and deletion. The difficulty that is common to all these operations is allowing random access to the data. Additionally, only lossless compression can be used with certain scientific databases such as pharmaceutical and medical databases, because accuracy is very important for these databases and cannot be traded for performance.

Several previous studies have discussed database compression and random access to compressed data files (27; 16; 26; 35; 23; 18; 17; 20; 11; 22). An overview of various compression techniques available for database compression and the benefits and disadvantages of these techniques are presented in (27). Furthermore, a comparative analysis of the level of granularity for compressed databases is discussed extensively in (26). The different levels considered in (26) include: file, page, tuple, and attribute. In (26), it was established that file and page level compression are inefficient because of the overhead associated with decompressing the entire file or page in search of a given tuple in database queries. Attribute level was selected over tuple level compression in (26; 14) for two reasons: first, there is a high overhead when decompressing the entire tu-

ple in order to retrieve a single attribute from the tuple, and second, there is a high level of similarity present within a single attribute across all the tuples. The approach used here combines both attribute level compression and tuple level compression within a single window. Furthermore, the overhead associated with decompression is limited because partial decompression is used when a query is issued against the compressed database. The first attribute of the first tuple in the window is decompressed in order to decide whether the desired data is contained in the window. The same partial decompression approach is used for a single tuple within the window.

The approach discussed in (14) uses an offline dictionary method that encodes each attribute based on the range of values in the domain of this attribute. It is applied on an attribute by attribute basis. This approach can not be directly applied to the drug candidate database because the values of the attributes span the entire domain. Additionally, as apposed to the approach discussed in (14), in this paper, compression is applied to multiple tuples at a time. As will be shown later in this paper, compression on a per tuple basis yields little improvement in performance. The approach discussed in (14) is general purpose whereas the approach discussed in this paper is more applicable to scientific database applications.

The approach proposed in this paper is based on a non-adaptive compression technique. This choice was made based on the findings stated in (16; 26) which indicate that an adaptive approach (i.e. without apriori knowledge of the characteristic of the data) results in insignificant reduction of the data size.

A strong evidence for the advantage of compression in databases is offered in (35) which provides a method for incorporating compression in DBMSs. In (35), it was also shown that compression incurs significant penalties for insertions, deletions, and updates. All the techniques considered in (35) are restricted to the compression of each tuple in the database independently. The compression proposed in this paper is applied to several tuples (belonging to a window) at a time. Furthermore, a delayed update approach, as discussed in Section 6, is used in conjunction with the window-based compression approach. These two characteristics yield a limited penalty for insertions, deletions, and updates in the compressed database.

Tuple differential coding was proposed for statistical database compression (23). This technique includes several steps: tuple reordering, attribute domain ranking, block partitioning and block encoding. Tuple differential coding (23) is based on the assumption that the attribute domains are simple and of finite size. The domain is usually quantified so that each attribute value falls in a predefined domain range. In the drug candidate database, the domain of the attributes is complex, and the length of the attribute values may range from one byte to multiple bytes. This complexity makes the compression technique proposed in (23) impractical. The window-based compression starts by sorting the database. Although different from the approach intro-

duced in (23), this sorting can be considered as a form of tuple reordering. Furthermore, in the window-based compression a similar technique to block partitioning and block encoding is used. The block partitioning discussed in (23) uses fixed size blocks, whereas the partitioning used in this paper uses variable size windows. Actually one of the objectives of this paper is to evaluate the impact of the size of the window on the access throughput to the database.

A semantic compression based on fascicles was introduced in (17). The underlying idea of this compression technique is to identify and group together records that share similar values for some attributes. The approach proposed in this paper also takes advantage of the similarity between records. The window-based compression technique relies on the similarity resulting from a database sorted according to the values of the attributes. Sorting the database is less compute-intensive than the clustering approach proposed in (17).

Current commercial DBMSs support various types of data compression. DB2 exploits the hardware compression implemented in S/390 (5; 15; 16). This hardware compression employs a modified LZ algorithm (39; 40) which is assisted by an auxiliary processor. A compression call instruction and compression dictionaries are required to activate the hardware compression. DB2 switches to software compression if the hardware compression support is not available. In order to achieve adequate gain from compression, the data have to meet certain specific criteria. The length of a tuple cannot be short, which means a tuple usually has to have many attributes. Furthermore, the tuples should have repeated strings, for example, reoccurring strings in the names and addresses. The tuples in the drug candidate database do not have these characteristics. They are short and do not include repeated strings.

Sybase also uses compression. In the Sybase IQ Server (31), which is a data warehouse for interactive decision support, a compression technique similar to gzip (39; 40) is used. This compression technique is implemented at the page level. Whenever a tuple is required by users, the entire page has to be decompressed.

Oracle supports a key compression technique (25) which compresses parts of the primary key column values in an index. Keys in an index are divided into two parts, a grouping part (the prefix entry) and a unique part (the suffix entry). The grouping part is shared by multiple unique parts allowing for an efficient compression of the index space. For example, in an index with three columns, suppose the grouping part consists of the first two columns and the unique part is the last column. For a list of keys (1,1,5), (1,1,6), (1,2,5), (1,2,7), (1,2,8) the repeated occurrences of (1,1) and (1,2) in the grouping part are compressed. The efficiency of key compression depends on the division of the key and the data in the key. The compression technique used in Oracle is similar to one of the steps used in the window-based technique. In addition to this processing step, the window-based technique groups the tuples in windows and uses encoding to compress the data in the tuple.

Char.	0	1	2	3	4
Freq.	6.74	8.51	9.95	9.89	7.72
Char.	5	6	7	8	9
Freq.	6.77	6.27	6.29	6.39	6.26
Char.	-	.	,	S	end of line
Freq.	0.14	0.83	7.62	8.54	8.08

Table 3: Frequency of occurrence (in percent)

In this section, compression techniques previously used with database applications have been reviewed. There are several other compression techniques, such as Elias (10) and Golomb (12) codes for Integers, modified Huffman coding of bigram and trigram (32), and arithmetic coding (19). Elias and Golomb codes are not suitable for the drug candidate database since attributes with floating point numbers make-up most of the data in the database. Under the observation of some frequently used strings in the database, bigrams are used in the proposed window-based compression. The amount of compression achieved by arithmetic codes and Huffman codes is comparable (29). In this paper, a modified Huffman code was selected.

4 Window-based Compression

In this section, the window-based compression algorithm is formally presented, followed by a description of the environment used to implement and analyze the proposed compression technique for the pharmaceutical drug candidate database.

4.1 Algorithm Description

The first step in the proposed compression technique consists of identifying the set of characters that are present in the drug candidate database and their corresponding frequency of occurrence. The frequency of occurrence is the ratio of the number of times a given character is used to the total number of characters in the drug candidate database. The drug candidate database under consideration contains mostly numerical values. Table 3, shows the frequency of occurrence of all the characters in the drug candidate database.

The character “,” is used as a separation between attributes within a record, the symbol “-” is the negative sign, the symbol “.” is used in floating point number, the symbol “S” represents a frequently used bigram “,1”, and the end of line symbol separates two adjacent records. Because the variance in the frequency of occurrence of characters is small, a uniform size (4 bits) code word was assigned to each character in the database. Variable size code words, based on the Huffman code (8) were also derived. However, these code words did not result in higher throughputs for the drug candidate database.

There are also other methods that generate variable code

words (6) for large databases. The method proposed in (6) derives the code words based on a small subset of the database rather than the entire database. This approach would be efficient in generating code words for the drug candidate databases if the frequencies of occurrences varies greatly among the characters and if these frequencies change when the instance of the drug candidate database increases in size. Various tests were conducted which showed that neither of these two characteristics were applicable to the drug candidate database.

The second step is to sort the data in the database using radix sorting (8). This is a time consuming operation for large pharmaceutical databases, but since the data in the database is stable and the only updates are tuple inserts which are performed few times per year, sorting the database and keeping it sorted is possible. The next step in the compression technique consists of grouping adjacent records into *windows*. An example of a window of size 4 before compression is shown below.

drug cand.,	desc. number,	desc. value
100000,	1001,	0.22156582
100000,	1002,	1.32567674
100001,	1001,	3.81434905
100001,	1002,	2.56932236

The first column of this window is the drug candidate number; the second and the third columns correspond to the descriptor number and descriptor value, respectively. Given that the database is sorted, the values in the first and second column are always in ascending order within the window. Furthermore, the difference between the values of these attributes from one record to the next is small. In order to take advantage of this similarity between adjacent records, a difference operation is performed on the attributes of columns 1 and 2. This operation, which is similar to the key compression approach used in Oracle, is shown below for the example window of size 4.

drug cand.,	desc. number,	desc. value
100000,	1001,	0.22156582
0,	1,	1.32567674
1,	0,	3.81434905
1,	1,	2.56932236

The difference operation is applied to a subset of the attributes. These attributes can be specified by the user. Alternatively, a simple algorithm that performs one pass over the database, after it is sorted, can determine whether or not a given attribute is a good candidate for the difference operation. For example, applying the difference operation to the third column of the drug candidate database will not result in any gain because of the lack of similarity.

The third and final step in the compression consists of replacing each character in the partitioned database by its corresponding code word. The window-based compression algorithm is shown in Figure 1.

```

1.   $w \leftarrow 0$ 
2.  WHILE (!eof(database))
3.      read a tuple  $t_w = (p_w, k_w, d_w)$ 
4.      IF ( $(w > 0) \ \&\& \ (w < \text{window size})$ )
5.           $t'_w \leftarrow (p_w - p_0, k_w - k_0, d_w)$ 
6.          compress  $t'_w$  using 4-bit code
7.      ELSE
8.          compress  $t_0$  using 4-bit code
9.      END IF
10.      $w \leftarrow w + 1$ 
11.     IF ( $w == \text{window size}$ )
12.          $w \leftarrow 0$ 
13.     END IF
14. END WHILE

```

Figure 1: Window-based Compression Algorithm

Following the classification introduced in (27), this technique is reversible (or lossless). Furthermore, the encoding step of the proposed technique falls under the restricted variability codes approach (27) and the difference operation falls under the differential compression technique (27). The combination of these approaches are intended to accommodate the fact that the drug candidate database consists of a combination of integers as well as floating point numbers. Differential compression is adequate for integers whereas the restricted variability codes are more suitable for floating point numbers than integer numbers. As stated in (27), floating point numbers tend to compress at the rate of 10%, whereas the compression rate for integers ranges between 45% and 65%.

4.2 Algorithm Implementation

The algorithm was implemented using the C language on a Sun Enterprise 3000 with 4 processors and 1 GB memory. A small DBMS that supports indexing and update operations was developed and used as a test bed for the implementation of the proposed compression algorithm. Under this test bed the database tables are stored in flat files. The implementation of the proposed compression algorithm using commercial DBMS was not possible due to the unavailability of source code. Additionally, using an existing open source DBMS would require rewriting most of the code in the database engine.

5 Evaluation

The performance of the window-based compression is measured according to three parameters: the compression ratio, the execution time of a full database scan, and the execution time associated with the retrieval of a set of descriptor values for a set of drug candidates. The latter operation is the most frequent operation performed on the pharmaceutical drug candidate database. It is used to validate

the computational model for the lead drug candidates associated with a given target disease. The first parameter allows us to compare the window-based compression to commonly available compression tools such as the Unix “compress” (39; 40). The full scan is the operation used to extract all lead drug candidates from the database. The performance of the window-based compression with respect to these three parameters is discussed in the following subsections.

5.1 Compression Ratio

The compression ratio is defined as the size of the original database divided by the size of the compressed database. Table 4 shows the compression ratio obtained by using the window-based technique for each descriptor program in the drug candidate database. The last row of this table includes the average compression ratio for the window-based compression technique over all the 12 descriptor programs in the drug candidate database. The second and third columns of the same table include the compression ratio obtained when the Unix “compress” and “gzip” (39; 40) are used, respectively. The last column of Table 4 corresponds to the compression ratio obtained when a variable window size is used. The variable window size refers to the case where a new window is started every time a new drug candidate number is encountered.

Desc. Prog.	Unix comp.	Unix gzip	Window Size				
			1	4	8	16	Var.
1	2.89	3.14	2.12	2.62	2.60	2.65	2.73
2	4.03	5.00	2.28	3.51	3.70	3.62	3.63
3	3.93	4.75	2.16	3.12	3.20	3.18	3.28
4	3.83	5.17	2.28	3.31	3.36	3.23	3.30
5	3.89	5.16	2.27	3.32	3.39	3.35	3.57
6	4.57	5.89	2.26	3.74	3.90	3.83	3.85
7	3.37	3.71	2.10	2.98	3.11	3.14	3.07
8	3.15	3.56	2.19	2.91	3.04	3.11	2.69
9	3.53	3.80	2.08	2.91	2.97	2.98	3.04
10	3.79	4.76	2.30	3.29	3.34	3.29	3.58
11	4.96	6.39	2.28	3.03	3.18	3.25	3.72
12	2.51	2.55	2.09	2.43	2.44	2.42	2.51
Aver.	4.30	5.30	2.24	3.19	3.31	3.33	3.55

Table 4: Compression ratios obtained using Unix compress and window-based compression with different window sizes.

Both “compress” and “gzip” achieved higher ratio than window-based compression. “compress” and “gzip” are based on the LZ algorithm (39; 40), which replaces the strings with pointers to the same strings that occur earlier in the text. If this technique is applied to the entire drug candidate database, it cannot efficiently support random access to the data in the database (11). It is also not suitable within each individual window for the following reasons: First, most data repetition is removed within each window after the difference operation is performed. Second, the data size in each window is small. A typical window with size 16 contains 100 to 250 bytes. If a very large window is chosen to favor the use of “compress” or “gzip”, the throughput during random access to the drug candidate database will

be low. This is because the entire window has to be decompressed in order to search for few random drug candidates. Random access is the most frequent operation used by scientists. Thus, a high throughput for this operation has to be achieved.

Table 4 shows that larger window sizes result in higher compression ratios. However, the difference in compression ratios between consecutive sizes diminishes as the window size increases. For example, the difference between the average compression ratios for window sizes 1 and 4 is 0.95. Whereas, the difference between the average compression ratios for window sizes 8 and 16 is 0.02. This observation indicates that using a fixed size window with 16 records or more will not result in a substantially higher compression ratio. The last column of Table 4 indicates that, on the average, the variable window size compression outperforms the fixed window size compression with respect to the compression ratio.

5.2 Extracting Lead Drug Candidates

Given a computational model, one of the operations involved in extracting lead drug candidates from the database is a full database scan. This full database scan consists of, first, reading the windows from the compressed descriptor programs into memory. These windows are then decompressed, parsed in memory, and written to an output file. In order to reduce system variability, each test was conducted three times in a single user mode. The reported numbers are the average of the results of the three tests.

The execution time of the full database scan operation using varying size windows is shown in Table 5. The variable window has the lowest execution time as indicated by the last column of Table 5. For fixed window sizes, compressions based on large size windows lead to higher execution times than those based on small size windows. In general, Table 5 shows that there is very little difference between the execution time of a full database scan using varying window sizes. This result was expected since partitioning the descriptor programs into windows favors access to a partial set of records in the database. In the full database scan operation, all the records are retrieved and decompressed. Thus, the window size has very little effect on the execution time of the full scan of the drug candidate database.

5.3 Computational Model Validation

In order to validate the computational model corresponding to lead drug candidates for a given target disease, scientists issue queries against the drug candidate database. These queries consist of retrieving a set of descriptor values for a list of drug candidates. The throughputs of these queries are evaluated by two types of tests. In the first type, the drug candidate list includes drug candidate numbers selected randomly from the database. In the second type, the drug candidate list consists of consecutive drug candidate

desc. prog.	win=1	win=4	win=8	win=16	Variable
1	47.76	49.70	52.93	49.76	47.95
2	263.76	275.57	280.75	276.41	265.61
3	280.67	289.90	301.85	288.58	277.80
4	148.39	156.50	160.11	160.36	154.49
5	80.93	85.62	85.24	86.89	82.21
6	1191.52	1211.57	1187.28	1209.00	1186.46
7	534.73	550.87	528.41	534.49	522.24
8	12.05	12.17	12.09	12.35	12.83
9	49.99	51.16	49.90	50.57	49.04
10	71.32	76.04	75.62	76.79	71.65
11	2260.18	2489.35	2400.68	2415.69	2218.25
12	104.66	111.52	103.65	105.36	99.85
Tot.	5045.97	5359.97	5238.53	5266.24	4988.38

Table 5: Full database scan time in seconds for the compressed drug candidate database using window-based compression with varying window sizes.

numbers. In these input lists, the number of drug candidates is varied to include 5000, 10000 and 20000 drug candidates. The descriptor set contains 100 descriptors selected randomly from all the descriptor programs. The choice of 100 for the number of descriptors is driven by the fact that, on average, a drug model consists of 100 descriptors. In addition to varying the type and size of the drug candidate list, the impact of indexing the compressed drug candidate database on the throughput is also investigated.

For comparison purposes we show the throughput obtained when Oracle 8i is used to implement the drug candidate database (Table 6). The numbers in the second row represent the size of the drug candidate list. The throughput is measured in drug candidates per second, which is obtained by taking the ratio of the number of drug candidates in the input drug candidate list to the query execution time in seconds. The higher the throughput, the more efficient is the data access approach. This measure allows us to compare performance across drug candidate lists with different sizes.

Query Structure	Random drug candidate list		
	5,000	10,000	20,000
Regular query	18.87	26.60	34.66
Parallel query	29.24	38.31	45.87
Query Structure	Consecutive drug candidate list		
	5,000	10,000	20,000
Regular query	43.10	36.76	27.74
Parallel query	53.76	49.50	44.25

Table 6: Throughput in Oracle DBMS measured in drug candidates per second.

The regular query in Table 6 does not incorporate any advanced option provided by Oracle 8i, while the parallel query includes the PARALLEL option in the Oracle SQL command and uses multiple processes to execute each query. In both cases, the Oracle key compression was used.

5.3.1 Non-Indexed Database

In a non-indexed database, the compressed database is searched for each drug candidate in the drug candidate list; that is, the first line of each window is read and decompressed. If the drug candidate is in the current window, the entire window is decompressed, and the values of the target descriptors are retrieved. While at most one pass is performed on the database (the database is sorted), this operation is still time consuming if the size of the database is very large.

Window Size	Random drug candidate list		
	5,000	10,000	20,000
1	1.29	2.59	5.16
4	3.59	7.03	13.50
8	5.13	9.92	18.51
16	6.20	11.84	21.69
32	7.36	13.81	24.51
64	7.86	14.47	24.99
128	7.94	14.19	23.51
Variable	9.31	17.87	31.62
Uncompressed	2.08	4.16	8.26

Window Size	Consecutive drug candidate list		
	5,000	10,000	20,000
1	9.10	16.21	26.45
4	22.02	35.57	50.96
8	28.92	44.35	59.78
16	32.93	49.10	64.09
32	36.98	53.09	67.46
64	39.13	55.16	68.68
128	40.00	55.77	68.84
Variable	44.25	62.33	76.25
Uncompressed	14.37	25.42	41.59

Table 7: Throughput in the compressed drug candidate database measured in drug candidates per second.

Table 7 shows the throughputs of the above query for different drug candidate lists when the database is compressed. The last row of this table shows the throughputs of the query when issued against the database in its original uncompressed form. Unlike the case of the full database scan, the throughput for the compressed database is significantly higher than that of the original uncompressed database except when the window size is 1. Additionally, the variable window size for the compressed database (second to last row of Table 7) has the highest throughput for both the consecutive and the random drug candidate list.

For the case of consecutive drug candidate lists, the throughputs obtained by using Oracle (Table 6) are less than those obtained using the window based compression technique with window sizes larger than 16 for input candidate lists with 10,000 and 20,000 drug candidates (columns 6 and 7 of Table 7). However, all the throughputs for random drug candidate lists are less than those in Table 6. This difference in behavior is due to the spatial locality of data in the case of consecutive drug candidates (i.e., sequential ac-

cess to the database) and the lack of spatial locality of data in the case of random drug candidates. The lower throughput in the random access to the window-based compressed database can be mostly attributed to the use of indexes in the Oracle database. One factor that is not accounted for in this comparison is the time Oracle takes to parse and optimize the query. Including this time in the query execution time for Oracle puts the Oracle application at a slight disadvantage.

For the database compressed using window-based compression, the throughputs for the consecutive candidate list are on average three times higher than that of the random candidate list. This result can be explained by the fact that the consecutive candidate list contains drug candidates that are located one after the other in the drug candidate database (spatial locality). Thus, when the first record is identified, the remaining records follow. In the case of the random drug candidate list, the drug candidate numbers are scattered over the entire drug candidate database. The throughput for the random drug candidate list can be improved through the use of an index that points to the start of each window. This approach is discussed in the next subsection.

Table 7 shows that the throughputs increase steadily with an increase in window size. This improvement in throughput tapers off when the window size reaches 64. This behavior can be explained by evaluating the benefit of the difference operation within a given window. The difference operation is performed with respect to the first tuple in the window. Because the database is sorted, tuples that are directly adjacent to this first tuple are very similar to it. Whereas, there are fewer similarities between the tuples that are at the end of the window and the first tuple. This is particularly true for windows with large sizes, which explains the lack of steady improvement for window sizes greater than 64.

5.3.2 Indexed Database

In order to improve the throughput, in particular for random candidate lists, an index is added to the compressed database. Different types of indexes have been traditionally used in databases. These indexes include hash tables, B-tree (33), and Bitmap index (24). There are several other index structures for documents and images which are described in (38). This paper uses a one level linear index structure which consists of two columns: a *primary key* and an *offset*. The intent of this simple index is to demonstrate the use and potential impact of indexing on compressed databases. The offset is the relative address of the window in the data file. Each descriptor program in the drug candidate database is associated with two files, one for the data and one for the index.

In order to perform the query necessary to validate the computational model, the record corresponding to a given drug candidate from the input drug candidate list specified by the scientist is first located in the index file. Using the

relative address of the window containing this drug candidate, the corresponding window is decompressed in memory and the target descriptor values are retrieved. Table 8 shows the throughputs of this query for the compressed and indexed drug candidate database. For comparison purposes, the last row of this table shows the throughput for the case where the database is indexed but not compressed.

Window Size	Random drug candidate list		
	5,000	10,000	20,000
1	2.24	4.35	8.05
4	6.04	11.55	20.56
8	8.48	15.75	27.31
16	10.40	19.42	32.18
32	11.51	21.19	34.26
64	12.07	21.76	34.34
128	11.97	21.09	32.55
Variable	68.54	82.59	93.20
Uncompressed	20.63	47.58	78.92
Window Size	Consecutive drug candidate list		
	5,000	10,000	20,000
1	15.96	23.37	32.44
4	42.30	56.17	66.14
8	59.17	71.49	77.38
16	73.64	82.54	86.51
32	82.57	88.46	90.21
64	88.15	91.60	91.79
128	91.69	93.44	92.82
Variable	87.22	89.70	89.64
Uncompressed	96.71	101.68	102.29

Table 8: Throughput in drug candidates per seconds for the indexed and compressed drug candidate databases

Table 8 shows that the throughputs increase steadily as the window size increases. Furthermore, there is no significant improvement in throughputs for window sizes greater than 64. These two observations are similar to the ones derived from Table 7 for the non-indexed drug candidate database case. As in the case of Table 7, these results are due to the decrease in similarities between tuples for large size windows.

A comparison of tables 7 and 8 shows that indexing improves random access. This improvement is highest in the case of the variable window size with an improvement factor higher than 3. This result was expected since in a non-indexed database, the search algorithm spends a substantial amount of the execution time locating the windows corresponding to the drug candidates from the random drug candidate list which are scattered across the entire database. To a lesser extent, indexing also improves the throughput for the consecutive drug candidate list. The improvement factor in this case ranges between 1.2 and 2.3 (i.e., by comparing the last three columns of tables 7 and 8).

When indexing is used, all throughputs for the consecutive drug candidate list, except for window sizes are 1 and 4, are also higher than the corresponding throughputs obtained using Oracle (Table 6). As the database is com-

pressed, more data can be retrieved from disk and placed in memory. Thus, the compressed database exploits spatial locality better. In Oracle, the index is compressed but the data are not compressed. Therefore, more I/O time is needed to access the data. In this comparison, the time of query parsing and analysis in Oracle was also included in the query execution time which puts Oracle at a slight disadvantage.

The variable window throughput for the random drug candidate list is at least twice the throughput obtained with Oracle for both regular and parallel queries (Table 6). This is due to the combined effect of indexing and compression. Each drug candidate number in the compressed database has a unique entry in the index file. Thus, the query can easily locate the requested records from the random drug candidate list. Additionally, compression reduces the time dedicated to I/O during the execution of the query.

The final observation that can be derived from Table 8 is in regard to the last two rows (Variable and Uncompressed) of the table. The variable window size has the highest throughput for the random drug candidate lists. Furthermore, the uncompressed and indexed database (last row of Table 8) has higher throughputs for the consecutive drug candidate list than the compressed and indexed database (second to last row of Table 8). This fact shows that if the most frequent type of access to the database is sequential, then compression should not be used. Indexing is sufficient; however, if the application requires random access to the database, then both compression and indexing should be used. In the case of the drug candidate database, both sequential and random access to the database are required. Duplicating the database is not a viable solution in a business environment because it is prone to inconsistencies in the database instances. Thus, the compressed and indexed database is the choice database implementation. This database implementation benefits from high throughputs for random access to records (performance gain as high as 3 for variable size windows) with limited loss in performance for the sequential access to records compared to the indexed uncompressed database (performance loss of less than 13%).

5.4 Summary

The experiments presented in this section show that the “compress” and “gzip” techniques achieve higher compression ratios than the window-based compression. However, they can not efficiently support random access the data in the drug candidate database. Under the window-based compression, large window sizes result in higher compression ratios, and the variable window size leads to the best compression ratio and full database scan time. When indexing and compression are used together, higher throughputs are achieved for random access with limited penalty for sequential access.

6 Update Implementation

The results of the previous sections were presented under the assumption that the database is read only. However, for the drug candidate database, although infrequent, operations such as insertions, deletions, and updates are used. One possible solution to accommodating these operations consists of decompressing the entire database, making the updates, compressing the database, and generating a new index. This solution is not efficient because it is time consuming.

An alternative solution is a deferred update policy which has been investigated in (13; 28; 36). The new data to be inserted, deleted, or updated are first organized in a temporary relation, and the database is updated off-line in batch mode. When this update approach is used, the query consists of two parts: database query and temporary relation query. The concept of deferred update has also been previously studied for client-server database systems (9) and for distributed database systems (7). In this paper a write-back disk buffer is used to hold the data that is associated with pending insertions, deletions or updates.

Each tuple in the buffer has four entries: drug candidate, descriptor number, descriptor value and flag. The flag can have one of three values: I (insertion), D (deletion) and U (update). The flag will be used in two cases: (1) to make sure that the data accessed is the most recent one, (2) to merge two tuples. The latter case occurs when there are two tuples in the buffer with the same drug candidate number and descriptor number. In this case the older tuple will be discarded and the most recent one will be retained.

A dirty bit is also added to the index file. The dirty bit is combined with the offset. Initially when the index is created for a drug candidate number, the dirty bit for each drug candidate is set to “+”. If an update occurs to a drug candidate, the dirty bit of the corresponding drug candidate will be set to “-”. When all the tuples related to a given drug candidate are deleted from the buffer, the dirty bit of the drug candidate will be reset to “+” in the index file. If a new drug candidate is inserted into the buffer, a new index tuple for this particular drug candidate is added to the index file where its dirty bit is “-” and its offset equals to 0.

If the buffer reaches a user-defined threshold size or expiration time, the data in the buffer is added to the database and the buffer is cleared. For a compressed database using a variable window size, all the tuples are read from the buffer into memory. For each tuple to be inserted, the window where the tuple will reside within the compressed database is located. This window is decompressed and the tuple is added to the window. The window is compressed again and its new offset is calculated. Finally the new window is written into the compressed database. This is accomplished by reading ahead a few windows into memory and then incrementally updating the old window. For fixed window size compression, the algorithm is complicated because some windows may be expanded and some may be reduced.

For the deletion of tuples, approaches similar to the ones

discussed for insertion are possible. One solution is to replace the tuples to be deleted with empty place holder tuples so that the other windows will not be affected. The update operation does not change the window size and thus its implementation is trivial.

In addition to modifying the data, the index file needs to be updated. The offset of a window has to be recalculated if an update occurs to a given window. Furthermore, once all the data is transferred from the buffer to the database and the buffer is cleared, all the dirty bits in the index file will be reset to “+”.

In Section 5, throughputs were measured with respect to a read only database. In the remainder of this section, the impact of adding a buffer on the throughput is investigated. The buffer is read prior to the execution of the query. However, the time needed to read the buffer is included in the query execution time, thus, providing a conservative evaluation of the gain afforded by the window-based compression technique for large databases.

Figures 2 and 3 show the throughput achieved in the presence of varying buffer sizes when the database is compressed with variable size windows. The number of tuples in the buffer is used to quantify its size. A buffer size of 0 means that a buffer is not used. The two figures show that the performance is sensitive to the buffer size. Larger buffer sizes result in a lower throughput.

7 Conclusions

Using compression in a very large drug candidate database can increase throughputs for random accesses to the database, which in turn may lead to increased productivity and early drug discoveries. This paper proposes a window-based compression technique that supports high throughput for random access while preserving competitive throughput for sequential accesses. The proposed compression technique takes advantage of the high level of similarity among tuples in the drug candidate database. The throughputs obtained with window-based compression were higher than those obtained with Oracle using the Oracle key compression approach. Compared to Oracle, the implementation environment used in this paper does not include the overhead of query parsing and analysis. However, the major portion of the difference in throughput can only be attributed to the use of window-based compression.

Insertion, deletion and update operations are handled using a buffer that implements a deferred update policy. This update policy was selected because it has limited overhead. The variable size window has the highest throughput and it is the easiest to implement particularly when insertion, deletion and update operations are considered.

In a different study, performance enhancing techniques for large databases were evaluated (2). These performance enhancing techniques included indexing, partitioning, window-based compression, and load balancing. Among these techniques, window-based compression and

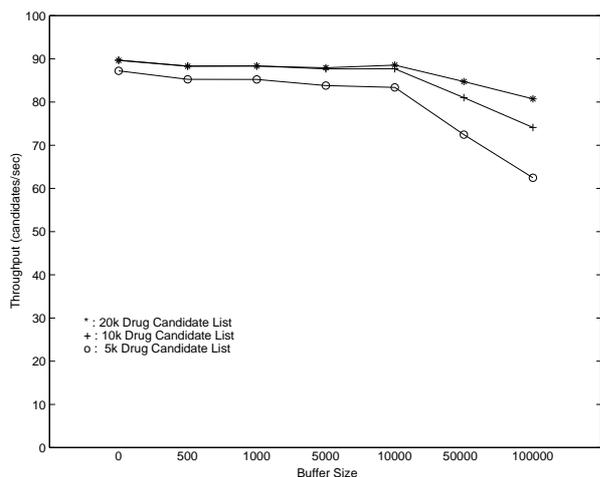


Figure 2: Throughput for consecutive drug candidate lists in the compressed and indexed database when an update buffer is used.

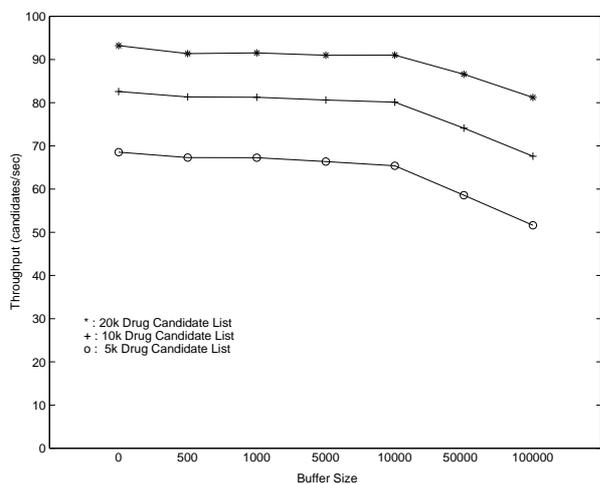


Figure 3: Throughput for random drug candidate lists in the compressed and indexed database when an update buffer is used.

indexing resulted in the highest increase in performance for random access to the drug candidate database.

The impact of schema transformation (21) and database instance scalability in the case of the drug candidate database were also investigated (3). In particular, these studies showed that one of the major bottleneck in random access to the large drug candidate database is I/O. This bottleneck becomes more important as the database instance increases and persists even after an optimal schema transformation is applied to the database. In this paper, it was shown that window-based compression can reduce the effect of this bottleneck.

Future extensions of this research include evaluating the window-based compression technique in conjunction with other indexing and encoding methods.

Acknowledgements

This research was supported in part by Eli Lilly and

Company.

References

- [1] Balaban, A.T. Highly Discriminating Distance-Based Topological Index. *Chemical Physics Letters*, 89(5):399–404, 1982.
- [2] Ben Miled, Z., Liu, Y., Neoh, C., Bukhres, O., Bem, M., Jones, R., and Oppelt, R. A Comparative Study of the Impact of Performance for Large Scientific Database. In *Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, Las Vegas, Nevada, June 2002.
- [3] Ben Miled, Z., Liu, Y., Powers, D., Bukhres, O., Bem, M., Jones, R., Oppelt, R., and Milosevich, S. Data Access Performance in a Large and Dynamic Pharmaceutical Drug Candidate Database. In *Proceedings of Supercomputing 2000*, Dallas, Texas, November 2000.
- [4] Ben Miled, Z., Liu, Y., Powers, D., Bukhres, O., Bem, M., Jones, R., and Oppelt, R. An efficient implementation of a drug candidate database. *to appear in the Journal of Chemical Information and Computer Sciences*, 2002.
- [5] Bruni, P. and Naidoo, R. DB2 for OS/390 and Data Compression. *IBM Redbook, SG24-5261-00*, 1998.
- [6] Cannane, A. and Williams, H.E. A Compression Schema for Large Databases. *Australasian Database Conference*, pages 6–11, 2000.
- [7] Chundi, P., Rosenkrantz, D. J., and Ravi, S. S. Deferred Update Protocols and Data placement in Distributed Databases. *12th International Conference on Data Engineering*, pages 469–476, New Orleans, LA, February 1996.
- [8] Cormen, T. H., Leiserson, C. E., and Rivest, R. L. *Introduction to Algorithms*. McGraw Hill, NY, 1998.
- [9] Delis, A. and Roussopoulos, N. Management of Updates in the Enhanced Client-Server DBMS. *14th International Conference on Distributed Computing Systems*, pages 326–334, Poznan, Poland, June 1994.
- [10] Elias, P. Universal Codeword Sets and Representations of the Integers. *IEEE Transactions on Information Theory*, 21(2):194–203, 1975.
- [11] Goldstein, H., Ramakrishnan, R., and Shaft, U. Compressing Relations and Indexes. *14th International Conference on Data Engineering*, pages 370–379, Orlando, FL, February 1998.
- [12] Golomb, S. W. Run-Length Encodings. *IEEE Transactions on Information Theory*, 12(3):399–401, 1966.

- [13] Gray, J., McJones, P., Blasgen, M., Lindsay, B., Lorie, R., Price, T., Putzolu, G., and Traiger, I. The Recovery Manager of System R Database Manager. *ACM Computer Survey*, 13(2):223–242, June 1981.
- [14] Hoque, A.S.M.L., McGregor, D.R., and Wilson, J. Database Compression Using an Offline Dictionary Method. *ADVIS*, pages 11–20, 2002.
- [15] IBM. ESA/390 Data Compression, Second Edition. *IBM Redbook, SA22-7208-01*, 1996.
- [16] Iyer, B. R. and Wilhite, D. Data Compression Support in Databases. *20th International Conference on Very Large Data Bases*, pages 695–704, Santiago de Chile, Chile, September 1994.
- [17] Jagadish, H. V., Madar, J., and Ng, R. T. Semantic Compression and Pattern Extraction with Fascicles. *25th International Conference on Very Large Data Bases*, pages 186–197, Edinburgh, Scotland, September 1999.
- [18] Korn, F., Jagadish, H. V., and Faloutsos, C. Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences. *1997 ACM SIGMOD International Conference on Management of Data*, pages 289–300, Tucson, AZ, May 1997.
- [19] Langdon, G. G. An Introduction to Arithmetic Coding. *IBM Journal of Research and Development*, 28(2):135–149, March 1984.
- [20] Lekatsas, H. and Wolf, W. Random Access Decompression using Binary Arithmetic Coding. *IEEE Data Compression Conference*, pages 306–315, Snowbird, UT, March 1999.
- [21] Liu, Y., Ben Miled, Z., Bukhres, O., Bem, M., Joses, R., and Oplet, R. Efficient schema design for a pharmaceutical data repository. In *Proceedings of the 13th IEEE symposium on Computer-Based Medical Systems*, pages 247–254, Houston, Texas, June 2000.
- [22] Manber, U. A Text Compression That Allows Fast Searching Directly in the Compressed File. *ACM Transactions on Information Systems*, 15(2):124–136, April 1997.
- [23] Ng, W. and Ravishankar, C. V. Block-Oriented Compression Techniques for Large Statistical Databases. *IEEE Transactions on Knowledge and Data Engineering*, 9(2):314–328, March/April 1997.
- [24] O’Neil, P. and Quass, D. Improved query Performance with Variant Indexes. *1997 ACM SIGMOD International Conference on Management of Data*, pages 38–49, Tucson, AZ, May 1997.
- [25] Oracle. Oracle8i Concepts Release 8.1.5. http://technet.oracle.com/docs/products/oracle8i/doc_index.htm.
- [26] Ray, G., Haritsa, J. R., and Seshadri, S. Data Compression: A Performance Enhancement Tool. *International Conference on Management of Data*, Pune, India, December 1995.
- [27] Roth, M. A. and Van Horn, S. J. Database Compression. *ACM SIGMOD Record*, 22(3):31–39, September 1993.
- [28] Stonebraker, M., Wong, E., Kreps, P., and Held, G. The design and implementation of INGRES. *ACM Transactions on Database Systems*, 1(3):189–222, September 1976.
- [29] Storer, J. A. *Data Compression: Methods and Theory*. Computer Science Press, Rockville, MD, 1988.
- [30] Stuper, A. J., Brugger, W. E., and Jurs, P. C. *Computer-Assisted Studies of Chemical Structure and Biological Function*. Wiley-Interscience, New York, 1979.
- [31] Sybase. Sybase IQ White Paper. <http://www.sybase.com/products/dataware/iqw-paper.html>.
- [32] Tunstall, B. P. Synthesis of Noiseless Compression Codes. *PhD thesis*, 1968.
- [33] Ullman, J. D. *Principles of database systems*. Computer Science Press, Rockville, MD, 1982.
- [34] Wallace, G. K. The JPEG Still Picture Compression Standard. *Communications of the ACM*, 34(4):30–44, April 1991.
- [35] Westmann, T., Kossmann, D., Helmer, S., and Morkotte, G. The Implementation and Performance of Compressed Databases. *ACM SIGMOD Record*, 29(3):55–67, September 2000.
- [36] Whang, K. and Krishnamurthy, R. Query optimization in a memory-resident domain relational calculus database system. *ACM Transactions on Database Systems*, 15(1):67–95, March 1990.
- [37] Wild, D. and Blankley, C. J. Comparison of 2D Fingerprint types and hierarchy level selection methods for structural grouping using Ward’s clustering. *J. Chem. Inf. Comput. Sci.*, 40:155–162, 2000.
- [38] Witten, I. H., Moffat, A., and Bell, T. C. *Managing Gigabytes*. Van Nostrand Reinhold, NY, 1994.
- [39] Ziv, J. and Lempel, A. A Universal Algorithm for Sequential Data Compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977.
- [40] Ziv, J. and Lempel, A. Compression of Individual Sequences via Variable-rate Coding. *IEEE Transactions on Information Theory*, 24(5):530–536, 1978.

Modelling Legal Acts by Means of Expert Systems, ECA Rules and High-level Petri Nets

Boštjan Berčič

Institute for Legal Informatics, 1000 Ljubljana, Slovenia

E-mail: bostjan.bercic@ipri-zavod.si

Keywords: expert systems, petri nets, legal informatics

Received: May 7, 2002

Modelling of legal acts often felt short of expectations because it didn't take into account legal theory. This paper proposes a different approach to modelling that is based on the theory of law. Legal theory (structure, hierarchy and types of legal rules) is considered a fundament and then interpreted with expert systems, high-level Petri nets and ECA rules. In particular, none of these methods alone is sufficiently strong to capture the semantics of legal rules. Put together, they represent a powerful means to overcome some difficulties legal modelers have encountered in the past. This paper takes into account procedural and substantive aspects of law, as well as factual and deontic ones. Methodology is presented alongside with notation and examples to clarify the idea.

1 Introduction

Modelers of legal procedures in the past often came from the field of informatics and had little knowledge of legal theory. They applied same techniques and methodologies to legal procedures as they would to business (or indeed any other) processes. Whereas this might work in certain cases, it certainly comes at a price: legal procedures are *sui generis* and not every legal rule can be expressed with business process methodology. There are complexities and peculiarities inherent to legal matter which can hardly be modelled by means of business processes.

For one thing, legal norms contain normative (deontic) and factual elements. Most modelers in the past focused their work on factual elements only. They made little or no reference to normative elements such as rights and duties.

Second, their approach was directed either at substantive or procedural law, but rarely at both. Substantive law defines contents of legal subject's rights and duties whereas procedural law defines procedures in which these rights and duties can be enforced in case legal subjects do not act in conformance with them.

Modelers dealt in the past mostly with procedural law, which can be modelled with state machines, workflow methods etc. Very often, one of these techniques was used to represent procedural part of legal rules as one can in general apply knowledge of (any kind of) processes to legal procedures.

On the other hand, modelers have often encountered problems with substantive legal norms. Technology of choice for substantive legal rules are expert systems. There are many legal expert systems shells available (such as Wysh, IcaR etc.) but general purpose expert system shells can be used as well. Legal rules are written in knowledge base, usually in form of PROLOG rules.

Some of these approaches are summarized bellow. They have been partial at least in some respect because none of them joined all (legal) elements under one hood.

Holt and Meldman (Meldman J.A., Holt A.W. (1971), Meldman J.A. (1978)) used Petri nets to model Federal Rules of Civil Procedure. They modelled only procedural rules and took into account only factual elements.

Sergot et al. (Sergot M.J., Sadri F., Kowalski R.A., Kriwaczek F., Hammond P., Cory H.T. (1986)) used PROLOG rules to express British Nationality Act (rules governing acquirement of British citizenship). These are substantive rules, but no mention is made of deontic elements (for example, what does it *mean* to be a British citizen in terms of rights and duties).

Lee et al. (Lee R.M., Bons R.W.H., Wrigley C.D., Wagenaar R. W. (1995)) used special kind of high-level Petri nets, called documentary Petri nets, to express procedural rules. Documentary Petri nets are high-level Petri nets augmented with deontic operators expressing obligations (duties) and rights (permissions) and special documentary places holding documents. Thus, they came a step closer in embedding factual and normative (deontic) contents within procedural framework. Separately, Lee et al. (Lee R. M., Ryu Y.U. (1994)) also considered deontic expert systems, with deontic operators augmented first-order logic (PROLOG knowledge base), but they have not integrated it in a single framework.

Burg and Van de Riet (Burg J.F.M., Van de Riet R.P. (1994)) devised a modelling technique called COLOR X. It can model procedural aspects of (deontic) rules with use of linguistics. They went a step further in introducing linguistic elements to modelling. Now, a fact is not merely a fact but a sentence in natural language which can be stripped down to words. Meaning of a word can be fully interpreted as every word can have its own existence. This

technique is object-oriented and could form basis for legal ontology building if extended a little bit further. Technique prescribes dynamic (CEM) as well as static (CSOM) model of events and objects. This technique can model not just legal domains but, because of its root in linguistics, any domain expressed in natural language. So, COLOR X can represent factual as well as deontic elements, but it focuses on procedural aspects. Substantive knowledge is left out.

This paper proposes a different approach. Our models are based on legal theory. First, legal structures (legal acts, legal rules, hierarchy of legal rules and acts etc.) are taken into account and interpreted with different information technologies and notations. High-level Petri nets are used to represent procedural aspects of legal rules. Expert system knowledge base is used to represent substantive legal rules. Factual elements of legal rules are mapped to Petri net transitions and deontic elements to Petri net places. Petri net transitions are extended with ECA rules in order to be able to express factual and deontic pre- and post-conditions.

2 Legal Rules

2.1 Structure of Legal Rules

Legal rules are composed of three (optionally four) components: the norm addressee (norm subject), deontic modality, object of a norm (contents) and optionally norm conditions (Kralingen van R. (1997), Breuker J., Valente A., Winkels R. (1997), Visser P.R.S., Bench-Capon J.M. (1997)).

Norm addressee is a subject addressed by the norm. Usually certain act or behavior is required from him.

There are two principle deontic modalities: obligation O and permission P which correspond to legal duties and legal rights. Every norm either prescribes or permits some behavior.

Object or theme of a norm is an act or behavior required from or allowed to norm subjects. Object of a norm is norm's contents. It prescribes what is allowed to do and what should be done.

Condition of a norm is norm's hypothesis. It describes state of affairs which must be satisfied in order for the norm to apply. Some norms have conditional part, some don't. Those that don't are unconditional. Conditional norms turn into unconditional ones once the condition is fulfilled.

We can write this succinctly as:

$$F_1(X) \Rightarrow P(F_2(X)) \quad (1)$$

where:

- X is norm subject
- $F_1(X)$ is norm condition
- $P(\dots)$ is deontic modality and
- $F_2(X)$ is object of a norm

If we take a closer look, we discover that there is a difference between norm's condition and norm's consequence. Norm's condition is *always* some state of affairs (legally relevant state of the world) and norm's consequence is *always* some deontic modality (defined on yet another, prescribed state of affairs), either obligation or permission.

2.2 States of Affairs

States of affairs are important elements of legal rules. They define legally relevant states of the world, i.e. states of the world that are of interest to legal order.

Legal rules feature states of affairs in both conditional and prescriptive part. States of affairs describe under which circumstances legal rules are applicable (rule condition) and what behavior is required or allowed. We can find them both in

$$F_1(X) \quad (2)$$

and in

$$P(F_2(X)) \quad (3)$$

States of affairs are often subject to different methods of legal exegesis.

2.3 Rights and Duties

Rights and duties (permissions and obligations) form prescriptive part of legal rules. They are deontic modalities and they cannot stand alone. They come in two flavors: O and P and they are *always* defined on some state of affairs.

$$O(F_1(X)) \quad (4)$$

$$P(F_2(X)) \quad (5)$$

In the first case (equation 4) person X is required to bring about state of affairs F_1 .

In the second case (equation 5) person X is allowed to bring about state of affairs F_1 .

Rights and duties can be of two types: *ought-to-do* and *ought-to-be* (*tun-sollen* and *sein-sollen*). Ought-to-do operators have subjects that they address whereas ought-to-be don't. This paper will deal only with first type operators.

2.4 Formal notation

So far, all components of legal rules have been formally defined. But rules are often interconnected. If legal rule is not obeyed there is usually another one that specifies what legal order should do in order to preserve rule conformity. This rule is called sanction.

This can be written as:

$$hypothesis \Rightarrow duty \quad (6)$$

$$secondary\ hypothesis \Rightarrow secondary\ duty \quad (7)$$

or, in common terms as:

$$\text{hypothesis} \Rightarrow \text{duty} \tag{8}$$

$$\text{duty} \wedge \text{violation of duty} \Rightarrow \text{sanction} \tag{9}$$

On general, if one denotes state of affairs with F(factual) and normative contents with N, one gets the following structure of legal rules:

$$\begin{aligned} F_1 &\Rightarrow N_1 \\ N_1 \wedge F_2 &\Rightarrow N_2 \\ N_2 \wedge F_3 &\Rightarrow N_3 \end{aligned} \tag{10}$$

Keyword here is structure. Legal rules don't stand alone. They are intertwined with each other in legal order. One rule's consequence may be another one's condition. If all rules from one legal system are put together in this manner, we obtain *legal order* in force.

Notice here the strict alternation of factual and deontic elements. According to legal theory, legal rules can have factual antecedent and factual consequence (e.g. legal definition) or even deontic antecedent and deontic consequence (another legal definition, e.g.: having one right means that you have another one, too). But legal rules can never have deontic antecedent and factual consequence (what should be is not the case by mere fact that it should be).

If we use our previous notation, we can now write legal order as:

$$\begin{aligned} F_1(X) &\Rightarrow P(F_2(X)) \\ P(F_2(X)) \wedge F_2(X) &\Rightarrow O(F_3(X)) \\ O(F_3(X)) \wedge \neg F_3(X) &\Rightarrow O(F_4(X)) \\ &\vdots \end{aligned} \tag{11}$$

2.5 Types of Legal Rules

2.5.1 Substantive vs. Procedural Rules

Substantive rules express contents of rights and duties that address legal subjects. They prescribe how legal subjects are to act in order to achieve desired results and what different states of affairs imply in terms of normative consequences.

Procedural rules, on the other hand, express procedural aspects of law enforcement. They prescribe procedures in which substantive rules can be enforced. Procedural rules govern legal processes such as criminal procedure and civil procedure which seek to remedy breaches (civil law) and crimes (penal law) committed by subjects of legal norms.

2.5.2 Legal Rules vs. Legal Definitions

Legal rules prescribe behavior that is required from norm addressees. Legal rules have conditional part and prescribed part. Conditional part contains description of states

of affairs to which norm applies. Consequential part prescribes deontic modality (right,duty) and act, which is to be (duty) or is allowed to be (right) performed by the norm subject. Unconditional legal rules contains only second part. It is of constitutive importance for legal rules to contain normative (deontic) contents.

$$F \Rightarrow N \tag{12}$$

Legal definitions, by contrast, do not prescribe any behavior but rather define some legal term. Legal definitions do not contain hypothetical (conditional) part which would describe when to apply the norm, nor do they contain normative elements which would say what behavior is required. Rather, legal definitions apply to legal terms themselves (not states of affairs) and further clarify their meaning (e.g. somebody falling under provisions of certain law means that ...).

$$F \Leftrightarrow F_1 \wedge \dots F_n \tag{13}$$

Legal definitions can apply to legal terms which contain states of affairs and legal terms which contain deontic modalities.

2.5.3 Rules of Conduct vs. Rules of Competence

There is an important difference between rules of conduct, which describe what is permitted to do or must be done by the norm addressees in terms of factual behavior and rules of competence which describe what powers (liabilities) norm subjects have in respect to creating new legal rules.

Most rules are rules of conduct. They prescribe required behavior from norm subjects.

Rules of competence, on the other hand, empower or oblige norm subjects to create new legal rules. A legal subject can be empowered to create new normative contents (rights and duties, e.g. legislator who passes laws or contractor who creates new obligations). Legal subject can also be obliged to create new rules (e.g. if you apply for citizenship, under some conditions, public administration office has to issue it)(Allen L.E. (1997)).

2.5.4 Monotonic vs. Nonmonotonic Rules

Monotonicity is defined with respect to temporal aspects of legal rules.

Monotonic rules are those that hold regardless of the event (events) that triggered them (e.g. once somebody is dead, his heirs are entitled to heritage).

Nonmonotonic rules are those, whose validity constantly depends on validity of the triggering conditions (e.g. when you get sick you can apply for remedies from your health care insurance policy, but you have no right to do so while healthy).

Monotonicity of rules has to do with repeatability of norm's conditional part. If states of affairs that describe

conditional part can vary over time, then we have non-monotonic rule. If, on contrary, it is a one-time event, then we have monotonic rule.

The same applies for truth values of legal definitions. Nonmonotonic conditions can hold over intervals of time and change (e.g. marital status can change several times in one's lifetime).

Some procedural rules are typically monotonic while some substantive rules are typically nonmonotonic.

3 Expert Systems

Expert systems contain expert system shell and knowledge base. Knowledge base is used as an input to inference mechanism, which resides in expert system shell and infers consequences. Knowledge base can be expressed in PROLOG in form of Horn clauses (clauses with implicit existential quantifier)(Bratko I. (2001)):

$$\Phi : -\Psi_1, \dots, \Psi_n \quad (14)$$

$$\Phi : -\Psi_1; \dots; \Psi_n \quad (15)$$

These PROLOG clauses can be conveniently expressed in more familiar form:

$$\Phi \Leftarrow \Psi_1 \wedge \dots \wedge \Psi_n \quad (16)$$

$$\Phi \Leftarrow \Psi_1 \vee \dots \vee \Psi_n \quad (17)$$

Expert systems lend themselves to express nonmonotonic rules (rules whose truth values change over time because of changing states of affairs). Expert system can be interrogated by user in every moment about truth values of consequences of rules in knowledge base. If rule's conditions are satisfied, system infers its consequences.

Expert systems have been mostly used in substantive law. They can be used in legal definitions:

$$F \Leftrightarrow F_1 \wedge \dots \wedge F_n \quad (18)$$

where:

- F is complex state of affairs
- F_1, \dots, F_n are elementary states of affairs

and in legal rules as well:

$$N \Leftarrow F_1 \wedge \dots \wedge F_n \quad (19)$$

where:

- N is norm object (of normative type)
- F_1, \dots, F_n are norm conditions (of factual type)

For the most part, this paper will refer to expert systems' knowledge base for expressing factual elements of legal rules (states of affairs). Although deontic elements could be expressed as well (and sometimes they will be), it is probably better to have a clear delineation on the level of implementation between states of affairs and their deontic consequences. Deontic consequences will be expressed by means of places in Petri Nets.

4 ECA Rules

ECA rules have general form (Dittrich K.R., Gatzju S. (1993)):

```
on <event>
if <condition>
do <action>
```

ECA rules stem from active database community and are today most widely used in active database research and event-driven programming.

Event detector detects events, checks whether conditions hold and if they do, fires corresponding action.

Events in ECA rules can be of simple and composite types; composite types can be, for example, expressed with rules in knowledge base. The same holds for conditions which are typically expressed as boolean combination of simple conditions read from some database (or knowledge base). ECA rules can thus be easily integrated with expert system's knowledge base.

ECA rules will be used to express various elements of states of affairs. States of affairs typically contain either some state or event or both. Event part of ECA rules will be used to specify events of some complex state of affairs and condition part of ECA rules will be used to express some static state of affairs. Example will clarify this:

If somebody wants to enter into a contract, he must perform some action (e.g. sign a contract). But in order for the contract to be valid, contactor must have competence to enter into it. Competence to sign a contract is static part and signing a contract active part of the state of affairs which must be fulfilled in order for the contract to be valid. Thus, we map signing a contract into event part and legal competence into condition part of ECA rule. Both should be there if the event is to trigger some consequences. Note that this paragraph doesn't deal with deontic consequences of these acts, which is left for later when Petri nets are considered).

Finally, the action part (which is optional) may express some change in the state of affairs (for example, if you marry someone, your marital status changes to married).

4.1 Events

Events can be primitive or composite events. Composite events are made up of primitive ones with the use of operators (disjunction, sequence, conjunction, periodicity) of event algebra (Gatzju S. (1993), Chakravarthy S., Mishra

D. (1991)) . Event algebra itself could be specified with Petri nets or with rules in knowledge base (boolean operators plus attribute for time). This paper implements it with knowledge base in order to have expert systems cover all factual elements.

4.2 Conditions

Conditions express conditions which must hold in order for the event to fire. Conditions can be any combination of first-order predicate logic statements. They are implemented with expert systems, as well.

4.3 Actions

Actions are an optional part of an ECA rule in our interpretation. They can express an update or change in the state of affairs. Actions always refer to state of affairs (matters of fact), never to deontic modalities (rights and duties). Petri net markings take care of the latter.

5 High-level Petri Nets

A HLP-net is a structure $HLPN = (P; T; CT; C; Pre; Post; M_0)$ (Billington J. (1997)) where:

- P is a finite set of elements called Places,
- T is a finite set of elements called Transitions, which are disjoint from P ($P \cap T = \emptyset$),
- $CT = \{N, F\}$ is a non-empty finite set of types (of places and transitions), where N denotes normative type and F denotes factual type
- $C : P \cup T \rightarrow CT$ is a function used to type places and determine transition modes, such that $C(P) = N, C(T) = F$,
- Pre is a pre mapping $Pre_{(p,t)} : C(t) \rightarrow N^{|C(p)|}$,
- Post is a post mapping $Post_{(p,t)} : C(t) \rightarrow N^{|C(p)|}$,
- M_0 is an initial marking of the net.

HLPN lend themselves to express procedural law.

They can be data(place) or transition driven. Data driven Petri nets fire transition whenever its preconditions are met (all input places contain tokens). Event driven Petri nets fire transition whenever its preconditions are met *and* event associated with it has occurred.

Event driven Petri nets will be used in this paper because states of affairs will be mapped to events in Petri net transitions.

Petri nets are most appropriate for procedural rules (which contain implicit timeline). Sometimes they can be used for substantive rules as well.

They are also appropriate for monotonic rules, i.e. rules which don't change their truth values over time. Their truth

space is monotonically increasing (once you file a complaint it will always be that you have filed it)

Petri net places and transitions are interpreted as follows.

5.1 Places

Places hold deontic contents (rights, duties). They are the only elements in whole structure that do so. Rights and duties are not expressed within knowledge base in this paper. Rather, they are all gathered at one place (consider implementation issues).

5.2 Transitions

Transitions hold events and are always of factual nature. They contain events that change rights and duties. In accordance with Petri net semantics, whenever event fires and its preconditions hold, postcondition hold after the event and preconditions stop to hold (e.g. if you have certain right and choose to exercise it, you lose that right (precondition) and possibly get another one (postcondition)).

Transition events can be simple, composite or empty. Composite events are made up of simple events. Sometimes transitions can synchronize: if one fires, the others fire, too. This happens if one event is mapped onto many transitions.

Transitions in HLPN can fire in different modes. Some priority function must be defined over modes, just as some priority function must be defined over ordinary transitions if they happen to be enabled at the same time. We say that they are in conflict.

5.3 Step Semantics

Step semantics determines in detail what effect firing of a transition has. It prescribes detailed sequence of steps taken by the system in order to arrive at desired state (determination of enabled transitions and modes, retrieval of relevant data, updates to net marking, etc.)

Check which transitions are enabled in what modes. For each transition and for each mode do:

1. resolve mode and transition priority
2. check whether transition contains event or null event
 - (a) if it contains event, check whether event has happened and if so: fire
 - (b) if it contains null event: fire
3. withdraw tokens from appropriate input places
4. put tokens into appropriate output places

6 Petri Nets, Expert Systems and ECA Rules Integrated

Now we can make use of all three components in one integrated structure. HLPN will be taken as a starting point. Then, ECA rules will be mapped onto transitions, making them event/condition/action transitions. Event, condition and action parts of ECA rules will be furthermore mapped into expert system knowledge base rules that will be processed via expert system's shell inference mechanism. This will enable the expression of complex events, conditions and actions.

A HLP-net is a structure $HLPN = (P; T; CT; C; Pre; Post; M_0)$ where

- P is a finite set of elements called Places
- T is a finite set of elements called Transitions disjoint from P ($P \cap T = \emptyset$)
- $CT = \{N, F\}$ is a non-empty finite set of types (of places and transitions), where N denotes normative type and F denotes factual type
- Γ is a mapping $\Gamma : T \rightarrow \{E, Co, A\}$
- $C : P \cup E \cup Co \cup A \rightarrow CT$ is a function used to type places and transitions such that $C(P) = N, C(E), C(Co), C(A) = F$
- Pre is a pre mapping $Pre_{(p,t)} : C(t) \rightarrow N^{|C(p)|}$
- Post is a post mapping $Post_{(p,t)} : C(t) \rightarrow N^{|C(p)|}$
- M_0 is an initial marking of the net
- Δ is a mapping $\Delta : E \rightarrow \Phi : -\Phi_1, \dots, \Phi_n$
- Λ is a mapping $\Lambda : C \rightarrow \Psi : -\Psi_1, \dots, \Psi_n$
- Σ is a mapping $\Sigma : A \rightarrow \Omega : -\Omega_1, \dots, \Omega_n$

High-level Petri net defines the global structure of the legal model. It contains places and transition. Places are of two (deontic) types: rights P and duties O. Transitions can fire in different modes and they contain states of affairs. Just as places and transitions strictly alternate in Petri net, so do factual and deontic elements in legal order. States of affairs trigger normative contents (rights and duties). It is never *vice versa*

Transitions represent states of affair. States of affair can be simple or composite. They can also be proper states or events.

This semantics is captured by ECA rules. Events part represent (potentially composite) active components of states of affairs and conditions represent (potentially composite) proper states of affairs. Action part permits rules to issue actions such as update on states of affairs.

Transitions are mapped to ECA rules, which are triples $\{E, Co, A\}$. All members of triple are of factual type. They contain events that trigger the transition, conditions

which implement guard function as to when transition is allowed to fire and action which can be set off as a consequence of firing of transition.

Composition of events, conditions and actions in ECA rule is done by means of expert system rules.

Composite event can be described in usual knowledge base manner. Composite event name is head of the knowledge base rule, simple events which constitute composite events are its tail, coupled with corresponding boolean operators.

The same goes for conditions. Conditions are rules in knowledge base. Rule name (condition name) is head of the rule and represents consequence, whereas tail contains simple conditions.

Each time that system has to check whether certain event or condition is fulfilled (for example in order to fire a transition, the system asks user whether this or that has happened), it calls expert system knowledge base. Each ECA rule contains rule head, which is matched against rule head in knowledge base. Head is then expanded with rule body (tail) which in turn contains heads of other rules in knowledge base. This process continues iteratively until list contains only elementary facts that can be matched either against present knowledge base or (in case of absence) required from the user.

After all required elementary facts have been retrieved, rule can be evaluated to be either true or false (alternatively we could have any type of function, not just booleans, which would operate on supplied data). If both event part is true (composite event has happened) and condition part is true (required conditions hold), the transition can fire.

Firing a transition means subtracting tokens from its input places and putting them in its output places. This can be done after the event and when the condition part has been evaluated to be true. Also, at the same time, action part of ECA rule fires.

In legal semantics presented in this paper firing of transition means, that a legal event has happened (e.g. a person has committed an act), with all its normative preconditions (places) present (e.g. right of a person to commit that act) and with all its factual conditions present (e.g. person's competence to commit that act). Consequences are twofold: normative and factual. Normative consequences are new rights and duties which result out of act of a person committing that act (e.g. contractual obligations arise from signing the contract). Sometimes factual consequences arise as well (e.g. date of the contract is set). Normative consequences are represented by tokens in PN places, factual consequences are written as facts in knowledge base.

Thus, this methodology delimits very neatly the normative and factual contents of legal acts. Deontic elements all lie in PN places, while factual elements are all stored in knowledge base and are invoked via ECA rules from PN transitions.

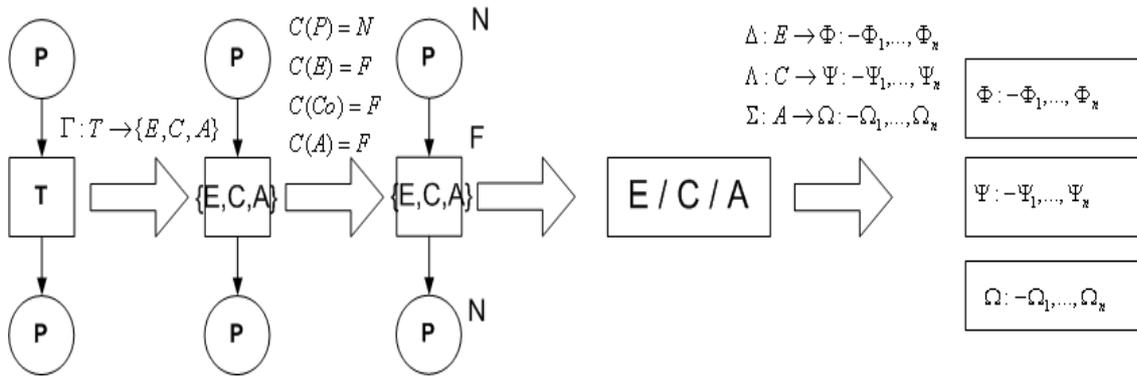


Figure 1: Mappings

6.1 Step Semantics

Step semantics can be now defined anew. Transitions are mapped to ECA rules and these are in turn mapped to expert system queries. Step semantics has to take into account possible composite nature of events, conditions and actions.

- Check which transitions are enabled in what modes. For each transition and for each mode do:

 1. resolve mode and transition priority
 2. check whether transition contains simple event, complex event or null event
 - (a) if it contains simple event, check whether event has happened and if so go to the condition part
 - (b) if it contains complex event call ES inference machine
 - i. infer simple events from a complex one
 - ii. for each simple event: check it against the database or ask user
 - iii. evaluate truth function of a complex event; if it evaluates to true: go to the condition part
 - (c) if it contains null event: go to the condition part
 3. check whether condition is simple, complex or empty
 - (a) if the condition is simple and is satisfied: fire
 - (b) if the condition is complex then call ES inference machine
 - i. infer simple conditions from a complex one
 - ii. for each simple condition: check it against the database or ask user
 - iii. evaluate truth function of complex condition; if it evaluates to true: fire
 - (c) if the condition is empty: fire
 4. withdraw tokens from appropriate input places
 5. put tokens into appropriate output places
 6. set off appropriate action from action part of ECA rule

7 Examples

A real world example can now be presented that is based on this model. We will take first few chapters of Federal Rules of Civil Procedure and try to express them in our model. We take this example because it requires modelling of both procedural and substantive law. Also, these rules have already been modelled in (Meldman J.A., Holt A.W. (1971) and Meldman J.A. (1978)).

Procedure begins with plaintiff filing a complaint.

Rule 3 Commencement of action A civil action is commenced by filling a complaint with the court

Then, court issues summons.

Rule 4 Process Upon filing of the complaint the clerk shall forthwith issue a summons and deliver it for service to the marshal or to a person specially appointed to serve it.

Many events have been collected under one umbrella here: issue summons, deliver summons, and serve defendant with summons. System calls knowledge base and retrieves body of *summon* rule, which consists of three simple events: issue summons, deliver summons, serve the defendant. System matches these events with those written in the knowledge base and eventually asks user about them (is it the case that ...). Note that this is the event part of the ECA rule. Condition part is empty (null). We also have action part here, which sets latest respond time within which defendant must answer or else be confronted with default judgement.

As a consequence of this, defendant now has the right to answer with pleading, counterclaim, motion or default.

Rule 7 Pleadings allowed There shall be a complaint and an answer; [which may or may not contain a counterclaim, and a reply to a counterclaim.] No other pleadings shall be allowed...

Rule 12 Defenses: by pleading or motion A defendant shall serve his answer within 20 days after the service of the summons and complaint upon him...

Rule 55 Default When a party against whom a judgment for affirmative relief is sought has failed to plead or otherwise defend as provided by these rules...the clerk shall enter his default.

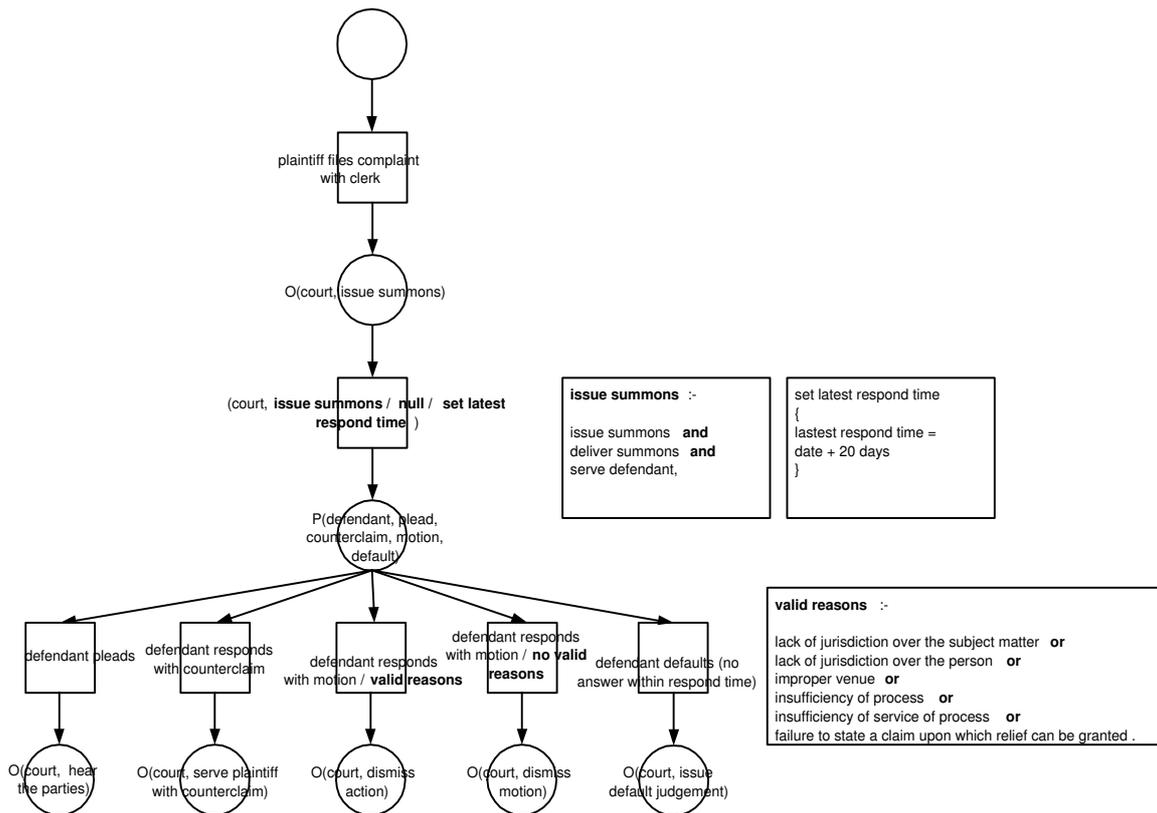


Figure 2: Civil Procedure

If he chooses to plead, court proceeds with action. If he responds with counterclaim, court then serves plaintiff with counterclaim. If he defaults (doesn't answer in latest respond time, which the system has set before), court issues default judgement against him.

If he answers with motion, everything depends on validity of reasons for motion.

Rule 12b ... Every defense, in law or fact, to a claim for relief in any pleading, whether a claim, or counterclaim, ... shall be asserted in the responsive pleading thereto... except that the following defenses may at the option of the pleader be made by motion: (1) lack of jurisdiction over the subject matter, (2) lack of jurisdiction over the person, (3) improper venue, (4) insufficiency of service of process, (6) failure to state a claim upon which relief can be granted...

These are conditions. System searches the knowledge base for *valid reasons for motion* and retrieves: lack of jurisdiction over subject matter, lack of jurisdiction over persons, improper venue, insufficiency of process, insufficiency of service of process, failure to state a claim upon which relief can be granted. Any of these reasons (conditions) make court dismiss action. If none of them is satisfied, court dismisses motion.

Here, the procedure goes on, of course. Model could be extended further, but we stop here because it serves demonstration purposes.

8 Conclusions

This paper has shown how high-level Petri nets, expert systems and ECA rules can be combined to represent semantics of legal rules. Different aspects of legal rules can be covered: factual, deontic, procedural and substantive. All of these are put into one picture.

Purpose of this paper is to show how semantics of legal rules can be mapped to different technologies and notations and how they can work together. This methodology has been applied to a few examples. Federal Rules of Civil Procedure, presented in this paper, is one of them. Both factual and deontic, procedural and substantive rules were covered in it.

Model could be extended further; one obvious way is to include linguistic elements as a micro structure (macro structure being Petri net). For example, if one wants to operate with finer elements than rights and duties (components like norm subjects, norm objects, prescribed behaviors) they must be interpreted individually. One way of doing this is by parsing text of legal rule and obtaining individual words or atoms like legal subjects, legal objects etc.). Then, not only facts, but single words also, could be subjects of queries and rules of expert systems (e.g. what is the meaning of word *due* in legal expression *due diligence*?).

Also, model could be made executable. A mapping

could be defined from the model to some programming language data types or DB schema. Models would then be migrated to this platform. Of course, a lot of implementation issues would have to be solved. Linguistics, again, could be of great help in determining atoms of such model.

Time aspects have only been very briefly touched in this paper. Since time is ubiquitous in information systems, model should be augmented with it. One way to do this is with Time or Timed Petri nets, where time is attributed to places or transitions or both.

Petri nets themselves could be replaced by more flexible structures. Petri nets semantics require strict alternation of transitions and places. Some real-world legal situations may escape this logic and we may very well find ourselves in need of a more flexible semantics.

References

- [18] Federal Rules of Civil Procedure (2001) www.house.gov/judiciary/civil00.pdf
- [18] Allen L.E. (1997) The Language of LEGAL RELATIONS (LLR): Useful in a Legal Ontologist's Toolkit?. *Proceedings of the First International Workshop on Legal Ontologies LEGONT'97*, Melbourne, Victoria, Australia, p. 47-60.
- [18] Bratko I. (2001), *Prolog Programming for Artificial Intelligence - 3rd edition*, Addison-Wesley.
- [18] Breuker J., Valente A., Winkels R. (1997) Legal Ontologies: A Functional View. *Proceedings of the First International Workshop on Legal Ontologies LEGONT'97*, Melbourne, Victoria, Australia, p. 23-36.
- [18] Burg J.F.M., Van de Riet R.P. (1994) Syntax, Semantics and Pragmatics of COLOR-X Event Models Specifying the Dynamics of Information and Communication Systems. *Technical Report IR-365, Vrije Universiteit, Amsterdam, 1994*
- [18] Burg J.F.M., Van de Riet R.P. (1994) COLOR-X: Object modelling profits from linguistics. *Technical Report IR-365, Vrije Universiteit, Amsterdam*
- [18] Chakravarthy S., Mishra D. (1991) Snoop: An Expressive Event Specification Language For Active Databases. *Tech. Report UF-CIS-TR-93-007*, Gainesville, Florida.
- [18] Dittrich K.R., Gatzui S. (1993) Time Issues in Active Database Systems. *Proceedings of International Workshop on an Infrastructure for Temporal Databases*, Arlington, Texas, p. 1-6.
- [18] Fedorov S. (1991) ICaR project, - applying relational data model and expert systems technology to represent and use legal knowledge. *Industrial Report ICAIL-99 Conference, 1991*
- [18] Gatzui S. (1993) Events in an Active Object-Oriented Database System. *Proceedings of the 1.st International Workshop on Rules in Database Systems*, Edinburgh.
- [18] Jonathan Billington (Editor) (1997) High-level Petri Nets - Concepts, Definitions and Graphical Notation. *Committee Draft ISO/IEC 15909*, October 2, 1997 Version 3.4
- [18] Kralingen van R. (1997) A Conceptual Frame-based Ontology for the Law. *Proceedings of the First International Workshop on Legal Ontologies LEGONT'97*, Melbourne, Victoria, Australia, p. 15-22.
- [18] Lee R. M., Ryu Y.U. (1994) DX: A Deontic Expert System. *Journal of Management Information Systems, Vol. 12, No. 1, 1995*, pp. 145-169.
- [18] Lee R.M., Bons R.W.H., Wrigley C.D., Wagenaar R. W. (1995) Modelling Inter-organizational Trade Procedures Using Documentary Petri Nets. *Proceedings of the Hawaii International Conference on System Sciences 1995*.
- [18] Meldman J.A., Holt A.W. (1971) Petri Nets and Legal System. *Jurimetrics Journal 12/2 1971*, ,p. 65-75.
- [18] Meldman J.A. (1978) A Petri-Net Representation of Civil Procedure. *IDEA The Journal of Law and Technology 19/2 1978* , , p. 123-148.
- [18] Sergot M.J., Sadri F., Kowalski R.A., Kriwaczek F., Hammond P., Cory H.T. (1986) The British Nationality Act as a logic program. *Communications of the ACM 370, 1986*
- [18] Visser P.R.S., Bench-Capon J.M. (1997) A Comparison of Two Legal Ontologies. *Proceedings of the First International Workshop on Legal Ontologies LEGONT'97*, Melbourne, Victoria, Australia, p. 37-46.

INFORMATION SOCIETY 2003
Ljubljana, Slovenia
13.-17. October 2003

Members of programme committee

Cene Bavec, chair
 Tomaž Kalin, co-chair
 Jozsef Györkös, co-chair
 Marko Bohanec
 Jaroslav Berce, Ivan Bratko
 Maja Bučar, Dušan Caf
 Saša Divjak, Tomaž Erjavec
 Bogdan Filipič, Matjaž Gams
 Marko Grobelnik, Nikola Guid
 Marjan Heričko, Igor Jerman
 Borka Jerman Blažič Džonova
 Gorazd Kandus,
 Igor Kononenko
 Marjan Krisper, Andrej Kuščer
 Jadran Lenarčič,
 Dunja Mladenič, Franc Novak
 Vladislav Rajkovič
 Ivan Rozman
 Niko Schlamberger
 Franc Solina, Stanko Strmčnik
 Tomaž Šef, Jurij Tasič
 Denis Trček, Andrej Ule
 Tanja Urbančič
 David B. Vodušek
 Baldomir Zajc, Blaž Zupan
 Boštjan Vilfan

Members of international programme committee

Vladimir Bajic
 Heiner Benking
 Wray Buntine
 Se Woo Cheon
 Howie Firth
 Vladimir Fomichov
 Alfred Inselberg
 Jay Liebowitz
 Huan Liu
 Henz Martin
 Marcin Paprzycki
 Karl Pribram
 Claude Sammut
 Jiri Wiedermann
 Xindong Wu
 Yiming Ye
 Ning Zhong

Organizational committee

Matjaž Gams, chair
 Damjan Demšar
 Aleksander Pivk
 Mili Remetič
 Mitja Luštrek

You are kindly invited to cooperate on multi-conference Information Society – IS 2003, which will be held from 13th to 17th of October 2003 in Ljubljana. The multi-conference will include important achievements on the fields mentioned below. Emphasis will be given on the exchange of ideas and particular suggestions, which will be included in the final paper of individual conferences.

IS 2003 exists of nine carefully chosen conferences:

Collaboration and information society

- Complex systems in e-business CSeB'03
- Development and reengineering of information systems
- Education in information society
- Intelligent and computer systems
- Management and information society
- Collaboration and information society
- Cognitive science
- Theoretical computer science

Further information is available at <http://ai.ijs.si/is/is2003/> or <http://ai.ijs.si/is/is2003/conf03eng.html>.

Institutions, enterprises and donators are invited to present interesting new developments on their fields of work as 'normal' contributions. They can make a review of new developments and existing situation in their institutions and talk about problems of development in Slovenia, attitude of governmental institutions, and about the way Slovenia should be developing in the direction of information society. They can grant certain interesting activities, related to their work (please turn to the organizer, for example matjaz.gams@ijs.si).

The emphasis is on development, new ideas and trends in information society. If you have something interesting to tell or show to Slovenia, Information Society is the right place to be.

Invited are primarily all those, who have some knowledge about information society. Presentations of enterprises are welcome, especially from the functional point of view. To summarize, we will meet to tell what can we do in Slovenia, to exchange our experiences and to help Slovenia make a step forward in the direction of information society.

You are kindly invited to make a presentation and actively take part in the open exchange of ideas with your knowledge and achievements. The submission deadline is fall 2003.

Pictures from the IS 2002 conference can be found at <http://is.ijs.si/is/is2002/slike/indexSlike.html> and videos at <http://solomon.ijs.si/>.

ERK'2003
Electrotechnical and Computer Science Conference
Elektrotehniška in računalniška konferenca
September 25–26, 2003

Conference Chairman

Baldomir Zajc

University of Ljubljana
Faculty of Electrical Engineering
Tržaška 25, 1001 Ljubljana, Slovenia
Tel: 386 1 4768 349, Fax: 386 1 4264 630
E-mail: Baldomir.Zajc@fe.uni-lj.si

Conference Vice-chairman

Saša Divjak

University of Ljubljana
Faculty of Comput. and Inform. Science
Tržaška 25, 1001 Ljubljana, Slovenia
Tel: (061) 1768 260, Fax: 386 1 4264 647
E-mail: Sasa.Divjak@fri.uni-lj.si

Program Committee Chairman

Jurij Tasič

University of Ljubljana
Faculty of Electrical Engineering
Tržaška 25, 1001 Ljubljana, Slovenia
Tel: 386 1 4768 260, Fax: 386 1 4264 630
E-mail: Jure.Tasic@fe.uni-lj.si

Programme Committee

Tadej Bajd

Genevieve Baudoin

Saša Divjak
Janko Drnovšek
David J. Evans
Matjaž Gams
Ferdo Gubina
Marko Jagodič
Karel Jezernik
Drago Matko
Miro Milanovič
Nikola Pavešič
Franjo Pernuš
Kurt Richter
Borut Zupančič
Andrej Žemva

Publications Chairman

Franc Solina

University of Ljubljana
Faculty of Comput. and Inform. Science
Tržaška 25, 1001 Ljubljana, Slovenia
Tel: 386 1 4768 389, Fax: 386 1 4264 647
E-mail: franc@fri.uni-lj.si

Advisory Board

Rudi Bric

Damjan Dittrich
Miloš Urbanija

Call for Papers

for the twelfth **Electrotechnical and Computer Science Conference ERK'2003**, which will be held on 25–26 September 2003 in Portorož, Slovenia.

The following areas will be represented at the conference:

- *electronics,*
- *telecommunications,*
- *automatic control,*
- *simulation and modeling,*
- *robotics,*
- *computer and information science,*
- *artificial intelligence,*
- *pattern recognition,*
- *biomedical engineering,*
- *power engineering,*
- *measurements,*
- *didactics.*

The conference is organized by the **IEEE Slovenia Section** together with the Slovenian Electrotechnical Society and other Slovenian professional societies:

- Slovenian Society for Automatic Control,
- Slovenian Measurement Society,
- SLOKO–CIGRE,
- Slovenian Society for Medical and Biological Engineering,
- Slovenian Society for Robotics,
- Slovenian Artificial Intelligence Society,
- Slovenian Pattern Recognition Society,
- Slovenian Society for Simulation and Modeling,
- Slovenian Language Technologies Society.

Authors who wish to present a paper at the conference should send two copies of their final camera-ready paper to Baldomir Zajc, Faculty of Electrical Engineering, Tržaška 25, 1001 Ljubljana. The paper should be max. four pages long. More information on <http://www.ieee.si/erk00/>

Time schedule: Camera-ready paper due: *July 11, 2003*
Notification of acceptance: *End of August, 2003*

Faculty of Management Koper

Annual Conference

Call for papers

4th international conference

»Knowledge society – Challenges to Management: Globalisation, Regionalism and the EU Enlargement Process«

November 20-22, 2003, GH Emona, Portorož, Slovenia

The enlargement of the EU comprises 13 countries, 10 of which will be joining the EU in 2004. The whole process of accession raises a number of issues and questions about the future of education, research and employment policy in the New Europe. The scope of this conference is to address these issues and to challenge politicians, researchers, academics and practitioners to discuss questions related to the concept of knowledge society, i.e. how it is built and what are its implications for national education systems in relation to the Common European area.

We look forward to meeting academics, researchers and practitioners from educational and research organisations as well as management of companies from EU and acceding countries directly involved in the implementation of the *acquis communautaire* and in the field of education and research.

Aims of the conference:

- to discuss current legal and other issues related to the legal framework for education and research within the European Union
- to discuss creation and use of knowledge in Europe and its interconnectedness with the labour market after the admission of new EU member states
- to present case studies and analyses from different national contexts that raise questions about management of organisations and their focus on investments into knowledge
- to provoke and indicate changes of the integration process in education and research areas

Topics of the conference:

- Staff recruiting and development
 - international comparisons of legal and legislative issues
 - national examples of “good practice”
 - national approaches to staff development in organisations
 - transition and post-transition period as reflected in recruitment and staff development
- Education and research
 - legal issues
 - approaches to a common education and research area
 - management of education and research institutions

- integration processes in education and research for countries in transition
- Creation and use of knowledge
 - use of management knowledge
 - staff and creatibility
 - organisation structures that enable knowledge transfer, creation and use creation of knowledge
 - creation of knowledge and culture and its use in education and research approaches and dilemmas
 - action research in management as a means of creating and using knowledge
- Countries in transition
 - emerging issues in labour market
 - political changes in relation to policy making process

We kindly invite participants to submit abstracts for papers, round-table discussions, poster sessions by July 1, 2003 electronically to e-mail address: conference@fm-kp.si. For submission forms please visit our web site: <http://www.fm-kp.si>

The Programme Board will inform authors about acceptances by July 15, 2003 the latest. The conference proceedings will be published after the conference.

Selected papers will be considered for the international journal *Managing Global Transitions*, which is published by The Faculty of Management Koper. Articles in magazine are peer reviewed.

Final papers are expected by October 1, 2003 by e-mail address: conference@fm-kp.si

Programme Board

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan-Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 700 staff, has 500 researchers, about 250 of whom are postgraduates, over 200 of whom have doctorates (Ph.D.), and around 150 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of Slovenia (or S^{lo}venia). The capital today is considered a crossroad between East, West and Mediter-

anean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

In the last year on the site of the Jožef Stefan Institute, the Technology park "Ljubljana" has been proposed as part of the national strategy for technological development to foster synergies between research and industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

At the present time, part of the Institute is being reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park "Ljubljana". The project is being developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park will take the form of a shareholding company and will host an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of Economic Relations and Development, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.:+386 1 4773 900, Fax.:+386 1 219 385
Tlx.:31 296 JOSTIN SI
WWW: <http://www.ijs.si>
E-mail: matjaz.gams@ijs.si
Contact person for the Park: Iztok Lesjak, M.Sc.
Public relations: Natalija Polenec

INFORMATICA
AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS
INVITATION, COOPERATION

Submissions and Refereeing

Please submit three copies of the manuscript with good copies of the figures and photographs to one of the editors from the Editorial Board or to the Contact Person. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible directly on the manuscript, from typing errors to global philosophical disagreements. The chosen editor will send the author copies with remarks. If the paper is accepted, the editor will also send copies to the Contact Person. The Executive Board will inform the author that the paper has been accepted, in which case it will be published within one year of receipt of e-mails with the text in Informatica L^AT_EX format and figures in .eps format. The original figures can also be sent on separate sheets. Style and examples of papers can be obtained by e-mail from the Contact Person or from FTP or WWW (see the last page of Informatica).

Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the Contact Person.

QUESTIONNAIRE

Send Informatica free of charge

Yes, we subscribe

Please, complete the order form and send it to Dr. Drago Torkar, Informatica, Institut Jožef Stefan, Jamova 39, 1111 Ljubljana, Slovenia.

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than ten years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering the European computer science and informatics community - scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica is free of charge for major scientific, educational and governmental institutions. Others should subscribe (see the last page of Informatica).

ORDER FORM – INFORMATICA

Name:	Office Address and Telephone (optional):
Title and Profession (optional):
.....	E-mail Address (optional):
Home Address and Telephone (optional):
.....	Signature and Date:

Informatica WWW:

<http://ai.ijs.si/informatica/>
<http://orca.st.usm.edu/informatica/>

Referees:

Witold Abramowicz, David Abramson, Adel Adi, Kenneth Aizawa, Suad Alagić, Mohamad Alam, Dia Ali, Alan Aliu, Richard Amoroso, John Anderson, Hans-Jurgen Appelrath, Iván Araujo, Vladimir Bajič, Michel Barbeau, Grzegorz Bartoszewicz, Catriel Beerli, Daniel Beech, Fevzi Belli, Simon Beloglavec, Sondes Bennisri, Francesco Bergadano, Istvan Berkeley, Azer Bestavros, Andraž Bežek, Balaji Bharadwaj, Ralf Birkenhead, Ralph Bisland, Jacek Blazewicz, Laszlo Boeszöermeenyi, Damjan Bojadžijev, Jeff Bone, Marco Botta, Ivan Bratko, Pavel Brazdil, Bostjan Brumen, Jerzy Brzezinski, Marian Bubak, Davide Bugali, Troy Bull, Leslie Burkholder, Frada Burstein, Wojciech Buszkowski, Rajkumar Bvyya, Netiva Caftori, Patricia Carando, Robert Cattral, Jenny Carter, Jason Ceddia, Ryszard Choras, Wojciech Cellary, Se Woo Cheon, Wojciech Chybowski, Andrzej Ciepielewski, Vic Ciesielski, Mel Ó Cinnéide, David Cliff, Maria Cobb, Jean-Pierre Corriveau, John Cowell, Travis Craig, Noel Craske, Matthew Crocker, Tadeusz Czachorski, Milan Češka, Honghua Dai, Bart de Decker, Deborah Dent, Andrej Dobnikar, Sait Dogru, Peter Dolog, Georg Dorfner, Ludoslaw Drelichowski, Matija Drobnič, Maciej Drozdowski, Marek Druzdzel, Marjan Družovec, Jozo Dujmović, Pavol Ďuriš, Amnon Eden, Johann Eder, Hesham El-Rewini, Darrell Ferguson, Warren Fergusson, David Flater, Pierre Flener, Wojciech Fliegner, Vladimir A. Fomichov, Terrence Forgarty, Hans Fraaije, Hugo de Garis, Eugeniusz Gatnar, Grant Gayed, James Geller, Michael Georgiopolus, Michael Gertz, Jan Goliński, Janusz Gorski, Georg Gottlob, David Green, Herbert Groiss, Jozsef Gyorkos, Marten Haglind, Abdelwahab Hamou-Lhadj, Inman Harvey, Jaak Henno, Marjan Hericko, Elke Hochmueller, Jack Hodges, Doug Howe, Rod Howell, Tomáš Hruška, Don Huch, Simone Fischer-Huebner, Peter Innocent, Alexey Ippa, Hannu Jaakkola, Radha Jagadeesan, Sushil Jajodia, Ryszard Jakubowski, Piotr Jedrzejowicz, A. Milton Jenkins, Eric Johnson, Polina Jordanova, Djani Juričić, Marko Juvancic, Sabhash Kak, Li-Shan Kang, Ivan Kapustok, Orlando Karam, Roland Kaschek, Jacek Kierzenka, Jan Kniat, Stavros Kokkotos, Fabio Kon, Kevin Korb, Gilad Koren, Andrej Krajnc, Henryk Krawczyk, Ben Kroese, Zbyszko Krolikowski, Benjamin Kuipers, Matjaž Kukar, Franz Kurfess, Aarre Laakso, Les Labuschagne, Ivan Lah, Phil Laplante, Bud Lawson, Herbert Leitold, Ulrike Leopold-Wildburger, Timothy C. Lethbridge, Joseph Y-T. Leung, Barry Levine, Xuefeng Li, Alexander Linkevich, Raymond Lister, Doug Locke, Peter Lockeman, Matija Lokar, Jason Lowder, Kim Teng Lua, Ann Macintosh, Bernardo Magnini, Andrzej Małachowski, Peter Marcer, Andrzej Marciniak, Witold Marciszewski, Vladimir Marik, Jacek Martinek, Tomasz Maruszewski, Florian Matthes, Daniel Memmi, Timothy Menzies, Dieter Merkl, Zbigniew Michalewicz, Gautam Mitra, Roland Mittermeir, Madhav Moganti, Reinhard Moller, Tadeusz Morzy, Daniel Mossé, John Mueller, Jari Multisilta, Hari Narayanan, Jerzy Nawrocki, Rance Necaie, Elzbieta Niedzielska, Marian Niedźwiedziński, Jaroslav Nieplocha, Oscar Nierstrasz, Roumen Nikolov, Mark Nissen, Jerzy Nogieć, Stefano Nolfi, Franc Novak, Antoni Nowakowski, Adam Nowicki, Tadeusz Nowicki, Daniel Olejar, Hubert Österle, Wojciech Olejniczak, Jerzy Olszewski, Cherry Owen, Mieczyslaw Owoc, Tadeusz Pankowski, Jens Penberg, William C. Perkins, Warren Persons, Mitja Peruš, Stephen Pike, Niki Pissinou, Aleksander Pivk, Ullin Place, Gabika Polčicová, Gustav Pomberger, James Pomykalski, Dimithu Prasanna, Gary Preckshot, Dejan Rakovič, Cveta Razdevšek Pučko, Ke Qiu, Michael Quinn, Gerald Quirchmayer, Vojislav D. Radonjic, Luc de Raedt, Ewaryst Rafajłowicz, N. Raja, Sita Ramakrishnan, Kai Rannenber, Wolf Rauch, Peter Rechenberg, Felix Redmill, James Edward Ries, David Robertson, Marko Robnik, Colette Rolland, Wilhelm Rossak, Ingrid Russel, A.S.M. Sajeev, Kimmo Salmenjoki, Pierangela Samarati, Bo Sanden, P. G. Sarang, Vivek Sarin, Iztok Savnik, Ichiro Satoh, Walter Schempp, Wolfgang Schreiner, Guenter Schmidt, Heinz Schmidt, Dennis Sewer, Zhongzhi Shi, Mária Smolárová, Carine Souveyet, William Spears, V. Sriram, Hartmut Stadler, Olivero Stock, Janusz Stokłosa, Przemysław Stpicyński, Andrzej Stritar, Maciej Stroinski, Leon Strous, Tomasz Szmuc, Zdzisław Szyjewski, Jure Šilc, Metod Škarja, Jiří Šlechta, Chew Lim Tan, Zahir Tari, Jurij Tasič, George Tecuci, Piotr Teczynski, Stephanie Teufel, Ken Tindell, A Min Tjoa, Vladimir Tosic, Wiesław Traczyk, Roman Trobec, Marek Tudruj, Andrej Ule, Amjad Umar, Andrzej Urbanski, Marko Uršič, Tadeusz Usowicz, Romana Vajde Horvat, Elisabeth Valentine, Kanonkluk Vanapipat, Rangarajan A. Vasudevan, Alexander P. Vazhenin, Jan Verschuren, Zygmunt Vetulani, Olivier de Vel, Valentino Vranić, Jozef Vyskoc, Eugene Wallingford, Matthew Warren, John Weckert, Michael Weiss, Tatjana Welzer, Lee White, Gerhard Widmer, Stefan Wrobel, Stanislaw Wrycza, Janusz Zalewski, Damir Zazula, Yanchun Zhang, Ales Zivkovic, Zonling Zhou, Robert Zorc, Anton P. Železnikar

Informatica

An International Journal of Computing and Informatics

Archive of abstracts may be accessed at USA: <http://>, Europe: <http://ai.ijs.si/informatica>, Asia: <http://www.comp.nus.edu.sg/liuh/Informatica/index.html>.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Vožarski pot 12, 1000 Ljubljana, Slovenia.

The subscription rate for 2003 (Volume 27) is

- USD 80 for institutions,
- USD 40 for individuals, and
- USD 20 for students

Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

L^AT_EX Tech. Support: Borut Žnidar, Kranj, Slovenia.

Lectorship: Fergus F. Smith, AMIDAS d.o.o., Cankarjevo nabrežje 11, Ljubljana, Slovenia.

Printed by Biro M, d.o.o., Žibertova 1, 1000 Ljubljana, Slovenia.

Orders for subscription may be placed by telephone or fax using any major credit card. Please call Mr. R. Murn, Jožef Stefan Institute: Tel (+386) 1 4773 900, Fax (+386) 1 219 385, or send checks or VISA card number or use the bank account number 900–27620–5159/4 Nova Ljubljanska Banka d.d. Slovenia (LB 50101-678-51841 for domestic subscribers only).

Informatica is published in cooperation with the following societies (and contact persons):

Robotics Society of Slovenia (Jadran Lenarčič)

Slovene Society for Pattern Recognition (Franjo Pernuš)

Slovenian Artificial Intelligence Society; Cognitive Science Society (Matjaž Gams)

Slovenian Society of Mathematicians, Physicists and Astronomers (Bojan Mohar)

Automatic Control Society of Slovenia (Borut Zupančič)

Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Igor Grabec)

ACM Slovenia (Dunja Mladenič)

Informatica is surveyed by: AI and Robotic Abstracts, AI References, ACM Computing Surveys, ACM Digital Library, Applied Science & Techn. Index, COMPENDEX*PLUS, Computer ASAP, Computer Literature Index, Cur. Cont. & Comp. & Math. Sear., Current Mathematical Publications, Cybernetica Newsletter, DBLP Computer Science Bibliography, Engineering Index, INSPEC, Linguistics and Language Behaviour Abstracts, Mathematical Reviews, MathSci, Sociological Abstracts, Uncover, Zentralblatt für Mathematik
--

The issuing of the Informatica journal is financially supported by the Ministry of Education, Science and Sport, Trg OF 13, 1000 Ljubljana, Slovenia.

Informatica

An International Journal of Computing and Informatics

Introduction		117
Perception and Emotion Based Reasoning: A Connectionist Approach	A. Ayesh	119
Emotional Influences on Perception in Artificial Agents	P. Baillie-de Byl	127
Enhancing the Performance of Neurofuzzy Predictors by Emotional Learning Algorithm	C. Lucas, A. Abbaspour, A. Gholipour, B.N. Araabi, M. Fatourehchi	137
Emotion-Based Decision and Learning Using Associative Memory and Statistical Estimation	B.D. Damas, L.M. Custódio	147
Computational Models of Emotion for Autonomy and Reasoning	D.N. Davis, S.C. Lewis	159
Emotional Learning as a New Tool for Development of Agent-based Systems	M. Fatourehchi, C. Lucas, A.K. Sedig	167
Learning Behavior-selection in a Multi-Goal Robot Task	S.C. Gadanho, L. Custódio	175
Multiple Emotion-Based Agents using an Extension of DARE Architecture	M. Maças, L. Custódio	185
Timidity: A Useful Emotional Mechanism for Robot Control?	M. Neal, J. Timmis	197
Emotional Information Retrieval for a Dialogue Agent	R. Rzepka, K. Araki, K. Tochinal	205
<hr/>		
Data Compression in a Pharmaceutical Drug Candidate Database	Z.B. Miled, H. Li, O. Bukhres, M. Bem, R. Jones, R. Oppelt	213
Modelling Legal Acts by Means of Expert Systems, ECA Rules and High-level Petri Nets	B. Berčič	225