

Volume 22 Number 3 October 1998

ISSN 0350-5596

Informatica

**An International Journal of Computing
and Informatics**

Special Issue:

**Parallel Computing With
Optical Interconnections**

The Slovene Society Informatika, Ljubljana, Slovenia

Informatica 22 (1998) Number 3, pp. 253-403



Informatica

An International Journal of Computing and Informatics

Basic info about Informatica and back issues may be FTP'ed from `ftp.arnes.si` in `magazines/informatica` ID: `anonymous` PASSWORD: `<your mail address>`
FTP archive may be also accessed with WWW (worldwide web) clients with
URL: `http://turing.ijs.si/Mezi/informatica.htm`

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 1998 (Volume 22) is

- DEM 50 (US\$ 35) for institutions,
- DEM 25 (US\$ 17) for individuals, and
- DEM 10 (US\$ 7) for students

plus the mail charge DEM 10 (US\$ 7).

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) 61 1773 900, Fax (+386) 61 219 385, or use the bank account number 900-27620-5159/4 Ljubljanska banka d.d. Slovenia (LB 50101-678-51841 for domestic subscribers only).

According to the opinion of the Ministry for Informing (number 23/216-92 of March 27, 1992), the scientific journal Informatica is a product of informative matter (point 13 of the tariff number 3), for which the tax of traffic amounts to 5%.

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 (Janez Peklenik)

Informatica is surveyed by: AI and Robotic Abstracts, AI References, ACM Computing Surveys, Applied Science & Techn. Index, COMPENDEX*PLUS, Computer ASAP, Computer Literature Index, Cur. Cont. & Comp. & Math. Sear., Current Mathematical Publications, Engineering Index, INSPEC, Mathematical Reviews, MathSci, Sociological Abstracts, Uncover, Zentralblatt für Mathematik, Linguistics and Language Behaviour Abstracts, Cybernetica Newsletter

The issuing of the Informatica journal is financially supported by the Ministry for Science and Technology, Slovenska 50, 1000 Ljubljana, Slovenia.

Post tax payed at post 1102 Ljubljana. Slovenia tax Percue.

Introduction to Parallel Computing with Optical Interconnections

Advances in semiconductor technologies coupled with progress in parallel processing and network computing are placing stringent requirements on inter-system and intra-system communications. With advances in silicon and Ga-As technologies, processor speed will soon reach the gigahertz (GHz) range. Thus, the communication technology is becoming and will remain a potential bottleneck in many systems. This dictates that significant progress needs to be made in the traditional metal-based interconnects, and/or that new interconnect technologies, such as optics, be introduced in these systems.

Optical means are now widely used in telecommunication networks and the evolution of optical and optoelectronic technologies tends to show that they could be successfully introduced in shorter distance interconnection systems such as parallel computers. These technologies offer a wide range of techniques that can be used in interconnection systems. But introducing optics in interconnect systems also means that specific problems have yet to be solved while some unique features of the technology must be taken into account in order to design optimal systems. Such problems and features include device characteristics, network topologies, packaging issues, compatibility with silicon processors, system level modeling, algorithm design, etc.

Papers in this special issue were selected to address the potential for using optical interconnections in massively parallel processing systems, and their effect on system and algorithm design. Optics offer many benefits for interconnecting large numbers of processing elements, but may require us to rethink how we build parallel computer systems and communication networks, and how we write algorithms. Fully exploring the capabilities of optical interconnection networks requires an interdisciplinary effort. We hope this special issue serves the reader in this respect.

Through rigorous reviews, six papers were chosen from a pool of papers submitted to this special issue. This is reflected in the high quality of the papers accepted. In this issue, the papers by Middendorf and ElGindy, and Sahni and Wang are published. In the next issue we expect to publish the remaining four papers. We wish the reader enjoy reading the papers in this special issue and the following one and find useful information there.

Middendorf and ElGindy present an algorithm for matrix multiplication on an array of processors with optical pipelined row and column buses (APPB). They show that two $n \times n$ matrices A and B with elements that can be represented by $O(w)$ bits and where the number of nonzero elements of B is at most $k_B \cdot n$, $1 \leq k_B \leq n$ can be multiplied on an $n \times n \times q$ APPB in time $O(\frac{k_B}{q} + \max\{\log w, \log \log n\})$.

Sahni and Wang study the the optical transpose interconnection system (OPIS), in particular the OTIS-Hypercube architecture. They obtain basic properties and basic permutation routing algorithms for this architecture.

Guizani and Memon propose a fault-tolerant high-performance switch for optical interconnection networks. The switch is designed using electro-optical components which allow the exploitation of spatial parallelism of optics.

Hypernetworks are characterized by hypergraphs, and are especially suitable for optical interconnects. In their paper, Zheng *et al* consider using the dual Q_n^* of hypercube of Q_n as an interconnection network. They investigate the properties of Q_n^* , and present a set of fundamental data communication algorithms for Q_n^* . Their results indicate that hypernetwork Q_n^* is a useful and promising interconnection structure for high-performance parallel and distributed computing systems.

Awwal *et al* presented a high speed parallel adder which can perform carry-free addition of two modified signed digit quaternary numbers. Binary coded quaternary logic design is provided as one possible way of implementing the digital and/or optical system. Experimental results are also included for the proposed adder circuit.

In his paper, Wan proposes the star graph as the cluster interconnection topology as it possesses many attractive properties. He gives an optimal conflict-free channel set assignment for this new interconnection topology.

Acknowledgment

We thank all those who submitted papers to this special issue. We are also thankful to Dr. Matjaz Gams, the editor of Informatica, for his guidance and support in integrating this issue. Many thanks also go to the reviewers for their prompt responses and helpful comments.

References

- [1] R. Melhem and D. Chiarulli, eds., Special Issue on Optical Computing and Interconnection Systems of *Journal of Parallel and Distributed Computing*, vol. 17, no. 3, 1993.
- [2] M. A. Karim and M. A. G. Abushagur, eds., Special Issue on Optical Computing, *Optics and Laser technology*, vol. 26, no. 4, 1994.

- [3] M. M. Eshaghian and E. Schenfeld, ed., Special Issue on Parallel Computing with Optical Interconnects," *Journal of Parallel and Distributed Computing*, vol. 41, no. 1, 1997.
- [4] K. Li, Yi Pan, and S.Q. Zheng, *Parallel Computing Using Optical Interconnections*, Kluwer Academic Publishers, Boston, USA, September 1998.

Guest Editors

- Professor Yi Pan, Computer Science Department, University of Dayton, Dayton, OH 45469-2160, USA, Email: pan@cps.udayton.edu.
- Professor Keqin Li, Dept. of Mathematics and Computer Science, State University of New York, New Paltz, New York 12561-2499, USA, Email: li@mcs.newpaltz.edu.
- Professor Mounir Hamdi, Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, Email: hamdi@cs.ust.hk.

by Yi Pan, Keqin Li and Mounir Hamdi

Matrix Multiplication on Processor Arrays with Optical Buses

Martin Middendorf

Institut für Angewandte Informatik und Formale Beschreibungsverfahren

Universität Karlsruhe, D-76128 Karlsruhe, Germany

E-mail: mmi@aifb.uni-karlsruhe.de

AND

Hossam ElGindy

Department of Electrical and Computer Engineering

University of Newcastle, Newcastle NSW 2308, Australia

E-mail: hossam@ee.newcastle.edu.au

Keywords: processor arrays, optical buses, matrix multiplication, sparse matrices

Edited by: Yi Pan

Received: October 16, 1997

Revised: January 20, 1998

Accepted: June 19, 1998

In this paper we present an algorithm for matrix multiplication on an array of processors with optical pipelined row and column buses (APPB). We show that two $n \times n$ matrices A and B with elements that can be represented by $O(w)$ bits and where the number of nonzero elements of B is at most $k_B \cdot n$, $1 \leq k_B \leq n$ can be multiplied on an $n \times n \times q$ APPB in time $O(\frac{k_B}{q} + \max\{\log w, \log \log n\})$. Our result improves some results obtained by Pavel and Akl on a reconfigurable array with optical buses (AROB) concerning the size of the array. Moreover, the APPB is a weaker model than the AROB.

1 Introduction

Optical connections are emerging as alternative to electrical ones due to their ability to transfer multiple signals simultaneously in short time over long distances. These features are due to the high propagation speed, the unidirectional nature and predictable delays of light signals. The high speed of light in optical buses makes it reasonable to assume that a signal can propagate over a long distance between two processors, attached to such a bus, in constant time (see e.g. [7]). This is also often assumed for electrical buses. But some authors doubt whether this is a practical assumption for large arrays of processors (see [1, 7]). The unidirectional nature of light signals allows for several signals to travel simultaneously in a pipelined fashion. In contrast, electrical signals require exclusive access to the electrical bus which may thus become a bottleneck for solving problems on multiprocessor architectures. These features make optical buses an attractive alternative for connecting very large arrays of processors to be used for solving communication intensive problems.

In this paper we study $n \times n$ matrix multiplication on meshes of processors connected by optical row and column buses (APPB's). Matrix multiplication is one of the basic procedures that is used by many algorithms. Therefore, several solutions to this problem have been proposed for different models of processor arrays with optical buses.

Qiao [9] gave an algorithm that multiplies two $n \times n$ matrices on a $n \times n$ array of processors with optical row and column buses that is enhanced with switches that allow for connecting row and column buses. Pavel and Akl [8] investigated matrix operations on reconfigurable arrays with optical buses (AROB) (see [7] for the AROB model). This model is the optical analogue to the classical reconfigurable arrays that use electrical buses (see [6] for the classical reconfigurable array model). In every bus-cycle switches allow each processor to connect or disconnect some of the up to four buslinks to which it is connected. Row and column buses are only one possible connection pattern that can be realised on this architecture. Pavel and Akl [8] showed that $n \times n$ matrix multiplication with elements represented by $\log n$ bits can be done in time $O(r)$ on an $n \times n \times \frac{n \log n}{r}$ or $n \times n \times n \times \frac{\log n}{r}$ AROB if $1 \leq r \leq \log n$. If $\log n \leq r \leq n$ they showed that the problem can be solved in time $O(r + \log(\frac{n}{r}))$ on an $n \times n \times \frac{n}{r}$ AROB. In this paper we improve on some of their results by showing that the problem can be solved in time $O(r)$ on an $n \times n \times \frac{n}{r}$ APPB if $\log \log n \leq r \leq n$. Our contribution is a new algorithm which uses a smaller size array to achieve the same running time as in [8] for $\log \log n \leq r < \log n$ and matches the running time and required array size for $\log n \leq r \leq n$. More importantly is that our algorithm assumes the APPB model which is a weaker model than the AROB used by Pavel et al. [8].

A special case of matrix multiplication is the matrix-

vector multiplication. The algorithm of Pavel and Akl can multiply an $n \times n$ matrix with an $n \times 1$ vector in time $O(1)$ on an $n \times n \times \log n$ AROB. We show that this problem can be solved in time $O(\log \log n)$ on an $n \times n$ APPB.

In many applications the emerging matrices are sparse, i.e. a lot of their elements are zero. Several definitions of sparseness can be found in the literature. Here we take the definition of [3] and call a matrix A k_A -sparse if it contains at most $k_A \cdot n$ nonzero elements. Pavel and Akl [8] studied matrix multiplication when both matrices are special forms of k -sparse matrices. Here we consider the case that only one of the matrices is sparse. We show that two $n \times n$ matrices A and B with elements represented by w bits and where B is k_B -sparse, $1 \leq k_B \leq n$ can be multiplied in time $O(q + \max\{\log w, \log \log n\})$ on an $n \times n \times \frac{k_B}{q}$ APPB for $1 \leq q \leq k_B$.

In the next section 2 the APPB model is described. Section 3 contains some basic techniques that are used later for the matrix multiplication. The matrix multiplication algorithm is described in Section 4. A conclusion is given in Section 5.

2 The Model

A linear array of processors with a pipelined bus (linear APPB) consists of processors connected via two directional couplers to an optical bus as depicted in Figure 1. The upper (*transmitting*) segment of the bus is used to write data on it whereas the lower (*receiving*) segment is used to read data from it. Data propagates along the bus only in one direction from the transmitting segment to the receiving segment. Communication between the processors is done synchronously, i.e. the processors that want to transmit a message do this simultaneously. This is possible since the propagation delay of optical signals is predictable and it is assumed that each processor is separated from its neighbour by a signal time delay $d = bp$ where b is the size of a message and p is the bit duration time. Therefore, during a communication-cycle every processor can safely send one message without physically overlapping a message send by another processor. Each message may contain a number of size $\max\{w, \log n\}$ bits where w is the maximum number of bits used to represent the elements of the matrices considered. Several techniques have been proposed that allow each processor to address any subset of the processors when sending a message (see [7]). Most of these techniques can be used for the SIMD or MIMD model. In this paper we use an SIMD array. During a communication-cycle each processor can send at most one message and receive at most one message. Also, we assume that it is not allowed that several messages are sent to the same processor during a communication-cycle. Each communication cycle requires constant time (see [7] for a

foundation of such an assumption). Each processor has only a constant number of registers and each register can store a $\max\{w, \log n\}$ -bit number.

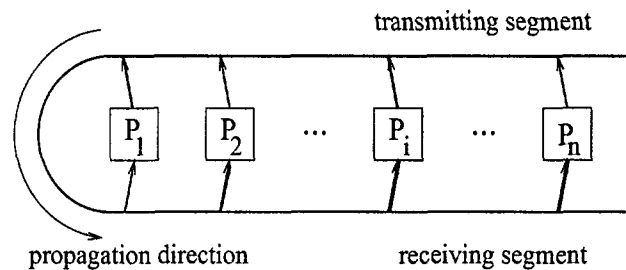


Figure 1: Linear array of processors with optical pipelined buses (linear APPB)

In this paper we consider two dimensional $n \times n$ arrays or three dimensional $n \times n \times r$ arrays of processors which are connected by optical row buses and column buses (and buses along the third dimension, respectively), i.e. each row and each column of the mesh is a linear APPB. This model is also called array of processors with pipelined optical buses (APPB) [2]. It should be noted that several extensions of this model have been proposed where switches allow to connect column buses with row buses (see e.g. [7, 9]). One interesting model is the reconfigurable array of processors with optical buses (AROB) introduced by Pavel and Akl [7] as an analogue to the reconfigurable array with electrical buses (see e.g. [4]).

3 Basic Techniques

In this section we describe some basic techniques that are used later for matrix multiplication. First we cite the following result of Pavel and Akl [8].

Lemma 1 ([8]) *The binary prefix sums $v_1 + v_2 + \dots + v_i$ of n one-bit values v_1, v_2, \dots, v_n where $v_i, i \in [1 : n]$ is stored in processor P_i of a linear APPB can be computed in constant time.*

We need the following operation: A processor P_h of a linear APPB that has stored a w -bit number z sends the j -th bit of z to a processor P_{i_j} for each $j \in [1 : w]$ where $i_s \neq i_t$ for $s \neq t, s, t \in [1 : w]$. Observe that this operation can be done in constant time as follows: P_h sends z to processors $P_{i_1}, P_{i_2}, \dots, P_{i_w}$ and processor P_{i_j} selects the j th bit of $z, j \in [1 : w]$.

Now, we describe an algorithm for the addition of n^2 w -bit numbers z_1, z_2, \dots, z_{n^2} on an $n \times n$ APPB in time $O(\log w + \log^* n)$. The idea of the algorithm, which follows a method of Nakano et al. [5] for summing integers on a reconfigurable mesh with electrical buses, is to use a method for summing w -bit numbers that is based on the summation of 1-bit values. To explain the method we first describe how n w -bit

numbers can be summed before we give the algorithm for summing n^2 w -bit numbers.

The sum S of n w -bit numbers z_1, z_2, \dots, z_n can be obtained as follows. First, the i th bit of z_j is sent to processor $P_{i,j}$, $i \in [1 : w]$, $j \in [1 : n]$. Then in each row $i \in [1 : w]$ the sum S_i of all i th bits for $i \in [1 : w]$ is computed. Clearly, each sum S_i has at most $\log n$ bits. It remains to compute $S = S_w \cdot 2^{w-1} + S_{w-1} \cdot 2^{w-2} + \dots + S_1 \cdot 2^0$. This is done by sending the j th bit, $j \in [1 : \log n]$, of S_i , $i \in [1 : w]$, to processor $P_{i+j-1,j}$ and taking a new sum S_i of the bits in each row. Now, each S_i has only at most $\log \log n$ bits. We repeat this at most $\log^* n$ times until we obtain sums S_i which have only one bit. Then, S_i is the i th bit of the sum $z_1 + z_2 + \dots + z_n$, $i \in [1 : w]$. It remains to form a binary value from the bits s_i , $i \in [1 : w]$. A process that can be performed in $O(\log w)$ steps.

Now, we give an outline of our algorithm ADD for computing the sum S of n^2 w -bit numbers on the $n \times n$ APPB where initially each processor contains one of the numbers. We assume $\log n \leq w \leq \frac{n}{2 \log n}$. Otherwise it is easy to see that we can add the numbers in time $O(\log w + \log^* n)$. First, the numbers in each $w \times 1$ submesh are added. Then, for the so obtained $\frac{n^2}{w}$ numbers the sum S_i of their i th bits is computed in parallel for $i \in [1 : w]$. Finally, S is obtained from the bits of the numbers S_i , $i \in [1 : w]$.

Algorithm ADD:

1. In each $w \times 1$ submesh add w numbers in time $O(\log w)$.
Time: $O(\log w)$.
2. For $i \in [1 : w]$ compute the sum S_i of the i th bits of the remaining $\frac{n^2}{w}$ numbers as follows (bits are numbered starting with the low order bit):
 - (a) For each $i \in [1 : w]$ send the i th bits of the remaining $\frac{n^2}{w}$ numbers along a column bus to the processors of the i th $\frac{n}{w} \times n$ submesh (each processor receives one bit). Then, add the bits in each row. Now, it remains to add for each $i \in [1 : w]$ the computed $\frac{n}{w} \log n$ -bit partial sums to obtain the sum S_i of the i th bits.
Time: $O(1)$.
 - (b) For each $i \in [1 : w]$, $j \in [1 : \log n]$ send the j th bits of the $\frac{n}{w}$ partial sums of S_i to the processors of row $(i-1) \log n + j$. Then, add these bits in each of the rows $(i-1) \log n + j$ and let $S_{i,j}$ be the obtained result. Clearly, each value $S_{i,j}$ has at most $\log n$ bits.
Time: $O(1)$.
 - (c) For each $i \in [1 : w]$, $j \in [1 : \log n]$ send the h th bit, $h \in [1 : \log n]$ of $S_{i,j}$ to a processor

of row $(i-1) \cdot 2 \log n + j + h$. Then, compute a new $S_{i,j}$ by adding these bits in each row $(i-1) \cdot 2 \log n + j$, $i \in [1 : w]$, $j \in [1 : 2 \log n]$. Clearly, each result $S_{i,j}$ has at most $\log \log n$ bits.

Repeat this step $O(\log^* n)$ times, until the bits $S_{i,1}, S_{i,2}, \dots, S_{i,2 \log n}$ of S_i (where $S_{i,1}$ is the low order bit) have been computed for each $i \in [1 : w]$.

Time: $O(\log^* n)$.

Time: $O(\log^* n)$.

3. Proceed similar as in step (2.c) to compute the bits of $S = z_1 + z_2 + \dots + z_n$ in an $w \times 2 \log n$ submesh from the bits $S_{i,1}, S_{i,2}, \dots, S_{i,2 \log n}$ of S_i , $i \in [1 : w]$.
Time: $O(\log^* n)$.
4. Since the w bits of S are spread over w processors it remains to form a binary number with value S .
 - i. If $w \leq \log n$ then the processor that holds the i th bit sends the value of this bit to all processors of the last column with row index between 2^{i-1} and $2^i - 1$. Now the sum S over these one-bit values is computed.
 - ii. If $w > \log n$ we divide the w bits of S into at most $j = \frac{w}{\log n}$ sequences of consecutive bits. The bits of each sequence are sent to the processors of a single column. Similar as before we compute a binary number for each sequence and multiply the j th number with $2^{(j-1) \log n}$, $j \in [1 : \frac{w}{\log n}]$. Then the sum of the $\frac{w}{\log n}$ values obtained is computed in time $O(\log \frac{w}{\log n})$.
Time: $O(\log \frac{w}{\log n})$.

We obtain the following lemma.

Lemma 2 Addition of n^2 w -bit numbers on an $n \times n$ APPB where each processor holds one of the numbers can be done with algorithm ADD in time $O(\log w + \log^* n)$.

Let an $n \times n$ matrix B be stored in an $n \times n$ APPB such that processor P_{ij} holds element b_{ij} of B . The algorithm NUMBER-NONZEROS below labels the nonzero elements of B in column-major order. The algorithm has a running time of $O(\log^* n)$. Let $f(b_{ij})$ be the corresponding number of nonzero element b_{ij} , $i, j \in [1 : n]$. Algorithm NUMBER-NONZEROS consists of three steps. In the first step the nonzero elements in each single column are numbered ($f^c(b_{ij})$ is the corresponding number of nonzero element b_{ij}). Then, in the second step the prefix sums over the numbers of nonzero elements in the columns are determined. Finally, each processor P_{ij} with a nonzero element b_{ij} computes $f(b_{ij})$ from $f^c(b_{ij})$ and the total

number of nonzero elements in columns 1 to $i - 1$.

Algorithm NUMBER-NONZEROS(B):

1. Number the nonzero elements of B in each column (Hint: Set a flag in each processor with a nonzero element and compute prefix sums over the flags in each column). Let $f^c(b_{ij})$ be the number of nonzero element b_{ij} and let f_i^c be the total number of nonzero elements in column i .

Time: $O(1)$.

2. Compute the prefix sums over $f_1^c, f_2^c, \dots, f_n^c$ as follows. Let $f_{[1:i]}^c = f_1^c + f_2^c + \dots + f_i^c$ for $i \in [1 : n]$.

- (a) Send f_i^c to all processors in the first $\log n$ processors of column $i, i \in [1 : n]$.

Time: $O(1)$.

- (b) For $j \in [1 : \log n]$ determine the prefix sums over the j th bit of the f_i^c 's in row j and multiply the results with 2^{j-1} .

Time: $O(1)$.

- (c) For each column $i, i > 1$ compute the sum of the values obtained in step (2.b) in an $\log n \times \log n$ submesh to obtain $f_{[1:i]}^c$. This is done as in step (3) of algorithm ADD but with the difference that, sums over bits are now computed only in the rows of each submesh (Technical details are left to the reader). Finally, for each $i \in [2 : n]$ a binary number is formed of the bits of each sum $f_{[1:i]}^c$ that are spread over $\log n$ processors similarly as in step (4.i) of algorithm ADD.

Time: $O(\log^* n)$.

3. For $i \in [1 : n - 1]$ send $f_{[1:i]}^c$ to all processors in column $i + 1$. Then each processor P_{ij} with a nonzero element determines $f(b_{ij}) = f_{[1:i-1]}^c + f^c(b_{ij})$ if $i \in [2 : n]$ and $f(b_{ij}) = f^c(b_{ij})$ if $i = 1$.

Time: $O(1)$.

Lemma 3 *The nonzero elements of an $n \times n$ matrix stored in an $n \times n$ APPB can be numbered in column-major order with algorithm NUMBER-NONZEROS in time $O(\log^* n)$.*

Finally we describe a technique for labelling the nonzero elements within each submatrix of a partitioned matrix. First we introduce the following definitions. For an $i \in [1 : n]$ let $P_{i,j_1}, P_{i,j_2}, \dots, P_{i,j_m}, 1 \leq j_1 < j_2 < \dots < j_m \leq n$ be the processors in row i of the APPB that store a nonzero number in their register X . To *compress* the contents of the registers X of the processors in row i to the left means that processor P_{i,j_h} sends the nonzero number stored in its register

X to processor $P_{i,h}$. Then, each processor that has received a number stores it in its register X and the other processors set their registers X to zero. Clearly, it is possible to compress the contents of the registers X of the processors in all rows of the APPB to the left in time $O(1)$.

Assume that matrix B can be partitioned into k_B $n \times k_t$ submatrices $B_t, t \in [1 : k_B]$, such that B_t is of size $n \times k_t, k_1 + k_2 + \dots + k_{k_B} = n$, and each submatrix B_t contains exactly n nonzero elements. The algorithm NUMBER-NONZEROS-BLOCKWISE below labels the nonzero elements of each submatrix B_t in row-major order. The algorithm has a running time of $O(\max\{k_B, \log^* n\})$. Let $g(b_{ij})$ be the corresponding number of nonzero element b_{ij} .

Algorithm NUMBER-NONZEROS-BLOCKWISE consists of three steps. In the first step the parameter k_B is determined and the matrices $B_t, t \in [1 : k_B]$ are identified. In the second step the number g_i^t of nonzero elements in each row $i, i \in [1 : n]$ of every matrix $B_t, t \in [1 : k_B]$ is determined. Further, the nonzero elements in each row of every matrix B_t are numbered from left to right. Prefix sums over the values $g_i^t, i \in [1 : n]$ are computed for all $t \in [1 : k_B]$ in the third step. Then, using this prefix sums and the number of each nonzero element b_{ij} in its row of matrix B_t the values $g(b_{ij})$ are computed.

Algorithm NUMBER-NONZEROS-BLOCKWISE(B):

1. Number the nonzero elements of B in column major order with algorithm NUMBER-NONZEROS. Each nonzero element b_{ij} with $(t - 1)n + 1 \leq f(b_{ij}) \leq tn$ belongs to submatrix $B_t, t \in [1 : k_B]$.

Time: $O(\log^* n)$.

2. For each $t \in [1 : k_B]$ number in each row of the mesh the nonzero elements of B_t as described next. Let $g^t(b_{ij})$ be the corresponding number of nonzero element b_{ij} of B_t and let g_i^t be the total number of nonzero elements b_{ij} of B_t in row i .

- (a) In each row compress the numbers $f(b_{ij})$ obtained in (1) to the left.

Time: $O(1)$.

- (b) Each processor that has received in the last step a number $f(b_{ij})$ with $(t - 1)n + 1 \leq f(b_{ij}) \leq tn, t \in [1 : n]$ examines whether its right neighbour has also received a number with a value between $(t - 1)n + 1$ and tn . If this is not the case or if there is no right neighbour that has received a nonzero value then the processor sends its column index to processor $P_{i,t}, t \in [1 : k_B]$ if it has a value $f(b_{ij})$ with $(t - 1)n + 1 \leq f(b_{ij}) \leq tn$. Then, repeat the same but with "left" instead of "right".

Time: $O(1)$.

- (c) For $i \in [1 : n]$, $t \in [1 : k_B]$: If processor $P_{i,t}$ has received two column indices x_i^t, y_i^t , $x_i^t \leq y_i^t$ in the last step it determines $g_i^t = y_i^t - x_i^t + 1$ from these values and otherwise sets $g_i^t = 0$.

Time: $O(1)$.

- (d) For $i \in [1 : n]$, $t \in [1 : k_B - 1]$: If $g_i^t > 0$ processor $P_{i,t}$ sends x_i^t to all processors in its row containing a nonzero element b_{ij} of B_t .

Time: $O(1)$.

- (e) Number the nonzero elements in each row from left to right and let $g^*(b_{ij})$ be the number of element b_{ij} . Then each processor $P_{i,j}$ with a nonzero element b_{ij} computes $g^t(b_{ij}) = g^*(b_{ij}) - x_i^t + 1$.

Time: $O(1)$.

Time: $O(1)$.

3. FOR $l = 1$ TO $\frac{k_B \cdot \log^2 n}{n}$ DO

- (a) For each $t \in [(l - 1) \frac{n}{\log^2 n} + 1 : \min\{k_B, l \frac{n}{\log^2 n}\}]$ compute the prefix sums $g_{[1:i]}^t$ over the values g_i^t , $i \in [1 : n]$ in an $n \times \log^2 n$ submesh:

- i. Compute in an $n \times \log n$ submesh prefix sums over the j th bits, $j \in [1 : \log n]$ of the g_i^t 's, $i \in [1 : n]$ and multiply the result for the j th bit with 2^{j-1} . Then add the obtained values for each $i \in [1 : n]$ in an $\log n \times \log n$ submesh similarly as in step (3) of algorithm ADD. The bits of each result are spread over $\log n$ processors. The corresponding binary values are formed in the next step.

Time: $O(\log^* n)$.

- ii. FOR $t = (l - 1) \frac{n}{\log^2 n} + 1$ TO $\min\{k_B, l \frac{n}{\log^2 n}\}$ DO

Form a binary number of the bits of each sum $g_{[1:i]}^t$, $i \in [1 : n]$ that are spread over $\log n$ processors similarly as in step (4) of algorithm ADD in constant time.

Time: $O(\min\{k_B, \frac{n}{\log^2 n}\})$.

Time: $O(\max\{k_B, \log^* n\})$.

- (b) For each $i \in [1 : n - 1]$, $t \in [1 : k_B]$ send $g_{[1:i]}^t$ to all processors in row $i + 1$ that have a nonzero element b_{ij} of B_t . Then, for each $i \in [1 : n]$, $t \in [1 : k_B]$ a processor that has a nonzero element b_{ij} of B_t computes $g(b_{ij}) = g_{[1:i-1]}^t + g^t(b_{ij})$ if $i > 1$ and $g(b_{ij}) = g^t(b_{ij})$ if $i = 1$.

Time: $O(1)$.

Time: $O(\max\{k_B, \log^* n\})$.

Lemma 4 Let k_B $n \times k_t$ matrices B_t , $t \in [1 : k_B]$, $k_t \in [1 : n]$ each containing at most n nonzero elements are stored in disjoint submeshes of an $n \times n$ APPB. Then for all matrices B_1, B_2, \dots, B_{k_B} their nonzero elements can be numbered in row-major order with algorithm NUMBER-NONZEROS-BLOCKWISE in time $O(\max\{k_B, \log^* n\})$.

4 Matrix Multiplication Algorithm

In this section we describe algorithm MATR-MULT that multiplies two $n \times n$ matrices A and B with elements that can be represented by w bits and where B has at most $k_B \cdot n$ nonzero elements, $1 \leq k_B \leq n$ on an $n \times n$ APPB in time $O(k_B + \max\{\log w, \log \log n\})$. The idea of the algorithm is as follows. Matrix B is partitioned into $n \times k_t$ submatrices B_t , $t \in [1 : y]$, $k_1 + k_2 + \dots + k_y = n$ such that each submatrix contains at most n nonzero elements. Clearly we can choose $y \leq 2k_B$. For ease of description we assume $y = k_B$. A is multiplied with B such that the computation of the products of the elements of A with the elements on B_t is done before we multiply A with B_{t+1} , $t \in [1 : k_B - 1]$. The computation of the elements of the product matrix $C = A \times B$ from these products is done in a pipelined manner, i.e. after computing the products of the elements of A with the elements of B_t we add only so many partial sums of the C -elements of $A \times B_1, A \times B_2, \dots, A \times B_t$ as necessary to obtain free space for the computation of the products of elements of A with elements of B_{t+1} , $t \in [1 : k_B - 1]$. Finally, we add all the remaining partial sums necessary to compute the C -elements. In the following algorithm we assume $\log n \leq w \leq \frac{1}{2}\sqrt{n}$. Otherwise, easy modifications of algorithm MATR-MULT will show that A and B can be multiplied in time $O(k_B + \max\{\log w, \log \log n\})$.

We now give a more detailed outline of algorithm MATR-MULT. In the first two steps the parameter k_B is determined and the submatrices B_t , $t \in [1 : k_B]$ are identified. Also the numbers g_i^t and $g_{[1:i]}^t$ are computed for $t \in [1 : k_B]$, $i \in [1 : n]$. Further, for each nonzero element b_{ij} of B the numbers $f(b_{ij})$ and $g(b_{ij})$ are computed. In the third step for $t = 1$ to k_B the products of elements of A with the elements of B_t are computed. This is done such that in row $l \in [1 : n]$ all products of the elements of row l of A with elements of B_t are computed. Observe, that these products are already in their final row. The products are formed by sending each nonzero element b_{ij} of B_t to all processors in column $g(b_{ij})$. Then each nonzero element a_{li} of A is sent in its row to the processors with column indices $g_{[1:l-1]}^t + 1, g_{[1:l-1]}^t + 2, \dots, g_{[1:l-1]}^t + g_i^t$ (these processors contain all nonzero elements from row l of

B_t) and multiplied with the local nonzero element of B_t . To make the computation of the C -elements from the products easy the products are ordered in each row such that products that belong to the same target C -element are in neighbored processors. Since each processor has only a constant number of registers we will need to free registers before we can compute products of nonzero elements of A with nonzero elements of B_{t+1} . Therefore we form partial sums from some products that belong to the same C -element and route computed C -elements to their final processor in the row (the same is done with older partial sums for C -elements emerging from $A \times B_1, A \times B_2, \dots, A \times B_{t-1}$). We do this in such a way that we never use more than two registers per processor to store the products and partial sums for the C -elements. Note, that we do not compute all C -elements of $A \times B_t$ before we start computing $A \times B_{t+1}$ since this may take too much time (it can happen that n products belong to the same C -element). The remaining C -elements that have not been computed in step 3 are computed in the final step 4. This is done in two phases. In the first phase we sum so many partial sums that there remain in each row at most $\frac{n}{2\alpha^2}$ partial sums corresponding to at most $\frac{n}{2\alpha^2}$ C -elements where $\alpha = \max\{w, \log n\}$. In the second phase we first compute the sum of the j th bits of all remaining partial sums that correspond to the same C -element, $j \in [1 : w]$. Then, from the obtained w sums with at most $\log n$ bits each we compute the corresponding C -element in an $\alpha \times \alpha$ submesh and send the C -element to its final processor.

Algorithm MATR-MULT(A, B):

1. Number the nonzero elements of B in column-major order with algorithm NUMBER-NONZEROS and let $f(b_{ij})$ be the number of nonzero element b_{ij} . Let B_t be the submatrix of B that contains the nonzero elements with $(t - 1)n + 1 \leq f(b_{ij}) \leq tn, t \in [1 : k_B]$.
Time: $O(\log^* n)$.
2. For each $t \in [1 : k_B]$ number the nonzero elements of B_t in row-major order with steps 2 and 3 of algorithm NUMBER-NONZEROS-BLOCKWISE and let $g(b_{ij})$ be the corresponding number of nonzero element b_{ij} . Algorithm NUMBER-NONZEROS-BLOCKWISE also determines g_i^t which is the number of nonzero elements in row i of B_t as well as the prefix sums $g_{[1:i]}^t = g_1^t + g_2^t + \dots + g_i^t, t \in [1 : k_B], i \in [1 : n]$.
Time: $O(\max\{k_B, \log^* n\})$.
3. FOR $t = 1$ TO k_B DO
 - (a) Send each nonzero element b_{ij} of B_t together with its number $f(b_{ij})$ to all processors in column $g(b_{ij})$.

Time: $O(1)$.

- (b) For each $l \in [1 : n]$: i) send g_i^t and $g_{[1:l-1]}^t$ to each processor in column l with a nonzero element of A , ii) send each nonzero element a_{il} in its row to all processors with a column index $g_{[1:l-1]}^t + 1, g_{[1:l-1]}^t + 2, \dots, g_{[1:l-1]}^t + g_i^t$. Each of these processors contains a nonzero element of B_t that has to be multiplied with a_{il} .
Time: $O(1)$.
 - (c) Each processor that has received in the last step (3.b) a nonzero element a_{ij} multiplies it with the nonzero element b_{jh} that has been received in step (3.a).
Time: $O(1)$.
 - (d) Reorder the products obtained in the last step in the rows such that products which belong to the same C -element come to neighbored processors. This is done by sending the product $a_{ij} \cdot b_{jh}$ to the processor with column index $f(b_{jh}) - (t - 1)n$, where $(t - 1)n + 1 \leq f(b_{jh}) \leq tn$. The product is stored in the register X if $t > 1$ and in the register Y if $t = 1$.
Time: $O(1)$.
 - (e) IF $t > 1$ THEN
In every row of the mesh add the contents of some registers X with the same target C -element, send computed C -elements to their final destination, and compress the registers X to the left such that afterwards the registers X of processors in the right half of the mesh are all zero:
 - i. Each processor with an uneven column index sends the content of its register X to its right neighbour. There it is added to the content of the register X if both values have the same target C -element. If that was the case the register X of the sender is the set to zero.
Time: $O(1)$.
 - ii. Same as last step but now with “even” and “left” instead of “uneven” and “right”.
Time: $O(1)$.
 - iii. Send C -elements that have been computed to their final destination in the rows.
Time: $O(1)$.
 - iv. In every row compress the contents of the registers X to the left. Observe, that afterwards all registers X in the right half of the mesh are zero.
Time: $O(1)$.
- Time: $O(1)$.

(f) IF $t > 1$ THEN

Do the same as in step (3.e) but now with the registers Y instead of the registers X . Afterwards all registers Y in the right half of the mesh are zero. Then send the contents of the register X of each processor P_{ij} with $i \in [1 : n]$, $j \leq \frac{n}{2}$ to processor $P_{i, \frac{n}{2}+j}$ where it is stored in the register Y and set the register X of P_{ij} to zero. Now, all registers X are zero.

Time: $O(1)$.

Time: $O(k_B)$.

4. Do all the remaining additions necessary to compute the rest of the C -elements and send them to their final destinations, as follows. Let $\alpha = \max\{w, \log n\}$.

- (a) Partition the rows of the mesh in subarrays of length $2\alpha^2$ and add all partial sums (stored in the registers Y) with the same target C -element in every subarray. Send C -elements that have been computed to their final destination. Notice, that in every row there remain at most $\frac{n}{2\alpha^2}$ partial sums of at most $\frac{n}{2\alpha^2}$ C -elements that still have to be computed.

Time: $O(\log \alpha)$.

- (b) Partition the mesh into $n \alpha \times \frac{n}{\alpha}$ submeshes (each submesh is used to add the remaining partial sums of one row of the mesh). For each row send the i th remaining partial sum in the row, $i \in [1 : \frac{n}{\alpha^2}]$ to all processors in column i of the corresponding $\alpha \times \frac{n}{\alpha}$ submesh.

Time: $O(1)$.

- (c) For each $j \in [1 : \alpha]$ sum in row j of every $\alpha \times \frac{n}{\alpha}$ submesh all the j th bits corresponding to partial sums with the same target C -element (The details are left to the reader).

Time: $O(1)$.

- (d) In each $\alpha \times \frac{n}{\alpha}$ submesh it remains to compute for each of the at most $\frac{n}{2\alpha^2}$ remaining C -elements a sum of at most α numbers with at most $\log n$ bits each. This can be done similar to step (3) of algorithm ADD. Each sum is computed in an $\alpha \times \alpha$ submesh.

Time: $O(\log^* n)$.

- (e) In each $\alpha \times \alpha$ submesh form a binary number of the bits of each sum computed in the last step to obtain the corresponding C -element. Then, send the computed C -elements to their final destinations.

Time: $O(\log \alpha)$.

Time: $O(\log \alpha) = O(\max\{\log w, \log \log n\})$.

Now, we can state the main theorem.

Theorem 1 *Two $n \times n$ matrices A and B with elements that can be represented by $O(w)$ bits and where the number of nonzero elements of B is at most $k_B \cdot n$, $1 \leq k_B \leq n$ can be multiplied on an $n \times n \times q$ APPB in time $O(\frac{k_B}{q} + \max\{\log w, \log \log n\})$ for $1 \leq q \leq n$.*

Proof: With algorithm MATR-MULT two matrices A and B with elements that can be represented by $O(w)$ bits and where the number of nonzero elements of B is at most $k_B \cdot n$, $1 \leq k_B \leq n$ can be multiplied on an $n \times n$ APPB in time $O(k_B + \max\{\log w, \log \log n\})$. On an $n \times n \times q$ APPB matrix B can be first split into q submatrices B_1, B_2, \dots, B_q each one containing $\frac{k_B}{q} \cdot n$ nonzero elements (Using algorithm NUMBER-NONZEROS this needs time $O(\log^* n)$). Then, each matrix B_i , $i \in [1 : q]$ is multiplied with A on an $n \times n$ submesh in time $O(\frac{k_B}{q} + \max\{\log w, \log \log n\})$. \square

Corollary 1 *Two $n \times n$ matrices A and B with elements that can be represented by $O(\log n)$ bits can be multiplied on an $n \times n \times \frac{n}{r}$ APPB in time $O(r)$ for $r \geq \log \log n$.*

Corollary 2 *An $n \times n$ matrix A and an $n \times 1$ vector with elements that can be represented by $O(\log n)$ bits can be multiplied on an $n \times n$ APPB in time $O(\log \log n)$.*

5 Conclusion

We have described an algorithm for matrix multiplication on arrays of processors with pipelined optical buses (APPB). Attention was given to the case that one of the matrices is sparse. As a special case we obtained an algorithm for fast matrix-vector multiplication. Our results improve some results obtained in [7] for the stronger model of a reconfigurable array of processors with optical buses (AROB). It is an interesting question whether our algorithm can be improved when using an AROB instead of an APPB. In general, it is worth to compare the power of the APPB and the AROB for other problems.

References

- [1] Beresford-Smith B., Diessel O. & ElGindy H. (1996) Optimal algorithms for constrained reconfigurable meshes. *Journal of Parallel and Distributed Computing*, 39, 1, p. 74-78.
- [2] Guo Z., Mehlem R. M., Chiarulli R. W. & Levitan S. P. (1991) Pipelined communication in optically interconnected arrays. *Journal of Parallel and Distributed Computing*, 12, p. 269-282.

- [3] Middendorf M., Schmeck H., Schröder H., & Turner G. Multiplication of matrices with different sparseness properties on dynamically reconfigurable meshes. To be published in *VLSI-DESIGN*.
- [4] Miller R., Prasanna-Kumar V. K., Reisis D. I. & Stout Q. F. (1993) Parallel computations on reconfigurable meshes. *IEEE Trans. Comput.*, 42, 6, p. 678-692.
- [5] Nakano K. & Wada K. (1995) Integer summing algorithms on reconfigurable meshes. *Proceedings of the IEEE First Conf. on Algorithms and Architectures for Parallel Processing (ICA³PP 95)*, Brisbane, Australia, 19.-21. April, p. 187-196.
- [6] Park H., Kim H. J. & Prasanna V. K. (1993) An $O(1)$ time optimal algorithm for multiplying matrices on reconfigurable mesh. *Inform. Process. Lett.*, 47, p. 109-113.
- [7] Pavel S. & Akl S. G. (1995) On the power of arrays with reconfigurable optical buses. Technical Report No. 95-374, Dept. of Computing and Information Science, Queen's University, Kingston, Ontario.
- [8] Pavel S. & Akl S. G. Matrix operations using arrays with reconfigurable optical buses. to be published in *Journal of Parallel Algorithms and Applications*.
- [9] Qiao C. (1995) Efficient matrix multiplication in a reconfigurable array with spanning optical buses. *Proceedings of the Fifth Symp. on the Frontiers of Massively Parallel Computing (Frontiers '95)*, McLean, Virginia, 6.-9. February, p. 273-279.

BPC Permutations on the OTIS-Hypercube Optoelectronic Computer¹

Sartaj Sahni and Chih-Fang Wang
 Department of Computer and Information Science and Engineering
 University of Florida, Gainesville, FL 32611
 Phone: 352-392-1527, Fax: 352-392-1220
 {sahni,wang}@cise.ufl.edu

Keywords: OTIS-Hypercube, BPC permutations, diameter

Edited by: Yi Pan

Received: October 30, 1997

Revised: January 20, 1998

Accepted: June 19, 1998

We show that the diameter of an N^2 processor OTIS-Hypercube computer ($N = 2^d$) is $2d + 1$. OTIS-Hypercube algorithms for some commonly performed permutations – transpose, bit reversal, vector reversal, perfect shuffle, unshuffle, shuffled row-major, and bit shuffle – are developed. We also propose an algorithm for general BPC permutations.

1 Introduction

Electronic interconnects are superior to optical interconnects when the interconnect distance is up to a few millimeters (Feldman et. al. 1988, Kiamilev et. al. 1991). However, for longer interconnects, optics (and in particular, free space optics) provides power, speed, and bandwidth advantages over electronics. With this in mind, Marsden et. al. (Marsden et. al. 1993), Hendrick et. al. (Hendrick et. al. 1995), and Zane et. al. (Zane et. al. 1996) have proposed a hybrid computer architecture in which the processors are divided into groups; intra-group connects are electronic, and inter-group interconnects are optical. Krishnamoorthy et. al. (Krishnamoorthy et. al. 1992) have demonstrated that bandwidth and power consumption are minimized when the number of processors in a group equals the number of groups. Marsden et. al. (Marsden et. al. 1993) propose a family of optoelectronic architectures in which the number of groups equals the number of processors per group. In this family – the optical transpose interconnection system (OTIS) – the inter-group connects (or optical interconnect) connect processor p of group g to processor g of group p . The intra-group interconnect (or electronic interconnect) can be any of the standard previously studied connection schemes for electronic computers. This strategy gives rise to the OTIS-Mesh, OTIS-Hypercube, OTIS-Perfect shuffle, OTIS-Mesh of trees, and so forth computers.

Figure 1 shows a generic 16 processor OTIS computer; only the optical connections are shown. The solid squares indicate individual processors, and a pro-

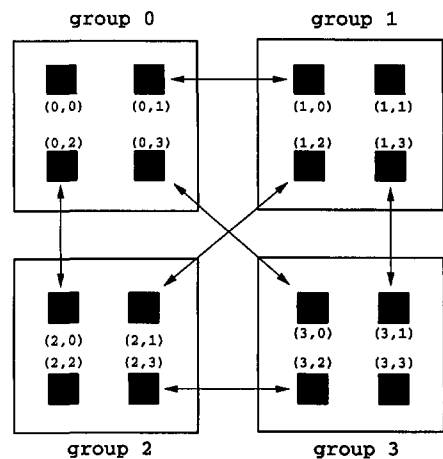


Figure 1: Example of OTIS connections with 16 processors

cessor index is given by the pair (G, P) where G is its group index and P the processor or local index. Figure 2 shows a 16 processor OTIS-Hypercube. The number inside a processor is the processor index within its group.

Hendrick et. al. (Hendrick et. al. 1995) have computed the performance characteristics (power, throughput, volume, etc.) of the OTIS-Hypercube architecture. Zane et. al. (Zane et. al. 1996) have shown that each move of an N^2 processor hypercube can be simulated by an N^2 processor OTIS-Hypercube using either one local electronic move, or one local electronic move and two optical inter group move using the OTIS interconnection. We shall refer the latter as OTIS moves. Sahni and Wang (Sahni & Wang 1997) and Wang and Sahni (Wang & Sahni 1997) have evalu-

¹This work was supported, in part, by the Army Research Office under grant DAA H04-95-1-0111.

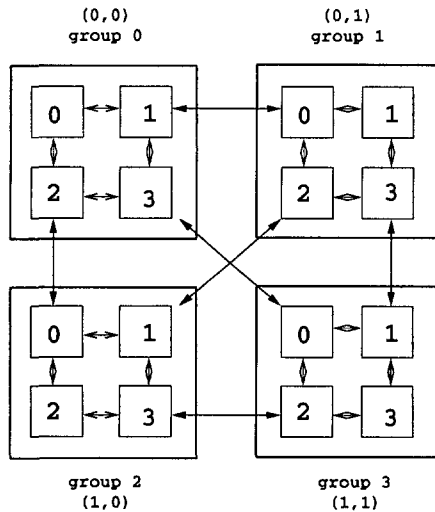


Figure 2: 16 processor OTIS-Hypercube

ated thoroughly the characteristics of the OTIS-Mesh architecture, developing algorithms for basic data rearrangements..

In this paper, we study the OTIS-Hypercube architecture and obtain basic properties and basic permutation routing algorithms for this architecture. These algorithms can be used to develop efficient application programs.

In the following, when we describe a path through an OTIS-Hypercube, we use the term **electronic move** to refer to a move along an electronic interconnect (so it is an intra-group move) and the **OTIS move** to refer to a move along an optical interconnect.

2 OTIS-Hypercube Diameter

Let $N = 2^d$ and let $d(i, j)$ be the length of the shortest path from processor i to processor j in a hypercube. Let (G_1, P_1) and (G_2, P_2) be two OTIS-Hypercube processors. The shortest path between these two processors fits into one of the following categories:

- (a) The path employs electronic moves only. This is possible only when $G_1 = G_2$.
- (b) The path employs an even number of OTIS moves. Paths of this type look like $(G_1, P_1) \xrightarrow{E} (G_1, P'_1) \xrightarrow{O} (P'_1, G_1) \xrightarrow{E} (P'_1, G'_1) \xrightarrow{O} (G'_1, P'_1) \xrightarrow{E} (G'_1, P''_1) \xrightarrow{O} (P''_1, G'_1) \xrightarrow{E} (P''_1, G''_1) \xrightarrow{O} \dots \xrightarrow{E} (G_2, P_2)$.

Here E denotes a sequence (possibly empty) of electronic moves and O denotes a single OTIS move. If the number of OTIS moves is more than two, we may compress the path into a shorter path that uses 2 OTIS moves only: $(G_1, P_1) \xrightarrow{E} (G_1, P_2) \xrightarrow{O} (P_2, G_1) \xrightarrow{E} (P_2, G_2) \xrightarrow{O} (G_2, P_2)$.

- (c) The path employs an odd number of OTIS moves. Again, if the number of moves is more than one, we can compress the path into a shorter one that employs exactly one OTIS move as in (b). The shorter path looks like: $(G_1, P_1) \xrightarrow{E} (G_1, G_2) \xrightarrow{O} (G_2, G_1) \xrightarrow{E} (G_2, P_2)$.

Shortest paths of type (a) have length exactly $d(P_1, P_2)$ (which equals the number of ones in the binary representation of $P_1 \oplus P_2$). Paths of type (b) and type (c) have length $d(P_1, P_2) + d(G_1, G_2) + 2$ and $d(P_1, G_2) + d(P_2, G_1) + 1$, respectively.

As a result, we obtain the following theorem:

Theorem 1 *The length of the shortest path between processors (G_1, P_1) and (G_2, P_2) is $d(P_1, P_2)$ when $G_1 = G_2$ and $\min\{d(P_1, P_2) + d(G_1, G_2) + 2, d(P_1, G_2) + d(P_2, G_1) + 1\}$ when $G_1 \neq G_2$.*

Theorem 2 *The diameter of the OTIS-Hypercube is $2d + 1$.*

Proof Since each group is a d -dimensional hypercube, $d(P_1, P_2)$, $d(G_1, G_2)$, $d(P_1, G_2)$, and $d(P_2, G_1)$ are all less than or equal to d . From theorem 1, we conclude that no two processors are more than $2d + 1$ apart. Now consider the processors (G_1, P_1) , (G_2, P_2) such that $P_1 = 0$ and $P_2 = N - 1$. Let $G_1 = 0$ and $G_2 = N - 1$. So $d(P_1, P_2) = d(G_1, G_2) = d(P_1, G_2) = d(P_2, G_1) = d$. Hence, the distance between (G_1, P_1) and (G_2, P_2) is $2d + 1$. As a result, the diameter of the OTIS-Mesh is exactly $2d + 1$. \square

3 Common Data Rearrangements

In this section, we concentrate on the realization of permutations such as transpose, perfect shuffle, unshuffle, vector reversal which are frequently used in applications. Nassimi and Sahni (Nassimi & Sahni 1982) have developed optimal hypercube algorithms for these frequently used permutations. These algorithms may be simulated by an OTIS-Hypercube using the method of (Zane et. al. 1996) to obtain algorithms to realize these data rearrangement patterns on an OTIS-Hypercube. Table 1 gives the number of moves used by the optimal hypercube algorithms; a break down of the number of moves in the group and local dimensions; and the number of electronic and OTIS moves required by the simulation.

We shall obtain OTIS-Hypercube algorithms, for the permutations of Table 1, that require far fewer moves than the simulations of the optimal hypercube algorithms.

As mentioned before, each processor is indexed as (G, P) where G is the group index and P the local index. An index pair (G, P) may be transformed into

Permutation	Optimal Hypercube Moves			OTIS-Hypercube Simulation	
	total	group dimension	local dimension	OTIS	electronic
Transpose	$2d$	d	d	$2d$	$2d$
Perfect Shuffle	$2d$	d	d	$2d$	$2d$
Unshuffle	$2d$	d	d	$2d$	$2d$
Bit Reversal	$2d$	d	d	$2d$	$2d$
Vector Reversal	$2d$	d	d	$2d$	$2d$
Bit Shuffle	$2d - 2$	$d - 1$	$d - 1$	$2d - 2$	$2d - 2$
Shuffled Row-major	$2d - 2$	$d - 1$	$d - 1$	$2d - 2$	$2d - 2$
$G_i P_u$ Swap	d	$d/2$	$d/2$	d	d

Table 1: Optimal moves for $N^2 = 2^{2d}$ processor hypercube and respective OTIS-Hypercube simulations

a singleton index $I = GP$ by concatenating the binary representations of G and P .

The permutations of Table 1 are members of the BPC (bit-permute-complement) class of permutations defined in (Nassimi & Sahni 1982). In a BPC permutation, the destination processor of each data is given by a rearrangement of the bits in the source processor index. For the case of our N^2 processor OTIS-Hypercube we know that N is a power of two and so the number of bits needed to represent a processor index is $p = \log_2 N^2 = 2 \log N = 2d$. A BPC permutation (Nassimi & Sahni 1982) is specified by a vector $A = [A_{p-1}, A_{p-2}, \dots, A_0]$ where

- (a) $A_i \in \{\pm 0, \pm 1, \dots, \pm(p-1)\}, 0 \leq i < p$ and
- (b) $[|A_{p-1}|, |A_{p-2}|, \dots, |A_0|]$ is a permutation of $[0, 1, \dots, p-1]$.

The destination for the data in any processor may be computed in the following manner. Let $m_{p-1}m_{p-2} \dots m_0$ be the binary representation of the processor's index. Let $d_{p-1}d_{p-2} \dots d_0$ be that of the destination processor's index. Then,

$$d_{|A_i|} = \begin{cases} m_i & \text{if } A_i \geq 0, \\ 1 - m_i & \text{if } A_i < 0. \end{cases}$$

In this definition, -0 is to be regarded as < 0 , while $+0$ is ≥ 0 .

In a 16-processor OTIS-Hypercube, the processor indices have four bits with the first two giving the group number and the second two the local processor index. The BPC permutation $[-0, 1, 2, -3]$ requires data from each processor $m_3m_2m_1m_0$ to be routed to processor $(1 - m_0)m_1m_2(1 - m_3)$. Table 2 lists the source and destination processors of the permutation.

The permutation vector A for each of the permutations of Table 1 is given in Table 3.

3.1 Transpose

$$[p/2 - 1, \dots, 0, p - 1, \dots, p/2]$$

The transpose operation may be accomplished via a single OTIS move and no electronic moves. The sim-

ulation of the optimal hypercube algorithm, however, takes $2d$ OTIS and $2d$ electronic moves.

3.2 Perfect Shuffle $[0, p - 1, p - 2, \dots, 1]$

We can adapt the strategy of (Nassimi & Sahni 1982) to an OTIS-Hypercube. Each processor uses two variables A and B . Initially, all data are in the A variables and the B variables have no data. The algorithm for perfect shuffle is given below:

Step 1: Swap A and B in processors with last two bits equal to 01 or 10.

Step 2: for $(i = 1; i \leq d - 1; i++)$ {

- (a) Swap the B variables of processors that differ on bit i only;
- (b) Swap the A and B variables of processors with bit i of their index I not equal to bit $i + 1$ of their index; }

Step 3: Perform an OTIS move on the A and B variables.

Step 4: for $(i = 0; i \leq d - 1; i++)$ {

- (a) Swap the B variables of processors that differ on bit i only;
- (b) Swap the A and B variables of processors with bit i of their index I not equal to bit $i + 1$ of their index; }

Step 5: Perform an OTIS move on the A and B variables.

Step 6: Swap the B variables of processors that differ on bit 0 only.

Step 7: Swap the A and B variables of processors with last two bits equal to 01 or 10.

Actually, in Step 1 it is sufficient to copy from A to B , and in Step 7 to copy from B to A .

Source			Destination		
Processor	(G, P)	Binary	Binary	(G, P)	Processor
0	(0,0)	0000	1001	(2,1)	9
1	(0,1)	0001	0001	(0,1)	1
2	(0,2)	0010	1101	(3,1)	13
3	(0,3)	0011	0101	(1,1)	5
4	(1,0)	0100	1011	(2,3)	11
5	(1,1)	0101	0011	(0,3)	3
6	(1,2)	0110	1111	(3,3)	15
7	(1,3)	0111	0111	(1,3)	7
8	(2,0)	1000	1000	(2,0)	8
9	(2,1)	1001	0000	(0,0)	0
10	(2,2)	1010	1100	(3,0)	12
11	(2,3)	1011	0100	(1,0)	4
12	(3,0)	1100	1010	(2,2)	10
13	(3,1)	1101	0010	(0,2)	2
14	(3,2)	1110	1110	(3,2)	14
15	(3,3)	1111	0110	(1,2)	6

Table 2: Source and destination of the BPC permutation $[-0, 1, 2, -3]$ in a 16 processor OTIS-Hypercube

Permutation	Permutation Vector
Transpose	$[p/2 - 1, \dots, 0, p - 1, \dots, p/2]$
Perfect Shuffle	$[0, p - 1, p - 2, \dots, 1]$
Unshuffle	$[p - 2, p - 3, \dots, 0, p - 1]$
Bit Reversal	$[0, 1, \dots, p - 1]$
Vector Reversal	$[-(p - 1), -(p - 2), \dots, -0]$
Bit Shuffle	$[p - 1, p - 3, \dots, 1, p - 2, p - 4, \dots, 0]$
Shuffled Row-major	$[p - 1, p/2 - 1, p - 2, p/2 - 2, \dots, p/2, 0]$
G_1P_u Swap	$[p - 1, \dots, 3p/4, p/2 - 1, \dots, p/4, 3p/4 - 1, \dots, p/2, p/4 - 1, \dots, 0]$

Table 3: Permutations and their permutation vectors

Table 4 shows the working of this algorithm on a 16 processor OTIS-Hypercube. The correctness of the algorithm is easily established, and we see that the number of data move step is $2d + 2$ ($2d$ electronic moves and 2 OTIS moves; each OTIS move moves two pieces of data from one processor to another, each electronic swap moves a single data between two processors).

The communication complexity of $2d+2$ is very close to optimal. For example, data from the processor with index $I = 0101 \dots 0101$ is to move to the processor with index $I' = 1010 \dots 1010$ and the distance between these two processors is $2d + 1$.

Notice that the simulation of the optimal hypercube algorithm for perfect shuffle takes $4d$ moves.

3.3 Unshuffle $[p - 2, p - 3, \dots, 0, p - 1]$

This is the inverse of a perfect shuffle and may be performed by running the perfect shuffle algorithm backwards (*i.e.*, beginning with Step 7); the for loops of Steps 2 and 4 are also run backwards. Thus the number of moves is the same as for a perfect shuffle.

3.4 Bit Reversal $[0, 1, \dots, p - 1]$

When simulating the optimal hypercube algorithm, the task requires $2d$ electronic moves and $2d$ OTIS moves. But with the following algorithm:

Step 1: Do a local bit reversal in each group.

Step 2: Perform an OTIS move of all data.

Step 3: Do a local bit reversal in each group.

we can actually achieve the rearrangement in $2d$ electronic moves and 1 OTIS move, since Steps 1 and 3 can be performed optimally in d electronic moves each (Nassimi & Sahni 1982).

The number of moves is optimal since the data from processor 0101...0101 is to move to processor 1010...1010, and the distance between these two processors is $2d + 1$ (Theorem 1).

index	initial	Step 1	Step 2		OTIS	Step 4				OTIS	Step 6	Step 7	variable
			<i>i</i> = 1			<i>i</i> = 0		<i>i</i> = 1					
			(a)	(b)		(a)	(b)	(a)	(b)				
0000	0	0	0	0	0	0	0	0	0	0	0	0	A
	-	-	2	2	2	4	4	8	8	8	-	-	B
0001	1	-	-	-	6	6	2	2	2	-	-	8	A
	-	1	-	-	4	2	6	10	10	-	8	-	B
0010	2	-	-	-	8	8	12	12	4	-	-	1	A
	-	2	-	-	10	12	8	4	12	-	1	-	B
0011	3	3	3	1	14	14	14	14	6	9	9	9	A
	-	-	1	3	12	10	10	6	14	1	-	-	B
0100	4	4	4	6	-	-	-	-	-	2	2	2	A
	-	-	6	4	-	-	-	-	-	10	-	-	B
0101	5	-	-	-	-	-	-	-	-	-	-	10	A
	-	5	-	-	-	-	-	-	-	-	10	-	B
0110	6	-	-	-	-	-	-	-	-	-	-	3	A
	-	6	-	-	-	-	-	-	-	-	3	-	B
0111	7	7	7	7	-	-	-	-	-	11	11	11	A
	-	-	5	5	-	-	-	-	-	3	-	-	B
1000	8	8	8	8	-	-	-	-	-	4	4	4	A
	-	-	10	10	-	-	-	-	-	12	-	-	B
1001	9	-	-	-	-	-	-	-	-	-	-	12	A
	-	9	-	-	-	-	-	-	-	-	12	-	B
1010	10	-	-	-	-	-	-	-	-	-	-	5	A
	-	10	-	-	-	-	-	-	-	-	5	-	B
1011	11	11	11	9	-	-	-	-	-	13	13	13	A
	-	-	9	11	-	-	-	-	-	5	-	-	B
1100	12	12	12	14	1	1	1	1	9	6	6	6	A
	-	-	14	12	3	5	5	9	1	14	-	-	B
1101	13	-	-	-	7	7	3	3	11	-	-	14	A
	-	13	-	-	5	3	7	11	3	-	14	-	B
1110	14	-	-	-	9	9	13	13	13	-	-	7	A
	-	14	-	-	11	13	9	5	5	-	7	-	B
1111	15	15	15	15	15	15	15	15	15	15	15	15	A
	-	-	13	13	13	11	11	7	7	7	-	-	B

Table 4: Illustration of the perfect shuffle algorithm on a 16 processor OTIS-Hypercube

3.5 Vector Reversal

$$[-(p - 1), -(p - 2), \dots, -0]$$

A vector reversal can be done using $2d$ electronic and 2 OTIS moves. The steps are:

Step 1: Perform a local vector reversal in each group.

Step 2: Do an OTIS move of all data.

Step 3: Perform a local vector reversal in each group.

Step 4: Do an OTIS move of all data.

The correctness of the algorithm is obvious. The number of moves is computed using the fact that Steps 1 and 3 can be done in d electronic moves each (Nasimi & Sahni 1982).

Since a vector reversal requires us to move data from processor $00 \dots 00$ to processor $11 \dots 11$, and since the

distance between these two processors is $2d+1$ (Theorem 1), our vector reversal algorithm can be improved by at most one move.

3.6 Bit Shuffle

$$[p - 1, p - 3, \dots, 1, p - 2, p - 4, \dots, 0]$$

Let $G = G_u G_l$ where G_u and G_l partition G in half. Same for $P = P_u P_l$. Our algorithm employs a $G_l P_u$ Swap permutation in which data from processor $G_u G_l P_u P_l$ is routed to processor $G_u P_u G_l P_l$. So we need to first look at how this permutation is performed.

3.6.1 G_1P_u Swap [$p-1, \dots, 3p/4, p/2-1, \dots, p/4, 3p/4-1, \dots, p/2, p/4-1, \dots, 0$]

The swap is performed by a series of bit exchanges of the form $B(i) = [B_{p-1}, \dots, B_0]$, $0 \leq i < p/4$, where

$$B_j = \begin{cases} p/2 + i, & j = p/4 + i \\ p/4 + i, & j = p/2 + i \\ j & \text{otherwise} \end{cases}$$

Let $G(i)$ and $P(i)$ denote the i th bit of G and P respectively. So $G(0)$ is the least significant bit in G , and $P(d)$ is the most significant bit in P . The bit exchange $B(i)$ may be accomplished as below:

Step 1: Every processor (G, P) with $G(i) \neq P(d/2 + i)$ moves its data to the processor (G, P') where P' differs from P only in bit $d/2 + i$.

Step 2: Perform an OTIS move on the data moved in Step 1.

Step 3: Processors (G, P) that receive data in Step 2 move the received data to (G, P'), where P' differs from P only in bit i .

Step 4: Perform an OTIS move on the data moved in Step 3.

The cost is 2 electronic moves and 2 OTIS moves.

To perform a G_1P_u Swap permutation, we simply do $B(i)$ for $0 \leq i < d/2$. This takes d electronic moves and d OTIS moves. By doing pairs of bit exchanges ($B(0), B(1)$), ($B(2), B(3)$), etc. together, we can reduce the number of OTIS moves to $d/2$.

3.6.2 Bit Shuffle

A bit shuffle, now, can be performed following these steps:

- Step 1:** Perform a G_1P_u swap.
- Step 2:** Do a local bit shuffle in each group.
- Step 3:** Do an OTIS move.
- Step 4:** Do a local bit shuffle in each group.
- Step 5:** Do an OTIS move.

Steps 2 and 4 are done using the optimal d move hypercube bit shuffle algorithm of (Nassimi & Sahni 1982). The total number of data moves is $3d$ electronic moves and $d/2 + 2$ OTIS moves.

3.7 Shuffled Row-major

$$[p-1, p/2-1, p-2, p/2-2, \dots, p/2, 0]$$

This is the inverse of a bit shuffle and may be done in the same number of moves by running the bit shuffle algorithm backwards. Of course, Steps 2 and 4 are to be changed to shuffled row-major operations.

4 BPC Permutations

Every BPC permutation A can be realized by a sequence of bit exchange permutations of the form $B(i, j) = [B_{2d-1}, \dots, B_0]$, $d \leq i < 2d$, $0 \leq j < d$, and

$$B_q = \begin{cases} j, & q = i \\ i, & q = j \\ q, & \text{otherwise,} \end{cases}$$

and a BPC permutation $C = [C_{2d-1}, \dots, C_0] = \Pi_G \Pi_P$ where $|C_q| < d$, $0 \leq q < d$, Π_G and Π_P involve d bits each.

For example, the transpose permutation may be realized by the sequence $B(d+j, j)$, $0 \leq j < d$; bit reversal is equivalent to the sequence $B(2d-1-j, j)$, $0 \leq j < d$; vector reversal can be realized by performing no bit exchanges and using $C = [-(2d-1), -(2d-2), \dots, -0]$ ($\Pi_G = [-(2d-1), -(2d-2), \dots, -d]$, $\Pi_P = [-(d-1), \dots, -0]$); and perfect shuffle may be decomposed into $B(d, 0)$ and $C = [2d-2, 2d-3, \dots, d, 2d-1, d-2, \dots, 1, 0, d-1]$ ($\Pi_G = [2d-2, 2d-3, \dots, d, 2d-1]$, $\Pi_P = [d-2, \dots, 1, 0, d-1]$).

A bit exchange permutation $B(i, j)$ can be performed in 2 electronic moves and 2 OTIS moves using a process similar to that used for the bit exchange permutation $B(i)$. Notice that $B(i) = B(i, i)$.

Our algorithm for general BPC permutations is:

Step 1: Decompose the BPC permutation A into the pair cycle moves $B_1(i_1, j_1)$, $B_2(i_2, j_2), \dots$, $B_k(i_k, j_k)$ and the BPC permutation $C = \Pi_G \Pi_P$ as above. Do this such that $i_1 > i_2 > \dots > i_k$, and $j_1 > j_2 > \dots > j_k$.

Step 2: If $k = 0$, do the following:

- Step 2.1:** Do the BPC permutation Π_P in each group using the optimal algorithm of (Nassimi & Sahni 1982).
- Step 2.2:** Do an OTIS move.
- Step 2.3:** Do the BPC permutation Π'_G in each group using the algorithm of (Nassimi & Sahni 1982).
- Step 2.4:** Do an OTIS move.

Step 3: If $k = d$, do the following:

- Step 3.1:** Do the BPC permutation Π'_G in each group.
- Step 3.2:** Do an OTIS move.
- Step 3.3:** Do the BPC permutation Π_P in each group.

Step 4: If $k < d/2$, do the following:

- Step 4.1:** Perform the bit exchange permutation B_1, \dots, B_k .

Permutation	Simulation		Our Algorithm	
	OTIS	electronic	OTIS	electronic
Transpose	$2d$	$2d$	1	0
Perfect Shuffle	$2d$	$2d$	2	$2d$
Unshuffle	$2d$	$2d$	2	$2d$
Bit Reversal	$2d$	$2d$	1	$2d$
Vector Reversal	$2d$	$2d$	2	$2d$
Bit Shuffle	$2d - 2$	$2d - 2$	$d/2 + 2$	$3d$
Shuffled Row-major	$2d - 2$	$2d - 2$	$d/2 + 2$	$3d$
$G_l P_u$ Swap	d	d	$d/2$	d

Table 5: Performance Comparisons

Step 4.2: Do Steps 2.1 through 2.4.

Step 5: If $k \geq d/2$, do the following:

Step 5.1: Perform a sequence of $d - k$ bit exchanges involving bits other than those in B_1, \dots, B_k in the same orderly fashion described in Step 1. Recompute Π_G and Π_P . Swap Π_G and Π_P .

Step 5.2: Do Steps 3.1 through 3.3.

The local BPC permutations determined by Π_G and Π_P take at most d electronic moves each (Nassimi & Sahni 1982); and the bit exchange permutations take at most d electronic moves and $d/2$ OTIS moves. So the total number of moves is at most $3d$ electronic moves and $d/2 + 2$ OTIS moves.

5 Conclusion

In this paper we have shown that the diameter of the OTIS-Hypercube is $2d + 1$, which is very close to that of an N^2 processor hypercube. However, each OTIS-Hypercube processor is connected to at most $d + 1$ other processors; while in an N^2 processor hypercube, a processor is connected to up to $2d$ other processors. We have also developed algorithms for frequently used data permutations. Table 5 compares the performance of our algorithms and those obtained by simulating the optimal hypercube algorithms using the simulation technique of (Zane et. al. 1996). For most of the permutations considered, our algorithms are either optimal or within one move of being optimal. An algorithm for general BPC permutations has also been proposed.

References

- [1] Feldman M., Esener S., Guest C., & S. Lee (1988) Comparison Between Electrical and Free-Space Optical Interconnects Based on Power and Speed Considerations. *Applied Optics*, 27, 9, p. 1742-1751.
- [2] Hendrick W., Kibar O., Marchand P., Fan C., Blerkom D. V., McCormick F., Cokgor I., Hansen M., & Esener S. (1995) Modeling and Optimization of the Optical Transpose Interconnection System. Optoelectronic Technology Center, Program Review, Cornell University.
- [3] Kiamilev F., Marchand P., Krishnamoorthy A., Esener S., & Lee S. (1991) Performance Comparison Between Optoelectronic and VLSI Multistage Interconnection Networks. *Journal of Lightwave Technology*, 9, 12, p. 1674-1692.
- [4] Krishnamoorthy A., Marchand P., Kiamilev F., & Esener S. (1992) Grain-Size Considerations for Optoelectronic Multistage Interconnection Networks. *Applied Optics*, 31, 26, p. 5480-5507.
- [5] Marsden G. C., Marchand P. J., Harvey P., & Esener S. C. (1993) Optical Transpose Interconnection System Architectures. *Optics Letters*, 18, 13, p. 1083-1085.
- [6] Nassimi D. & Sahni S. (1982) Optimal BPC Permutations On A Cube Connected Computer. *IEEE Transactions on Computers*, C-31, 4, p. 338-341.
- [7] Sahni S. & Wang C.-F. (1997) BPC Permutations On The OTIS-Mesh Optoelectronic Computer. *Proceedings of the fourth International Conference on Massively Parallel Processing Using Optical Interconnections (MPPOI'97)*, p. 130-135.
- [8] Wang C.-F. & Sahni S. (1997) Basic Operations on the OTIS-Mesh Optoelectronic Computer. *Technical Report 97-008*, CISE Department, University of Florida, available by anonymous ftp login from ftp.cise.ufl.edu under directory tech-report/tr97/tr97-008.ps.gz.
- [9] Zane F., Marchand P., Paturi R., & Esener S. (1996) Scalable Network Architectures Using the Optical Transpose Interconnection System (OTIS). *Proceedings of the second International Conference on Massively Parallel Processing Using Optical Interconnections (MPPOI'96)*, p. 114-121.

A Framework Supporting Specialized Electronic Library Construction

Daniel J. Helm, Jerry W. Cogle Jr. and Raymond J. D'Amore
 The Mitre Corporation, 1820 Dolley Madison Blvd.,
 McLean, VA 22102-3481, USA
 Phone: 703 883 6899, Fax: 703 883 7978
 Email: (dhelm,jcogle,rdamore)@mitre.org

Keywords: digital library, information retrieval, system architecture

Edited by: Rudi Murn

Received: March 25, 1998

Revised: July 28, 1998

Accepted: September 8, 1998

The Collaborative Electronic Library Framework (CELF) is an architecture enabling a group of users to build a specialized digital collection organized around a set of topical areas of interest to a community of users. The system provides services to collect network-based information and tools to publish collected information into different topical areas through both manual and automatic mechanisms. Services are also provided to enable users to locate and retrieve useful information from the repository by browse and search techniques. All access to CELF is via a Web browser.

1 Introduction

In recent years, with the proliferation of the World Wide Web, digital libraries have begun to flourish. There have been many viewpoints on what a digital library is or should be and many proposed approaches for their construction (Levy et. al. 95). Most existing digital libraries are built using manual labor intensive techniques, where a group of dedicated individuals collect and organize information. Other approaches have included more automatic techniques, although the resulting systems are typically less organized and coherent. Our approach is to fuse both manual and automatic approaches for digital library construction by leveraging useful benefits from each technique. These dual approaches are implemented in a prototype system called the Collaborative Electronic Library Framework (CELF).

CELF provides an environment for developing an electronic library using an integrated suite of advanced data processing tools. The system incorporates data collection utilities and categorization services for automatically assigning collected documents into different topical areas. Document contribution and review services are provided to enable manually submitted information to be added to the library. A browse and search subsystem is also provided to enable users to effectively access and "mine" the information repository. Certain concepts of CELF have evolved from the MIDS project (Helm et. al. 96), which is an ongoing MITRE sponsored research project involving research and development in the area of information organization and discovery.

CELF information is organized around topical hier-

archies (taxonomies) that provide effective information organization and knowledge discovery supporting general and very specific information needs. The actual underlying structure is similar in concept to Yahoo (Yahoo) and includes similar tools and services such as search and topic-based browsing. Notable differences are that CELF provides a work flow framework for effective collaborative content insertion and management, and also provides advanced tools to automatically collect and categorize documents into taxonomies. These advanced tools are critical in supporting effective content creation and management, and will be discussed in subsequent sections.

One basic principle behind CELF is to provide a framework whereby a group of users can build a specialized collection that is of value to a particular special interest community. A pilot project is underway to test the feasibility of the concept, where a group of participants were selected to build a specialized collection using the CELF framework.

The primary focus of this paper is to describe the functionality of CELF and to discuss the currently fielded operational prototype supporting a pilot program; a future paper is planned which will provide an analysis and experimental evaluation of key component technologies used within CELF.

1.1 System Overview

CELF includes two browsable and searchable repositories: a "core" manually generated high-quality collection, referred to as "library," and an automatically generated collection, referred to as "sources." The sources repository contains documents which are col-

lected and categorized by automatic techniques, while the library repository contains documents that are manually published by a group of users. Both the library and sources repositories are available for end-user access. The sources repository also serves as a primary collection that document contributors will access as a basis for locating documents for inclusion in the core library collection.

Information submitted to the library is initially entered into an HTML form interface, which enables meta-data describing a document to be provided, such as URL, title, description, abstract, and topical categories. Documents for which meta-data are entered can be obtained from many different sources, including the CELF sources repository, Intranets, and the Web. In general, only meta-data is actually stored within CELF for all collected documents. A URL meta-data field provides access to the full document stored at the remote location, although an option is provided on the document submission form to specify that a copy be made of the remote document. For copied documents, the URL meta-data field points to the local copy and an "external-URL" meta-data field provides access to the original remote document. A publishing option is also provided for a document contributor to create a document on their local computer and upload it to CELF.

After a document is submitted for the library repository, it goes into a "pending review" state, where a document reviewer will approve or reject the document using a review interface. Approved documents are periodically added and indexed in the library repository where they are made available to end-users via the browse and search interfaces. Rejected documents are removed from the system. A document reviewer can also add additional comments and/or update meta-data for a document prior to the approval stage.

2 Detailed System Description

CELF provides a set of integrated services including document collection, publishing, indexing, categorizing, browsing and retrieval, to provide a framework by which a specialized collection can be built. Software comprising CELF includes the Netscape Catalog Server 1.0 product (Catalog Server), as well as custom software developed by The MITRE Corporation.

2.1 Netscape Catalog Server

The Netscape Catalog Server product evolved from the Harvest (Bowman et. al. 94) research prototype system that provides an efficient information collection and retrieval framework. The Netscape Catalog Server framework includes two primary server components: a Resource Description Server (RDS) and Catalog Server. An RDS stores meta-data manually sub-

mitted by users or automatically collected by robots. Robots are specialized subsystems that can be configured to collect information from the Web and other repositories. A Catalog Server provides a search and browse interface to data collected and indexed from one or more RDS's. An integrated Verity (Verity) search engine provides the underlying retrieval framework for the Catalog Server.

2.2 System Architecture

CELF contains a configuration of RDS's and Catalog Servers as shown in Fig. 1. The figure shows three RDS's (Harvest RDS, Agent RDS and Manual RDS) and two Catalog Servers (Library Catalog and Sources Catalog). The Harvest RDS is a specialized subsystem that is used to recursively collect documents from different Web sites. The top level URL's from which to collect are manually specified in a configuration file. A robot associated with the RDS periodically collects Web pages beginning at the top level URL's specified. Meta-data extracted from the collected pages are stored in a document database associated with the RDS.

The Agent RDS is a customized subsystem that contains a search agent (Genesereth et. al. 94) that is configured to contact Web-based search engines as a basis for providing URL's for which an associated robot will collect documents. The search agent uses CELF keyword-based topical profiles (also used to automatically categorize collected documents) as canned queries to the search engines. URL's returned from the contacted search engines are then used to "seed" a robot configuration file (as with the Harvest RDS). The robot associated with the Agent RDS periodically collects the requested documents and stores their meta-data in the RDS's document database.

The Manual RDS stores the meta-data manually submitted to CELF. These manually submitted documents are stored in a database, which a select group of users will access during the review process. Approved documents will be flagged accordingly and rejected documents will be automatically removed from the document database.

The Sources Catalog is a browsable and searchable collection of documents that have been automatically collected by the Agent RDS and Harvest RDS. A categorization module augments this subsystem, by periodically categorizing the collected and indexed documents using the CELF topical profiles. Information not categorized can only be retrieved using the keyword-based search interface.

The Library Catalog is a browsable and searchable collection of "core" documents that have been manually submitted by users and approved by document reviewers. This subsystem periodically collects and indexes documents from the Manual RDS.

The client interface is a customized subsystem that enables users as well as document contributors to browse and search for documents contained in both the library and sources repository. Different options are available to affect the search process as well as the format of retrieval results.

CELF also includes an administrative interface that enables privileged users to perform such operations as document contribution, document review, as well as the administration of different servers and configuration files used in CELF. All features of CELF, including the administrative functions, are available via a Web browser.

2.2.1 Automatic Document Collection Methods

Two automatic data collection mechanisms are used within CELF: Agent and Harvest. The two mechanisms have unique features although ultimately both utilize robots for contacting Web sites, retrieving HTML pages and extracting meta-data for each document. The meta-data are stored in RDS's pending import into the Sources Catalog. Both automatic collection mechanisms are employed to collect large focused quantities of data from the Internet and Intranets. The desired effect of these collections for the document contributor, is sufficient recall from the Sources Catalog such that manual Internet searching need not routinely be performed. Both robots are seeded with a list of URL's to be collected.

The Harvest mechanism is configured to collect all meta-data from linked pages at specified URL's. User configurable parameters are also supported to limit the number of levels to recursively retrieve pages, and whether to include or exclude certain file types. The Harvest mechanism is scheduled to collect at periodic time intervals and, depending on the number of starting URL's and depth of recursion, can collect large volumes of information. A general problem with the harvest-based collection approach is that although large quantities of information can be retrieved, the relevancy of the average document is typically very low. It is a brute force approach that collects many non-relevant documents for the sake of locating a smaller number of "golden nuggets" of information. To augment this approach, we developed an Agent-based collection scheme that can perform more precise collection.

The Agent RDS collects data from pages identified as containing relevant information. The URL's are identified using The MITRE Corporation's search agent extension. This collection approach makes use of a set of profiles defined for each topic. These profiles were initially generated automatically using the names of the CELF topics in conjunction with Boolean logic; however, profiles can be enhanced manually and au-

tomatically to produce more focused search results. For automatic profile adaptation, the system can automatically extend profiles using statistically significant words and phrases extracted from sets of documents. These documents can include previously collected and categorized documents and/or documents deemed relevant via user relevance judgments. The core component of this subsystem is a search agent. The agent contacts a pre-configured list of Internet search engines using the profile queries to conduct canned searches for new pages. New URL results are stored and used to seed the Agent robot collection. The Agent robot is configured to collect a single page for each URL in the configuration list. This scheme provides a fairly precise collection mechanism by only retrieving pages that are of very high relevance to the CELF taxonomy. It is limited by the fact that only indexed collections can be contacted as a basis for document retrieval.

In concert, the two automatic collection mechanisms work to build the Sources Catalog. This repository contains information records which may be of interest but have not been manually reviewed. The purpose of this repository is to provide access to focused Internet information in a format familiar to both end users and data contributors.

2.2.2 Automatic Document Categorization

After documents have been retrieved and meta-data extracted for collected documents, a categorization module is invoked to classify all documents into the CELF topical hierarchy. This module uses topical profiles to classify all documents by comparing each profile to the document meta-data. These profiles are typically the same as those used in the Agent-based collection subsystem, although it is possible to utilize profiles unique to the categorization subsystem. The resulting output includes the topical labels applicable to each document. This meta-data is later indexed to support the CELF topic-based browse functionality.

At pre-defined time intervals, document meta-data collected via the Agent and Harvest subsystems are indexed by the integrated Verity search engine. A categorization module is then invoked, which uses the CELF profiles associated with each topic as canned queries to categorize the documents. For each query, a set of zero or more matching documents are returned. The categorization module will, for each returned document, set the "category" meta-data field to the topic used to perform the retrieval operation. Documents matching multiple queries/profiles are assigned all relevant topical labels. After the categorization module completes, the collection is re-indexed to make the newly assigned topical labels available to the browse interface.

The query-based categorization approach allows the

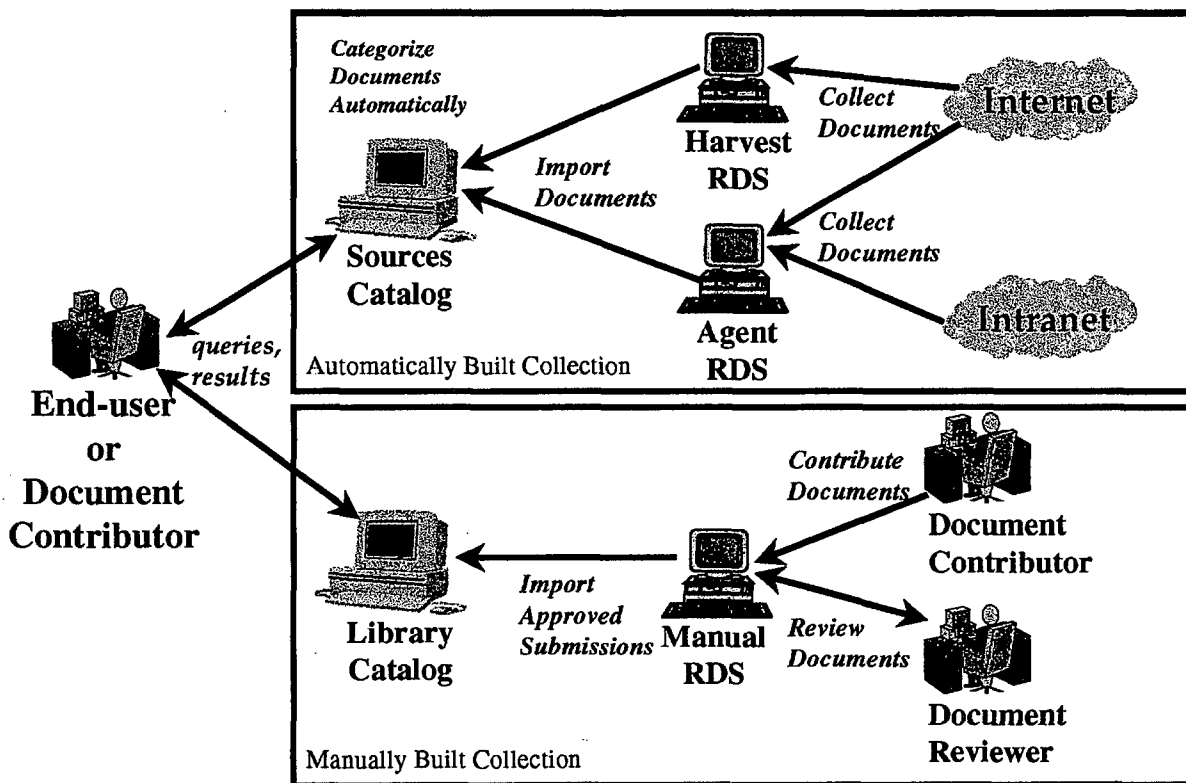


Figure 1: Collaborative Electronic Library Framework (CELf) Architecture

full power of the Verity search engine syntax to be utilized in our profiles, and also enables the entire collection to be re-categorized on-demand as new information items are added and/or profiles are modified.

2.3 Post-retrieval Tools

In addition to the backend collection, indexing, and categorization tools, CELf also supports post-retrieval tools to enable document contributors and end-users to effectively "mine" the information repository. Although full-text searching and topic-based browsing is supported, we have found the need to incorporate document clustering as a basis for further refining the list of documents returned on behalf of a particular query. Even when a user narrows a query down to a lower-level topic, it is still possible that a relatively large number of documents can be returned. Clustering provides an automatic method to break down large retrieval lists into dynamically generated subgroups (Rao et. al. 95). This aids the user by uncovering the salient themes in a list of returned documents. Fig. 2 provides example fragments of a standard document retrieval list (left screen) and a clustered retrieval list. The document list display presents meta-

data (e.g., title, description, URL) for each document. The clustered display depicts the document subgroups generated via a document clustering algorithm (Jain et. al. 88). The clustering module utilizes an adaptive nearest centroid sorting algorithm to group documents together whose similarity exceeds a configurable threshold. The cosine-similarity measure (Salton et. al. 83) is currently used to compute the similarity between documents. Each document subgroup is labeled with the top discriminating keywords and phrases that distinguish the set of documents. Documents with a higher score are more similar to the cluster summary.

Fig. 3 shows the major processing steps utilized in our clustering algorithm. The Vector Generator creates for each document, a list of the unique terms with corresponding frequency counts. All insignificant words (e.g., stop words) are excluded. The vectors are then passed to the Cluster Generator, which groups similar vectors together based on their cosine-similarity score. The results of the Cluster Generator are then passed to the Post-Analysis and Refinement stage, which can prune/split and/or merge clusters to improve cluster effectiveness. Essentially, this stage can setup the clustering algorithm for multiple passes, which is implied by the cycle shown above the

Documents: Shown: 1-50 of 998	
<p>1-Detail title: WoYaa! - African Countries description: Online Algeria, Angola, Benin, Botswana, Burkina Faso, Burundi, Cameroon, Cape Verde, Egypt, Equatorial Guinea, Eritrea, Ethiopia, Gabon, Gambia, Ghana, Guinea, Guinea Bissau, Madagascar, Malawi, Mali, Mauritania, Mauritius, Morocco, Mozambique, Namibia, Niger, Ni... Sierra Leone, Somalia, South Africa, Sudan, Swaziland, Tanzania, Togo, Tunisia, Uganda, W... Zambia, Zimbabwe, Cote d'Ivoire url: http://www.worona.com/80/Tree/Regional/Countries/</p> <hr/> <p>2-Detail title: Maps of Africa description: Maps of Africa. The Perry-Castaneda Library Map Collection holds more than 230,000 maps url: http://www.iss.usc.edu/Internet/Links/Africa.html</p> <hr/> <p>3-Detail title: Agricultural Engineering Societies description: Agricultural Engineering Societies Including selected related professional Engineering and Agr... Henry Professor of Agricultural Engineering... url: http://arsae.cyp.80th.oleth.edu/arsae.html</p> <hr/> <p>4-Detail title: http://education.leds.ac.uk:801-education/cei-tech.htm description: The Centre for Studies in Science and Mathematics Education has a long history of involvement... international journal Studies in Science Education is edited... url: http://education.leds.ac.uk:801-education/cei-tech.htm</p>	<p>Documents (clustered results)</p> <p>Group (11): british council, management education, british studies, economics, english language, arts, technology documentary exhibitions, law, africa, training development, commonwealth science, arts virtual campus</p> <p>Detail [score 100.00] The British Council - Cameroon Detail [score 98.02] The British Council - Eritrea Detail [score 97.53] The British Council - Botswana Detail [score 97.31] The British Council - Namibia Detail [score 97.27] The British Council - Senegal Detail [score 97.26] The British Council - Algeria Detail [score 97.24] The British Council - Lesotho Detail [score 97.04] no subject available Detail [score 96.17] The British Council - Morocco Detail [score 94.15] The British Council - Madagascar Detail [score 44.12] British Council - CCEM Parallel Convention - Exhibition</p> <hr/> <p>Group (10): africa online, chad geography, economy database, news information, flora, schedules events, education markets, information tickets, equator map references, texas land boundaries, site map, guinea bissau</p> <p>Detail [score 100.00] Welcome to Niger! Detail [score 78.92] Welcome to Mali! Detail [score 78.11] Welcome to Nigeria! Detail [score 74.74] Welcome to Chad! Detail [score 72.94] Welcome to Liberia! Detail [score 71.24] Welcome to Mauritania! Detail [score 60.40] Welcome to Benin! Detail [score 58.06] Welcome to Guinea! Detail [score 54.48] Chad Detail [score 40.52] Africa Online: News & Information</p>

Figure 2: Document and Clustered Retrieval Result Excerpts

two adjacent processing steps in Fig. 3. Finally, the Summary Generator module is invoked to generate enhanced cluster summaries from the list of top discriminating words output by the core clustering algorithm. The actual clustering algorithm is parameterized to allow for modification of the similarity thresholds used to determine document assignment into topical classes, the rules for managing multi-class assignments, and the method used to compute centroids dynamically as the cluster changes.

Previously we experimented with natural language processing (NLP) techniques for generating cluster summaries. Our former NLP approach would first perform a part-of-speech tagging (Brill 93) on the original documents, and would then extract noun phrases that contain the top discriminating keywords. This approach worked fine in a non-interactive batch mode, although the performance was not adequate for interactive "real time" processing. Due to this limitation, our current summary generation module is based on an approach that uses stop words and document-specific parsing heuristics as a basis for bounding and extracting phrases that contain the keywords. We have found that the greatly decreased processing time of this approach out-ways the marginally better phrases generated via the more advanced NLP technique.

3 Prototype System

The CELF prototype exercises the subsystems described in section 2. Section 3 describes in detail each subsystem of the prototype. It should be noted, the scope of the pilot is to build an electronic library of documents focusing on the continent of Africa. Each of the countries in Africa is associated with topics derived from sources including the World Fact Book (Fact Book). There are approximately 750 topics used for the pilot, organized into a hierarchy that is up to four levels deep.

3.1 Automatic Data Collection

The Harvest RDS, is currently scheduled bi-weekly to collect data from one hundred Africa related Web sites. The Harvest mechanism is configured to collect all meta-data from linked pages excluding image pages at each of the Africa sites. The sites pointed to by the Harvest robot were identified as housing enough significant content to justify collecting each entire site.

The Agent RDS is scheduled weekly to perform its collection. It currently only utilizes top-level profiles for its canned queries. Because collected documents are later automatically categorized by the categorization subsystem, we only need to utilize the top two levels of profiles. Deeper topical profiles (used by the categorization module) are more specific and typically

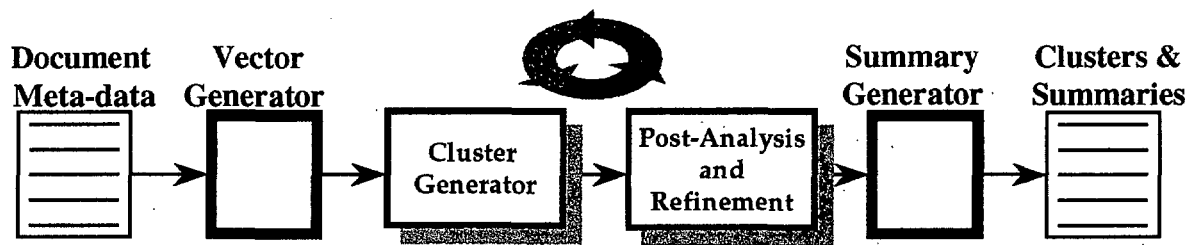


Figure 3: Document Clustering Stages

do not result in higher document recall for the search agent.

3.2 Manual Data Collection

The manual collection subsystem provides the mechanisms necessary for data contributors to publish information into the Library Catalog. The components of the publishing subsystem include the contributor browse/search interface, the information submission form and the review subsystem.

Using the enhanced browse interface, an information contributor can simultaneously browse his or her assigned topical areas in both the Library Catalog and the Sources Catalog. This feature allows the data miner to locate new information in the Sources Catalog and determine the need for this information in the Library Catalog. From the enhanced browse screen, the user has direct access to the information submission form. Clicking the submission form link automatically fills all available meta-data fields with the existing data for the record being submitted. Using the submission form, the contributor has the ability to add additional meta-data to the record as well as verify the correctness of the information before confirming its inclusion in the catalog.

Documents contributed to the Manual RDS require review by a group leader. Each member of the contribution team has been placed in a group and assigned a set of topics to cover. Each group has a leader who is responsible for reviewing the entries submitted by his or her group members. Ideally, the group members and leaders possess expertise in the topical areas assigned to them.

Fig. 4 displays screen shots which depict the flow of data manually submitted to the system. The right screen is the Browse interface. Notice the dual frame display of both the Library Catalog (in the left frame) and the Sources Catalog (in the right frame). Also note the "contribute to library" link used to generate a pre-filled submission form. The left screen displays the document submission form. The submission form allows the contributor to provide meta-data describing the document being submitted. Each field is editable, with convenient pick lists provided when appropriate.

The review interface is very similar in appearance to the submission form. Both interfaces were designed to provide easy access to all available meta-data for each document submitted. In the review mode, the reviewer has the ability to add additional comments or meta-data before accepting the document. It should also be noted, a document contributor can use the review interface to look over and modify his or her own documents prior to supervisor approval.

3.3 Search and Browse

The browse and search interfaces afford both end-users and document contributors easy access to documents stored in the Sources and the Library Catalogs. The customized search and browse interfaces use an HTML frame based implementation. The frame design was chosen for its ability to maximize screen real estate and present the user a consistent "toolbar" look and feel.

In the top frame of the browse interface (Fig. 4 right), the current topics for the level are displayed. Clicking on a linked topic moves the user down through the hierarchy to the next lowest level. Topics at the leaf level are not linked. At each topical level the user can perform a browse query to retrieve documents that have been categorized in the current topic.

The user also has the option to request the return of all documents from the current topic and all its sub-topics. The powerful browse interface also allows the inclusion of a query to augment the browse retrieval and return only documents from the topic matching the specified query. To the right of each topic in the top browse frame is a set of operation buttons. The options made available depend on the user authentication level and the current mode of operation. In the mode displayed in Fig. 4, the user has the options to retrieve a description of the topic and submit a new document for the topic. These options are represented by the "?" and the "C," respectively. The "?" is available to all users regardless of operation mode.

The bottom frame of the browse interface (Fig. 4 right) displays the results of the browse query. In this example, results from both repositories were retrieved. Again, the screen is divided into frames allow-

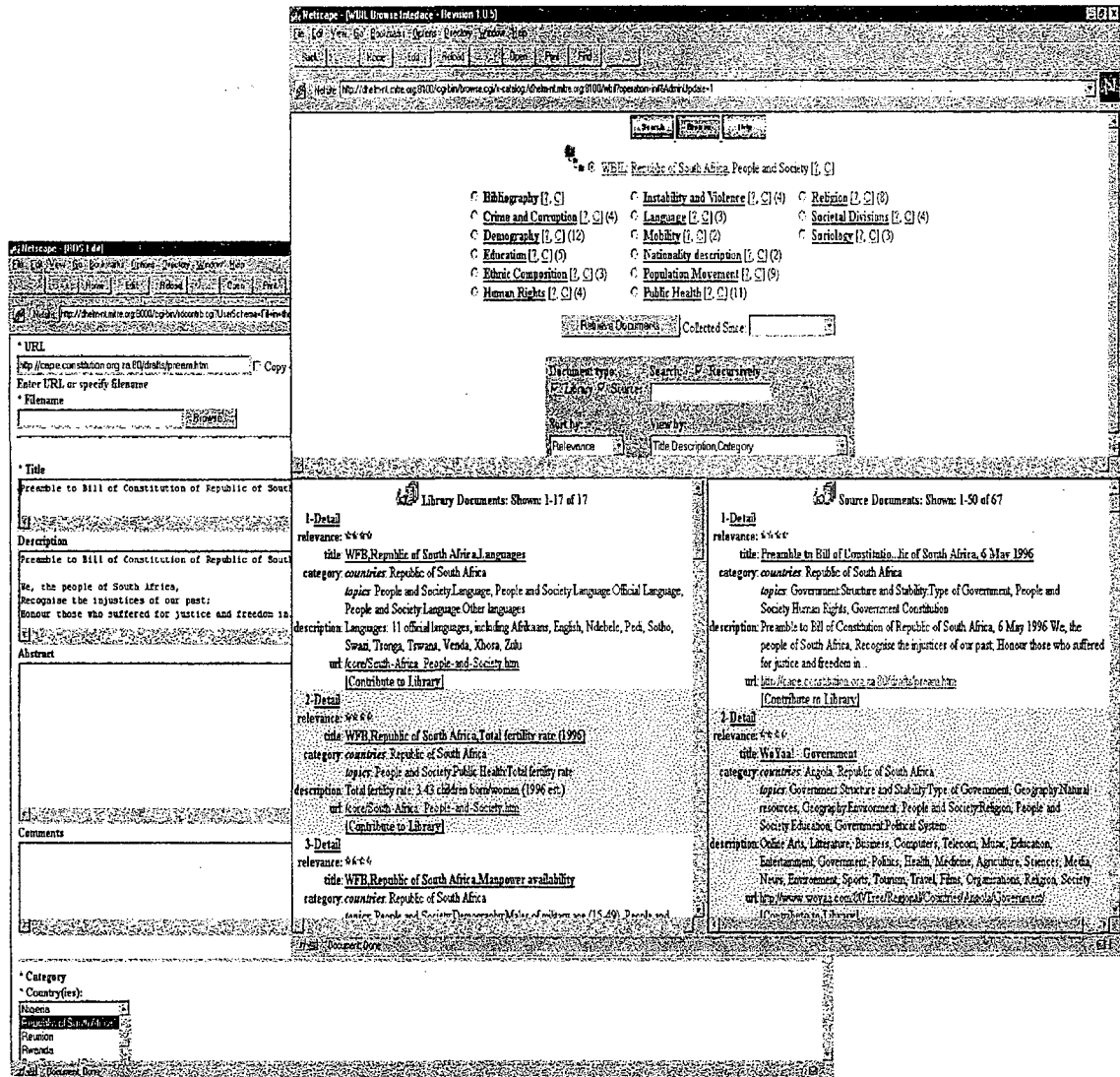


Figure 4: Document Browse and Contribution Screens

ing the user to preserve context for each search during scrolling and paging operations. The data mining process, on the part of the document contributor, requires verifying that the information he or she is about to submit does not already exist in the Library Catalog. End users find the ability to search both catalogs not only effective but also necessary. After unsuccessfully mining the Library Catalog for specific information, users often discover relevant data in the Sources Catalog.

The search interface, and the frames associated with it, is much the same as the browse interface. The search interface allows advanced users to specify complex queries, which generate focused search results. The significance of the search interface is its ability to retrieve documents across the entire database independent of topical assignments.

In the top frame of both the browse and search interfaces, users can specify various display options. These options include, "Sort By" and "View By." The "Sort By" option allows the user to specify how search results are to be ordered. Orderings include, "Relevance," "Alphabetical" by title, and "Newest." The "View By" option enables the user to select the meta-data fields to be displayed on the result screen. Display options range from condensed to verbose views including all meta-data fields.

The document viewing screens are depicted in Fig. 5. After performing a retrieval in either the browse or search interfaces, selected meta-data for each returned document are returned. As part of each document's result list is a "Detail" link located at the start of each document. The Detail link allows the user to retrieve all available meta-data for a document without

re-searching (Fig. 5 right). The result of accessing this link is displayed in a new copy of the browser, allowing the user to maintain context of the search results.

This link is essential for users who want to display a compact search result with limited meta-data, via the "View By" display option, and still access all of a document's meta-data without re-searching. Clicking the "Detail" link displays all meta-data including the URL for the document. The user can also view the actual document (Fig. 5 left) by clicking on the URL link.

4 Conclusion and Future Directions

CELf is currently being utilized in a pilot project that includes a group of users who are building a specialized collection focussing on the continent of Africa. The project will run for a number of months and will culminate with a report detailing "lessons learned" as a basis for building a more effective system in the future.

The CELf prototype includes a number of notable features. Among the most interesting are the two-repository approach that is comprised of the Library and Sources Catalogs, the automatic meta-data collection and categorization methods, and post-retrieval tools.

Preliminary feedback received during the pilot project has shown that the inclusion of both a manually generated and automatically generated collection proves to be quite useful. Typically, the information content in a manually generated repository can be of higher quality than one that is automatically generated; however, an automatically generated collection can include many more information items. This is primarily due to the fact that automatic techniques enable a larger volume of information to be harvested and categorized. The individual documents contained therein, however, may not necessarily be as relevant to the topical categories to which they are assigned due to recall and precision anomalies (Salton et. al. 83) inherent with utilizing automatic meta-data extraction and categorization techniques. The CELf techniques available for enabling document contributors to easily add documents from the sources repository into the core library repository have also proved to be quite effective. In addition, utilizing the topical profile driven search agent has been a very effective method to improve the relevancy of documents collected by robots for the sources repository, as compared with the technique that harvests documents from entire Web sites beginning with manually provided top-level URL's.

Planned future enhancements to CELf include support for more effective document categorization. This will include developing tools to support categoriza-

tion performance evaluation. We have found the need for tools enabling a knowledge engineer to determine categorization performance across entire CELf taxonomies. When dealing with complex taxonomies consisting of many topics organized into hierarchies several levels deep, we have found that effective tools are needed to enable one to visualize where there are problem areas. Problems can include incorrectly classified documents, insufficient numbers of classified documents for certain topics, or too much overlap between categorization results of related topics. Other enhancements will include additional tools to support effective post retrieval analysis and document understanding. This can include document summarization and visualization services that expand upon the currently integrated clustering tools as a basis for providing better document understanding of retrieved results. We also plan to extend the agent-based collection subsystem to access and collect data from heterogeneous systems. Currently only information available via URL's can be collected. This collection subsystem may be extended to access diverse repositories such as database systems.

References

- [1] Bowman C. M., Danzig P. B., Hardy D. R., Manber U., and Schwartz M. F. (1994) Harvest: A scalable, customizable discovery and access system. *Technical Report CU-CS-732-94*, Department of Computer Science, University of Colorado, Boulder, Colorado.
- [2] Brill E. (1993) A corpus-based approach to language learning. *Ph.D. thesis, Department of Computer and Information Science*, University of Pennsylvania.
- [3] Catalog Server. <http://home.netscape.com/comprod/server.central/product/catalog/datasheet.html>.
- [4] Fact Book. <http://www.odci.gov/cia/publications/nsolo/wfb-all.htm>.
- [5] Genesereth R. R. and Ketchpel S. P. (1994) Software Agents. *Communications of the ACM*, 37(7), pages, 48-53.
- [6] Helm D. J., D'Amore R. J., and Yan P. (1996) MIDS: A Framework for Information Organization and Discovery. *Journal of Network and Computer Applications*, 19(4), pages 381-394, Academic Press, November 1996.
- [7] Jain A. K. and Dubes R. C. (1988) *Algorithms for Clustering Data*. Prentice Hall.

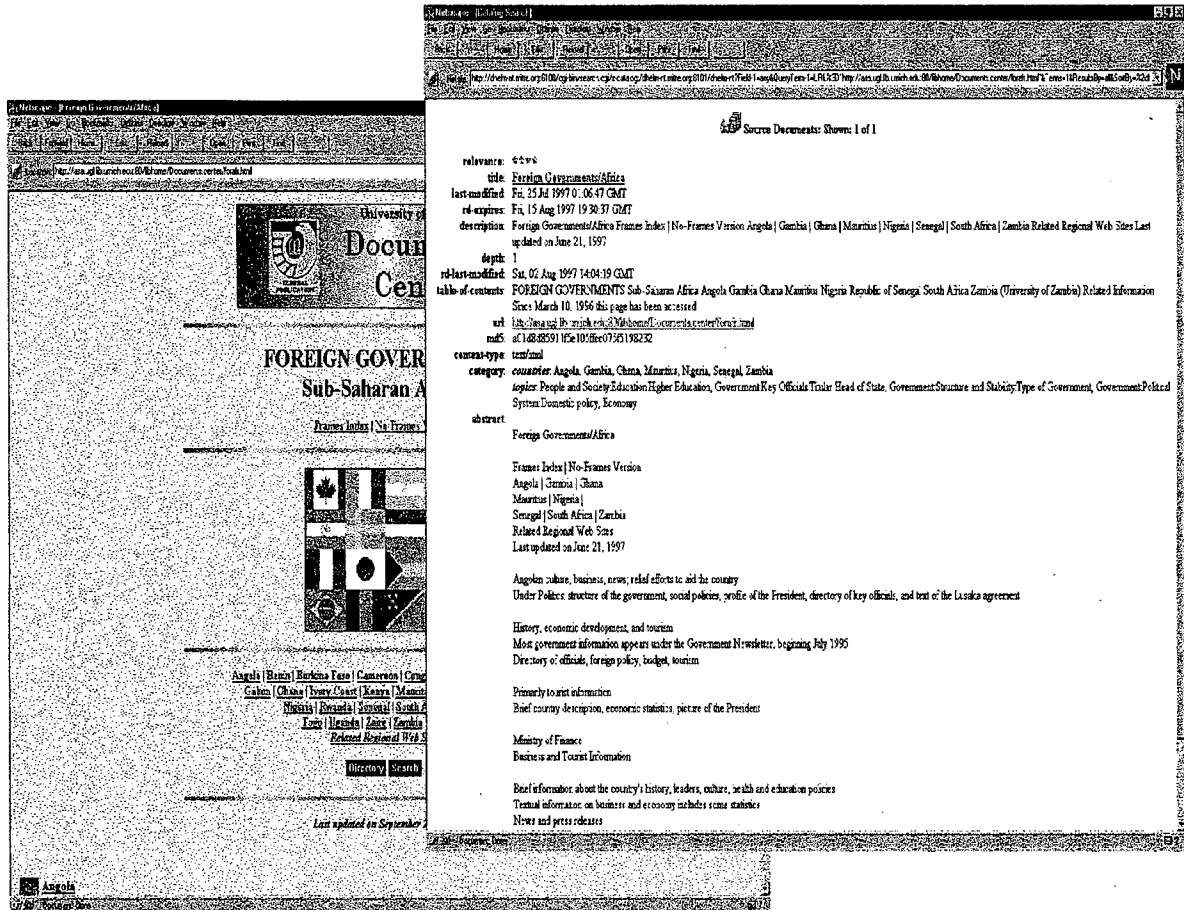


Figure 5: Document Display Screens

- [8] Levy D. M. and Marshall C. C. (1995) Going Digital: A Look at Assumptions Underlying Digital Libraries. *Communications of the ACM*, 38(4), pages, 77-84, April 1995.
- [9] Rao R., Pedersen J. O., Hearst M. A., Mackinlay J. D., Card S. K., Masinter L., Halvorsen P., and Robertson G. G. (1995) Rich Interaction in the Digital Library. *Communications of the ACM*, 38(4), pages, 29-39, April 1995.
- [10] Salton G. and McGill M. J. (1983) *Introduction to Modern Information Retrieval*. McGraw- Hill, Inc.
- [11] Verity. <http://www.verity.com>.
- [12] Yahoo. <http://www.yahoo.com>.

A Study in the Use of Parallel Programming Technologies in Computer Tomography

Eugene G. Sukhov
 Institute of Control Sciences,
 Russian Academy of Sciences,
 Profsoyuznaya 65, Moscow, Russia
 Phone: +095 334 79 51, E-mail: sukhov@ipu.rssi.ru

Keywords: parallel programming, computer tomography, parallel computations, O-O programming

Edited by: Branko Souček

Received: December 12, 1992

Revised: February 9, 1993

Accepted: March 1, 1993

In this paper we consider a parallel implementation of linear algebra algorithms for computer tomography and study two approaches for this purpose. The first concerns parallel Cholesky algorithm based on a block decomposition procedure. The second deals with parallel iteration procedure in terms of so called sections, where each section is a subset of 2D array components. Formulations given are good suited to object-oriented programming. The FORTRAN-like implementation is considered.

1 Introduction

In this paper we discuss a parallel implementation of image reconstruction algorithms for computer tomography. It is well-known that the image reconstruction demands a large number of computation, in some cases in real time, which is beyond the capability of present-day computer systems even when relatively modest image matrices are involved. The specialized systems can be based on fast analogue devices [2] or on transputer arrays [1, 3]. But using of general-purpose multiprocessor systems is also attractive for tomography computations [4]. In this paper we don't concern the emission tomography presented in [1, 4] and consider the image can be obtained through an ultrasonic diffraction procedure or through X-raying procedure. In the first case it is essentially to solve an inverse diffraction problem but numerical solution leads to a large scale linear algebraic system with a dense symmetric matrix. The second case gives a variety of ways to create the image reconstruction algorithm. Each algorithm represents the iterative versions of inverse Radon transform [5]. Our objectives are:

(1) to make direct numerical procedure for solution symmetric linear algebraic systems arised in computer tomography, which can be applicable to a wide range multiprocessor systems; (2) to make iterative parallel procedures based on a ray approach with the same applicability and (3) to estimate the efficiency of object-oriented (O-O) programming for problems under study.

We shall study Cholesky decomposition for the first point and the iterative "back-projection equalization" algorithm for the second one. Algorithms pro-

posed are good for Single-Instruction, Multiple-Data (SIMD) systems, beside implementation for Multiple-Instruction, Multiple-Data (MIMD) is possible without trouble. The examples below are given in convenient FORTRAN-like notation with O-O insertions. The reason is that the O-O programming facilities are available in the latest FORTRAN versions.

2 Cell Methods

The cell methods are very convenient for solving large-scale matrix problems in linear algebra. A variety of the cell methods and base operations proposed in paper [6, 7] good fits to parallel processing. It was shown that the cell decomposition can be evaluated in pyramidal recursive mode. We start the process from the top level down to the bottom level, where the base matrix operations are actually performed.

We developed a set of basic operations with matrices and vectors that plays a similar part as PBLAS for ScaLAPACK library [8] for non-parallel matrix computations. Our goal however was to create procedures for recursive implementation. The base is self-similarity of numerical procedures at each level of the pyramid. It was demonstrated that the SIMD system model of W processors ring-connected is a good instrument for the basic operations above. Another model is $W \times W$ - processor matrix with the ring-connected rows and columns.

The cell processors can form numerical modules for a large scale matrix parallel processing. Multycluster systems of this sort may also include a sharable memory modules of different types, and the multilevel de-

composition can be easily adopted. Parallelism available at the base level is intracellular. It can be hidden from the application designer by software facilities. Parallelism above the base level is intercellular. This kind of parallelism, if present, is explicitly affected by the computer structure and is not discussed in this paper.

3 Cholesky Scheme

The linear algebraic system under study can be written as

$$(1) \quad Ax = b,$$

where A is a real, symmetric and positively defined matrix of order N . The standard scheme for solution (1) is Cholesky decomposition :

$$(2) \quad A = LL^T,$$

followed by solving two arised systems

$$(3) \quad Ly = b, \quad L^T x = y,$$

where L is a lower triangular matrix, the upper index T denotes transpose.

LL^T -decomposition Let the matrix A be splitted in n^2 square cells, each cell is of order N/n . LL^T -decomposition algorithm (2) can be written in the next cellular form:

- 1) $B_{11} = L_{11}L_{11}^T;$
- 2) $B_{1j} \leftarrow B_{1j}(L_{11}^T)^{-1}, \quad j = 2, 3, \dots, n;$
- 3) For $i = 2, 3, \dots, n$:
 - 3.1) $A_{ik} \leftarrow A_{ik} - B_{jk}B_{ji}^T,$
 $j = 1, 2, \dots, i-1, \quad k = i, i+1, \dots, n;$
 - 3.2) $A_{ii} = L_{ii}^T, \quad B_{ii} \leftarrow A_{ii},$
 - 3.3) $B_{ik} \leftarrow A_{ik}(L_{ii}^T)^{-1},$
 $k = i+1, i+2, \dots, n.$

The expression $B = LL^T$ denotes LL^T -decomposition of cell B with preserving previous contents. An arrow instead of equality sign denotes rewriting previous contents. As the result matrix B has upper cellular-triangular form being the L^T -factor in Cholesky decomposition.

On the other hand, any cell in steps 1-3 can be splitted in cells of lower level. The operations can be expressed in terms of the same operations in lower level. At the lowest level operands are numbers, not matrices. The matrix LL^T -decomposition corresponds to the square rooting, the matrix AB^{-1} - multiplication corresponds to the numerical division, and so on. In such manner the algorithm above can be presented in recursive pyramidal form. An example of two - level decomposition is shown in Fig.1.

LL^T - decomposition is an algebraic operation, hence the numbers of arithmetic operations in their scalar and cellular versions are the same [7].

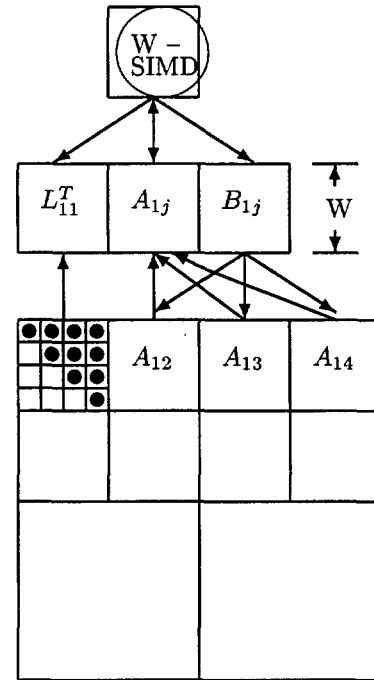


Figure 1: Parallel cell matrix processing (step 2 of LL^T -decomposition is depicted).

Triangular systems solution The first lower cellular triangular system of (3), $Ly = b$, where y and b are vectors of length N , can be decomposed in cells and subvectors in accordance with the decomposition of original matrix A . In these terms the solution algorithm can be written as follows:

- 1) $y_n \leftarrow L_{nn}^{-1}b_n,$
- 2) $y_i = b_i - L_{ij}x_j,$
 $i = n-1, \dots, 1, \quad j = i+1, \dots, n,$
- 3) $y_i \leftarrow L_{ii}^{-1}y_i$

Here we have the operations of cell-subvector multiplication and triangular system solution at the lower level. The recursive pyramidal decomposition above is also good working in the triangular case. The second system of (3) can be solved by using the same procedure.

4 Object-Oriented Programming

Object-oriented programming is now provided for parallel processing. For example, POOMA system [9] is an object-oriented framework for hydrodynamics and plasma simulation. It represents a library of C++ classes, such as field and particle classes designed for numerical methods. Computer tomography as well is a good field for application object technology.

The pyramidal cell-decomposition procedures above possess a lot of properties to implement object-oriented programming. These are:

1) Modularity - the problem solution can be expressed in terms of small number of the basic procedures and data types.

2) Hierarchy - the problem solution is level-composable.

3) Recursivity - all the operations at each level can be expressed in terms of the same operations at the level below, thus the operations are self-expressed.

In this approach the parallelism at each level can be implemented without regard to the other levels. We declare the main data types CELL and SWORD (superword, i.e. subvector) in FORTRAN-like notation as follows:

```

TYPE CELL
  ARRAY (N,N) OF CELL
  INTEGER LEVEL,ORDER
END TYPE
TYPE SWORD
  ARRAY (N) OF SWORD
  INTEGER LEVEL,ORDER
END TYPE

```

The procedures (methods) can be outlined in next unified form:

```

RECURSIVE SUBROUTINE S1
PUBLIC K
IF LEVEL=K THEN CALL S1
ELSE
LEVEL=LEVEL+1
CALL RECURSIVE SUBROUTINE S2
END

```

The procedure S2 represents implementation of the procedure S1 in terms of below level procedures. At the top level an initial matrix is decomposed in accordance with available system configuration. Let K levels in our case be:

$$N = n_1 n_2 \dots n_k,$$

hence n_i is the cell order at i -th level. For the i -th level procedures the value of n_{i-1} enabling decomposition is also known. The total number of levels K is the global value.

If $N = n_2^k$, it is possible to construct a fast recursive algorithm of Strassen type [7].

Example 1. Let host-computer has a memory size h_1 words and a SIMD-accelerator using W processors, each with h_W words of memory. A sharable global memory contains h_2 words. A matrix of order N is stored on the host disc. Assume the condition

$$N^2 \gg h_1 \gg h_2 \gg h_W \gg W$$

is fulfilled.

In each level we have to keep no less than three cells in the memory, hence the 4-level decomposition can be performed, and

$$n_1 \approx \sqrt{h_1/3}, n_2 \approx \sqrt{h_2/3}, n_3 \approx \sqrt{h_W/3}, n_4 = W$$

If $h_W = W$, then $n_3 = W$, and the fourth level is absent.

Example 2. An array of n^2 disk units keeps the "large" $n \times n$ cells of initial matrix. If the host computer has v_2 size of memory, we can set:

$$n_1 = N/n, \quad n_2 \approx \sqrt{v_2/3}.$$

5 Iterative Reconstruction Scheme

In the previous section we described the method to provide dense linear systems solution. Such kind of solvers are important for inverse scattering problems in diffraction computer tomography. But on the other hand, matrices arising in X-ray and gamma-ray imaging are sparse. That is the reason for using iterative algorithms [5]. Each iterative scheme is the version of numerical inverse Radon transformation. Input data in this case is the set of ray sums on the beams passed along the different ray traces.

In this paper we use a simple model of radiation attenuation based on plane parallel beam scanning. This attenuation may be due to Compton or combination scattering and photoelectric absorption. We think the image under study as a square matrix of pixels where the two-variable discretized function should be reconstructed.

Suppose that this function domain is covered by the grid included N^2 square pixels. Each pixel is located by index pair (i, j) in general matrix style and the function value attached to it is denoted by u_{ij} .

We think this function as pixel absorptance. In our case the radiation attenuation is exponential:

$$I/I_0 = \exp(-\alpha \Delta X),$$

where I_0 is an initial radiation intensity and I is the radiation intensity after passing through the layer with the thickness of ΔX , α is an attenuation factor. So the total absorptance can be conveniently expressed by the value of $-\ln(I/I_0)$. In that case it's possible to sum the pixel absorptances along the ray path. On the other hand it's necessary to take into account that each pixel is passed by the ray in its own manner. For this purpose we introduce weight factors g_{ij} . The maximum weight corresponds to the ray path through a geometric center of pixel. A weighted ray sum S can be written as

$$(4) \quad S = \sum_{i,j \in l} u_{ij} g_{ij},$$

where the summation is applied to pixels along the ray path l .

In practice this model conforms to plane-parallel monochromatic raying the object under investigation

and measuring of transferred radiation intensity. In that case total attenuation can be easily determined. The model above may be adopted to a variety of the scattering and absorption processes.

For image reconstruction a set of ray sums should be accumulated. Then the u_{ij} can be determined as the solution of the linear system (4), where the weight factors g_{ij} are known. Now the system (4) can be conveniently written in standard form

$$(5) \quad Gu = S$$

. Let image be stored as a matrix of the order $\nu = \sqrt{n}$, where n is the total number of pixels. The right side vector S is the set of projections, i.e. the measured values of total attenuation. Below we shall discuss the "back-projection equalization algorithm" as a good illustration of parallel approach to iterative solution (5).

6 Sections

Let the image be a square matrix of order N . Suppose the X-ray beam is directed to make angle α with the vertical ($(|\alpha| < |pi/2)$). In such geometry the beam can not enter the image through the bottom and exit through the top. We define a section $C(i, j, \alpha)$ as a sequence of triads:

$$C(i_1, j_1, \alpha) = (i_k, j_k, g_k), \quad k = 1, 2, \dots, M, \quad (6)$$

where i_k, j_k are the indices of pixels passed by the ray during the propagation, g_k is the weight factor associated with k -th pixel. If the beam is uniform and its step is equal to the mesh size, then the section is uniquely determined by the first passed pixel indices i_1, j_1 . The first pixel, and the last one i_m, j_m always are the boundary pixels of image, as it is illustrated in Fig.2.

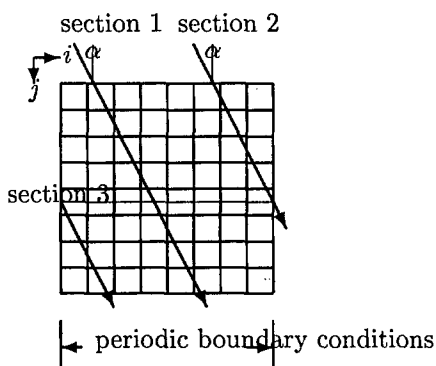


Figure 2: Section example on 8 x 8 image.

Any tracing algorithm may be applied to calculate index set for section in (6). These calculations are the pre-processing stage of reconstruction .

In O-O style we define two data types: PIXEL and SECTION

```

TYPE PIXEL
INTEGER I,J ! INDEX
REAL G ! WEIGHT
END TYPE
TYPE SECTION (I,J,ALPHA)
INTEGER I,J
REAL ALPHA
PRIVATE INTEGER M
PIXEL A(M) ! A=PIXEL ARRAY
END TYPE

```

The operations (methods) to be used in the sections are the next:

1. Initialization (making), i.e. tracing and weighting the separate ray-path. This may be done only once for all images in work. But for good imaging it is necessary to make a set of sections directed by the variety of angles. The initialization is headed as

```
SUBROUTINE INIT (I,J,ALPHA)
```

2. Opening, i.e. activation of the section:

```
SUBROUTINE OPEN(I,J,ALPHA)
```

All the operations below are specific to earlier opened sections.

3. Reading i.e. extracting and weighting matrix elements in accordance with the index set. Array V is used for data storage

```
SUBROUTINE READ(V)
```

```
DIMENSION V(M)
```

4. Writing, when the contents of array V is stored in accordance with the index set:

```
SUBROUTINE WRITE(V)
```

```
DIMENSION V(M)
```

5. Closing the section:

```
SUBROUTINE CLOSE(I,J,ALPHA).
```

Suppose that we have collected all the projection data arrays S_{ij} for all the sections in work. If the sections are initialized, we can write the back-projection equalization algorithm in the next form:

```

ITERATION: DO I=1,NU
DO J=1,NU
CALL OPEN(1,J,ALPHA(I))
CALL READ(V)
D(J)=S(I,J)-SUM(V)
DLT=D(J)/LENGTH()
V=V+DLT
CALL WRITE(V)
CALL CLOSE(1,J,ALPHA(I))
END DO
END DO
IF MAX(ABS(DLT)).LE.EPS THEN
EXIT
ELSE REPEAT ITERATION

```

Here the variable $NU = \nu$ is the order of pixel matrix and the number of direction angles, the small positive number EPS is the threshold value. The function SUM(V) sums all components of the array V, which length is determined by LENGTH() .

7 The Parallelism Outlines

The section-based scheme above presents inherent parallelism due to ability of the sections to be processed in parallel in SIMD-style. For improving parallelism we introduce an extra operation of section addition. This addition joins index sets and weights for the sections in operation. If the image matrix is processed on a parallel system, including NU processors, the addition provides length equalization for all the sections. This is important for load-balancing of parallel processors.

For length equalizing it's necessary to place the periodic boundary conditions on the left and right sides of image. These impose that all j -index increments in the initialization procedure should be evaluated modulo ν . In that case the section terminated on any side of image is added to its extension on the opposite side. As the result all the sections started at the top of the image with the same angle α have the same length and terminate at the bottom of the image.

Sections 2 and 3 in Fig.2 are added to produce the sum-section with the same length as section 1.

For section processing we can use model SIMD-system with ring connected processors. Such configuration is good suited to periodic boundary conditions. The total memory should be sufficiently large to store the whole image and the supplements.

On the other hand, the section addition is necessary to implement the cell methods in imaging. If the processing image is splitted into cells any full section is the sum of intracellular sections. After current cell processing the partial ray sums are stored. Values stored are the initial conditions for the ray sums in adjacent cells. But the implementation of cell sections is not quite elegant for the lack of recurrence.

If the image cell of order W is processed in parallel, then $O(W)$ parallel $ax + b$ -operations per one iteration are demanded. Numerical experiments showed sufficiently good convergence for $W = 32$.

8 Conclusion

The computer tomography problems are fruitful area for parallel computations implementing. The linear algebra cells method for large systems equations are the main key for getting solution. The methods discussed have a natural hierarchy (also recurcivity in algebraical case) and are good suited for massively parallel computing systems. The explicit hardware dependence can be located only at the lowest level, where the base numerical algorithms are applied. That is good for portability.

The O-O programming is an adequate approach to the problems under study. It is based on the same concepts that found the decomposition of large matrix problem for parallel computations.

References

- [1] K.A.Girodias, H.H.Barrett, and R.L.Shoemaker, "Parallel simulated annealing for emission tomography", *Phys. Med. Biol.*, Vol.36, pp.921-938, July 1991.
- [2] A.F.Gmitro, V.Tresp and G.R. Gindi, "Video-graphic tomography- Part I: Reconstruction with parallel-beam projection data", *IEEE Trans. Medical Imaging*, vol.9, pp.366-375, Apr. 1990.
- [3] F. Wiegand and B. S. Hoyle, "Simulations for parallel processing of ultrasound reflection-mode tomography with applications to two-phase flow measurement", *IEEE Trans. Ultrasonics, Ferroelectrics and Frequency Control*, Vol.36, pp.652-660, Nov. 1989.
- [4] M.I.Miller and B.Roysam, "Bayesian image reconstruction for emission tomography incorporating Good's roughness prior on massively parallel processors", *Proc. Natl. Acad. Sci. USA*, Vol.88, pp.3223-3227, Apr. 1991.
- [5] V.Picalov, N.G.Preobrazhenski, Reconstructive tomography in gas dynamics and plasma physics, Novosibirsk, Nauka. 1987.
- [6] E.G.Sukhov, Cell parallel methods for linear algebraic systems solution, *Avtomatica i Telemekhanika*, 1988, No 9. pp.139-143.
- [7] E.G.Sukhov, Matrix calculations on SIMD multi-processors, *Programmirovaniye*, 1981, No 4 pp.40-49.
- [8] J.Choi, J.Dongarra, and D.Walker, PB-BLAS: A Set of Parallel Block Basic Linear Algebra Sub-routines, *Concurrency: Practice and Experience*, 8 (1996), pp. 517-535.
- [9] S.Atlas et al., POOMA: A high performance distributed simulation environment for scientific applications, *Supercomputing'95 Conference*, (1995).

Topological Informational Spaces

Anton P. Železnikar

An Active Member of the New York Academy of Sciences

Volaričeva ulica 8

SI-1111 Ljubljana, Slovenia

anton.p.zeleznikar@ijs.si and at home s51em@lea.hamradio.si

http://lea.hamradio.si/~s51em/

Keywords: basis, closed system, connectedness, covering, discrete space, exterior, indiscrete space, informational space: operand formula vector, vector distributiveness (orthogonality), metrics (meaning); interior, linkage, neighborhood, open system, subbasis; system of informational formulas, operands and basic transition formulas; topology of systems

Edited by: Vladimir A. Fomichov

Received: April 28, 1998

Revised: August 18, 1998

Accepted: August 25, 1998

A system, Φ , of different informational formulas, φ , can possess various topological structures, \mathcal{D} . By this, topological informational spaces of the form (Φ, \mathcal{D}) can be constructed and the question arises: How can the topological structures be introduced reasonably for concrete systems of informational formulas?

A topology causes certain other concepts, e.g., those concerning closed topology, connectedness, continuum, interior, exterior, neighborhood, basis, subbasis, metric, space, etc. of systems, and especially the concept of meaning as a kind of the informational accumulation point. The paper treats topologies of three types of informational formula systems: Φ_φ , $\Phi_{\xi=\eta}$, and Φ_ξ . An example of bidirectional consciousness shell is presented enabling a complex engine modeling.

1 Introduction

A basic problem of topology¹ is to define a general space. By topology a mathematical concept (structure, branch) is meant² giving sense to various intuitive notions. Topological notions can be innovatively extended into realm of the informational, realizing one of the significant features of the so-called *informational space*. Such a space can be determined also from other points of view concerning, for instance, the distributivity of informational entities (operands)—which can proceed into different concepts of a vector space. In general, a more complete theory of informational space would need concepts of informational subtheories, such of concerning informational topological space, informational vector space, and informational graph theory.

Another, mathematically grounded view to the problem of graph is the so-called topological graph theory [14]. The primitive objective of this theory is to draw a graph on a surface, so that no two edges (graph

arrows representing informational operators) cross, an intuitive geometric problem that can be solved by specifying symmetries or combinatorial side-conditions (surface graph-embedding). Although potentially interesting for the informational graph theory, this kind of problem is not in the focus of informational graph investigation. Informationally, graph is merely a presentation of formula or formula system potentiality concerning the setting of the parenthesis pairs (parenthesizing³) in a formula or formula system.

Introducing the topology on (over) a system of informational formulas means a challenge of logic—both the informational and the philosophical one—which comes close to some known metamathematical problems [30]. Just imagine a topology in the realm of mathematical axiomatism where for a set of axioms a topology of axiomatic statements (e.g., true formulas) is constructed. Although topology is a general mathematical principle in the realm of the set and space theory, it seems nonobviously to take it as a set with topology on the set of formulas.⁴

In mathematics, topology may be considered as an

¹This paper is a private author's work and no part of it may be used, reproduced or translated in any manner whatsoever without written permission except in the case of brief quotations embodied in critical articles.

²Otherwise, topology is a science of position and relation of bodies in space. This paper concerns at least the following topological topics: point system (set) topology (general topology), metric space (e.g., meaning topology), and graph topology.

³Parenthesizing (in German, Einklammerung) has also the philosophical meaning in phenomenology, for instance, in Husserl [17].

⁴The author believes too that such an idea reaches beyond the conventional horizon of a mathematician. However, he believes that the following discussion will show the appropriateness of such a trait.

abstract study of the limit-point concept [16]. Which factors could dictate the introduction of a topology for a given system of informational formulas? In informational cases, different kinds of reasonable topologies, corresponding intuitive ideas what an understanding, interpretation, conception, perception and meaning should be, are coming to the consciousness, that is, into the modeling foreground.

In set theory, the concept of a set (collection, class, family, system, aggregate) itself is undefined. Similar holds for an element x of the set X . The phrases like is in, belongs to, lies in, etc. are used. In informational theory, topology may be considered as an abstract study of the concept of meaning [32, 35] (concerning interpretation, understanding, conceptualism, consciousness, etc. of the informational). Here, meaning of something, of some formula or formula system, functions as an informational limit point, to which it is possible to proceed as near as possible by the additional meaning decomposition of something. The concept of a set is replaced by the concept of a system of informational formulas or/and informational formula systems. In this respect, similar notions to those in mathematics can be used, however, considering the informational character of entities (operands) and their relations (operators).

Introducing topological concepts in informational theory, the reader will get the opportunity to experience what happens if the informational concepts, priority described by the author (e.g., [31, 32, 33, 34, 35, 36, 37, 38], to mention some of the available sources) are thrown into the realm of a topological informational space. In this view, informational serialism, parallelism, circularism, spontaneism, gestaltism, transitism, organization, graphism, understanding, interpretation, meaning, and consciousness will appear under various topological possibilities, complementing the already previously presented informational properties, structure, and organization.

Mathematical topology, as presented for example in [7, 8, 16, 18, 19, 21, 24, 25], roots firmly in the mathematical set theory [5, 6, 20]. In informational theory, the set is replaced by the concept concerning a *system of informational formulas* (system, informational system or IS, in short). A system is—said roughly—a set of *informationally (operandly, through or by operands) connected informational formulas*. The question is, which are the substantial differences occurring between the mathematical and the informational conceptualism in concern to topological structure?

Elements of a mathematical set are elements determined by a logical expression (defining formula, relation, statement) and, for example, by notation of the form

$$X = \{x_1, x_2, \dots, x_m\}$$

which presents a concrete structure of the set by its elements.

In informational theory, instead of a set, there is a system of informational formulas being elements of the system. Formulas are active, emerging, changing, vanishing informational entities (by themselves) which can inform in a spontaneous and circular manner. What does not change is their informational markers distinguishing the entities. Notation of the form

$$\Phi = \begin{pmatrix} \varphi_1; \\ \varphi_2; \\ \vdots \\ \varphi_n \end{pmatrix}, \text{ where}$$

$$\varphi_1, \varphi_2, \dots, \varphi_m \rightsquigarrow \varphi_1, \varphi_2, \dots, \varphi_m$$

presents,⁵ in fact, only an instantaneous description of the parallel system of markers φ_i , by a vertical presentation, denoting concrete formulas (or formula systems), and being separated by semicolons. These are nothing else as a special sort of informational operators, e.g. \models , meaning the parallel informing of formulas of the system Φ . Also, there is a substantial difference between the symbols $=$ and \rightsquigarrow ; the second one is read as ‘mean(s)’ and denotes meaning and not the usual equality.

Another notions to be determined informationally are informational union and informational intersection of systems. It has to be stressed that formulas in a system “behave” in the similar manner as the elements in a set in respect to the union and intersection operation. Thus, the same operators can be used as in mathematics, without a substantial conceptual difference.

2 A Mathematical vs. Informational Dictionary

The presented dictionary should bring the mathematical feeling into the domain of informational theory. It certainly concerns the topological terms priority. The correspondence between set-theoretical and system-informational terms yields the following comparative table⁶:

MATHEMATICAL VS. INFORMATIONAL TOPOLOGY	
set X	system Φ : general formula system Φ_φ ; transition formula system $\Phi_{\xi \models \eta}$; and operand formula system Φ_ξ
set braces: $\{, \}$	system parentheses: $(,)$

⁵For the system-conditional formula, $\varphi_1, \varphi_2, \dots, \varphi_m \rightsquigarrow \varphi_1, \varphi_2, \dots, \varphi_m$ see the discussion in Sect. 3.1.

⁶In this dictionary, the ‘formula φ of Φ ’ ($\varphi \in \Phi$) is to distinguish from the informational speech convention where the meaning of ‘of’ represents an informational function, that is, the informational Being-of [29], e.g., $\varphi(\alpha)$ for ‘ φ of α ’.

vertical snake-form
 operand-occurrence
 braces [5, 6]
 empty set \emptyset
 set element x
 x is an element of X ,
 x belongs to X ,
 x is in X , $x \in X$;
 negation: $x \notin X$
 subset A
 A is a subset of X ,
 A is included in X ,
 $A \subset X$;
 negation: $A \not\subset X$
 powerset of set X ,
 $\mathfrak{P}(X)$ or 2^X
 union of sets, $A \cup B$

intersection of sets,
 $A \cap B$

difference of sets Y
 and X , $Y \setminus X$
 complement of set X ,
 $\complement X$
 complement of set X ,
 regarding set Y , $\complement_Y X$
 open set O
 topology O
 topological structure
 O
 topological space
 (X, O) , simply, X
 carrier X
 point $x \in X$

formula operand occurrence
 floor brackets: \lfloor, \rfloor

empty system \emptyset
 system formula φ
 φ is a formula of Φ ,
 φ belongs to Φ ,
 φ is in Φ , $\varphi \in \Phi$;
 negation: $\varphi \notin \Phi$
 subsystem Ψ
 Ψ is a subsystem of Φ ,
 Ψ is included in Φ ,
 $\Psi \subset \Phi$;
 negation: $\Psi \not\subset \Phi$
 powersystem concerning
 system Φ , $\mathfrak{P}[\Phi]$ or $\mathcal{P}[\Phi]$
 union of systems, $\Phi \cup \Psi$;
 or $(\Phi; \Psi)$, informing of
 both systems

intersection of systems,
 $\Phi \cap \Psi$ means, e.g.,
 $(\Phi; \Psi; \Phi \models \Psi; \Psi \models \Phi)$,
 parallel informing of all
 the systems' components

difference of systems Ψ
 and Φ , $\Psi \setminus \Phi$
 complement of system Φ ,
 $\complement \Phi$
 complement of system Φ ,
 regarding system Ψ , $\complement_\Psi \Phi$
 open system \mathcal{O}
 informational topology \mathcal{O}
 informational topological
 structure \mathcal{O}
 topological informational
 space $\langle \Phi, \mathcal{O} \rangle$, simply, Φ
 informational carrier Φ
 informational point,
 formula, formula system
 $\varphi \in \Phi$

MATHEMATICAL VS. INFORMATIONAL GRAPH THEORY [4, 35, 39]

vertex (apex, in Russ.,
 вершина [39]), v
 set of vertices, V
 vertex connection
 (rib, ребро) u :
 arc, loop, link [39]
 set of ribs, U
 edge, unordered pair
 $e = \{v_1, v_2\}$, or
 ordered pair (v_1, v_2)
 set of edges, E
 (incidence function)
 path (route, цепь)
 $v_1 u_1 v_2 u_2 \dots u_{n-1} v_n$

operand, operand point, ξ

system of operands, Φ_ξ
 operator, operator arrow,
 marked by \models or by an
 operator particularization

list of operator markers
 basic transition, $\xi \models \eta$,
 with binary operator

system of basic transitions
 $\Phi_{\xi \models \eta}$
 informational route, path,
 formula scheme

$\xi_1 \models \xi_2 \models \dots \models \xi_n$
 graph $G = (V, E)$ [2] informational graph, pre-
 sented by $\Phi_{\xi \models \eta}$, derived
 from an actual system Φ_φ

Informational space shall mean a non-empty formula system which possesses some type of informational structure (and organization), e.g. metaphysicalism, meaning, informational vector, informational metric (in the form of informational distance) and/or informational topology. Within such a possible structure, the elements in an informational space will be called *formulas* or *points* ($\Phi_\varphi, \Phi_{\xi \models \eta}$) and, in a special case (Φ_ξ), *operands*.

3 Systems and Subsystems of Informational Formulas

3.1 Informationally Linked Formulas in a System

Informational linkage of formulas in a formula system deserves a special attention and theoretical treatment. The consequence of formula linkages via common operands makes the difference between formulas of a system on one side, and between the elements of a set on the other one.

DEFINITION 1 *If in informational formulas φ_1 and φ_2 a common operand α appears, that is, $\varphi_1[\dots, \alpha, \dots]$ and $\varphi_2[\dots, \alpha, \dots]$,⁷ respectively, notation*

$$\varphi_1 \overset{\alpha}{\rightsquigarrow} \varphi_2 \text{ or, simply, } \varphi_1 \rightsquigarrow \varphi_2$$

will be used and read as formula φ_1 informs formula φ_2 via operand α or, simply, formula φ_1 is informationally linked to formula φ_2 . This operation is informationally symmetric. Thus,

$$(\varphi_1 \overset{\alpha}{\rightsquigarrow} \varphi_2) \equiv (\varphi_2 \overset{\alpha}{\rightsquigarrow} \varphi_1)$$

Transitivity of operator \rightsquigarrow can exist in the following way: if α is common to φ_1 and φ_2 , and β is common to φ_2 and φ_3 , that is,

$$\varphi_1[\dots, \alpha, \dots], \varphi_2[\dots, \alpha, \beta, \dots] \text{ and } \varphi_3[\dots, \beta, \dots],$$

then φ_1 is linked informationally with φ_3 . Formally,

$$((\varphi_1 \overset{\alpha}{\rightsquigarrow} \varphi_2) \wedge (\varphi_2 \overset{\beta}{\rightsquigarrow} \varphi_3)) \implies (\varphi_1 \overset{\alpha, \beta}{\rightsquigarrow} \varphi_3)$$

Operator \wedge denotes informational conjunction (in fact, the operator of parallel informing, \models , or, usually, semicolon ';' [30]) and operator \implies informational implication [30]. □

⁷It is to stress that a notation $\varphi[\alpha_1, \alpha_2, \dots, \alpha_n]$ means informational operands $\alpha_1, \alpha_2, \dots, \alpha_n$ occurring in formula φ , and does not represent the so-called functional form, that is, informational Being-of [29] in the form $\varphi(\alpha_1, \alpha_2, \dots, \alpha_n)$. Evidently, $\varphi(\alpha_1, \alpha_2, \dots, \alpha_n) \implies \varphi[\alpha_1, \alpha_2, \dots, \alpha_n]$.

Further, there can exist more than one common operand, e.g., $\alpha_i, \alpha_j, \alpha_k, \dots, \alpha_m$ in φ_1 and φ_2 in a transitive manner. In this case,

$$\left(\varphi_1 \begin{matrix} \alpha_i \\ \alpha_j \\ \alpha_k \\ \vdots \\ \alpha_m \end{matrix} \varphi_2 \right) \Rightarrow \begin{pmatrix} \varphi_1 \xleftrightarrow{\alpha_i} \varphi_{i'}; \\ \varphi_{i'} \xleftrightarrow{\alpha_j} \varphi_{j'}; \\ \varphi_{j'} \xleftrightarrow{\alpha_k} \varphi_{k'}; \\ \vdots \\ \varphi_{l'} \xleftrightarrow{\alpha_m} \varphi_2 \end{pmatrix}$$

Transitivity of operator $\xleftrightarrow{\alpha}$ applies also to the case of more than one common operator through several formulas, and it can be defined from case to case.

Because common operands concern informing between formulas, the implication in the last definition can be expressed by means of a parallel system

$$\left(\begin{matrix} \varphi_1 \xleftrightarrow{\alpha} \varphi_2; \\ \varphi_2 \xleftrightarrow{\beta} \varphi_3 \end{matrix} \right) \Rightarrow (\varphi_1 \xleftrightarrow{\alpha, \beta} \varphi_3)$$

Another significant feature follows from the last definition:

THEOREM 1 *Let the linkages in a circular manner*

$$\begin{matrix} \varphi_1 \xleftrightarrow{\alpha} \varphi_2, \\ \varphi_2 \xleftrightarrow{\beta} \varphi_3, \\ \vdots \\ \varphi_{m-1} \xleftrightarrow{\gamma} \varphi_m, \\ \varphi_m \xleftrightarrow{\delta} \varphi_1 \end{matrix}$$

be given. Then,

$$\varphi_1, \varphi_2, \dots, \varphi_m \xleftrightarrow{\alpha, \beta, \dots, \delta} \varphi_1, \varphi_2, \dots, \varphi_m$$

This feature is called the *reflexivity of a circular informational linkage of formulas within (in the framework of) a formula system*. \square

PROOF 1 We have to prove that

$$(\varphi_1, \varphi_2, \dots, \varphi_m \xleftrightarrow{\alpha, \beta, \dots, \delta} \varphi_1, \varphi_2, \dots, \varphi_m) \Rightarrow (\varphi_i \xleftrightarrow{\alpha, \beta, \dots, \delta} \varphi_j);$$

$$i, j \in \{1, 2, \dots, m\}$$

Within this conditionality also

$$\varphi_i \xleftrightarrow{\alpha, \beta, \dots, \delta} \varphi_i; \quad i \in \{1, 2, \dots, m\}$$

holds in a transitive (consequently multiple-linkage) manner.

Another evident meaning of the theorem result is

$$\varphi_i \xleftrightarrow{\alpha, \beta, \dots, \delta} \varphi_1, \varphi_2, \dots, \varphi_m \text{ for all } i \in \{1, 2, \dots, m\}$$

It means that φ_i is informationally linked to each of $\varphi_1, \varphi_2, \dots, \varphi_m$, including to itself (informational circularity). This proves the theorem. \square

3.2 Formula Systems

In informational theory, a system of informational formulas corresponds to the notion of a set of elements in mathematics. A fundamental concept of informational theory is that of the system (short for the system of informational formulas).

DEFINITION 2 *Intuitively, a system is a well-defined list of informationally well-formed formulas (separated by semicolons).*

Formulas consist of operands, (binary) operators, and parenthesis pairs.

In a system, formulas inform in parallel to each other. In a proper system, formulas are informationally linked via common operands, directly or indirectly (transitively), in such a way that each formula is, to some extent, informationally linked with each other formula of the system.

In an improper system, some formulas are informationally isolated.

Informationally, only proper systems appear (inform) to be reasonable. Isolated formulas inform per se, beyond the informational context of other formulas or subsystems of formulas in a system. For the rest of the system, such formulas are unobservable and do not observe informationally other formulas or formula subsystems. \square

The *union* of two systems Φ_1 and Φ_2 , denoted by $\Phi_1 \cup \Phi_2$, means the system

$$(\Phi_1 \cup \Phi_2) \Rightarrow \left(\varphi \mid \underbrace{(\varphi \in \Phi_1) \vee (\varphi \in \Phi_2)}_{\text{system classifier}} \right)$$

The union classifier can be expressed also as

$$((\varphi \in \Phi_1) \vee (\varphi \in \Phi_2)) \Rightarrow \left(\begin{matrix} \varphi \in \Phi_1, \\ \varphi \in \Phi_2 \end{matrix} \right)$$

which represents the so-called *alternative* system, using comma instead of semicolon between formulas [35].

The *intersection* of two systems Φ_1 and Φ_2 , denoted by $\Phi_1 \cap \Phi_2$, means the system

$$(\Phi_1 \cap \Phi_2) \Rightarrow \left(\varphi \mid \begin{matrix} \varphi \in \Phi_1; \\ \varphi \in \Phi_2 \end{matrix} \right)$$

If systems Φ_1 and Φ_2 do not have any formulas in common, $\Phi_1 \cap \Phi_2 \Rightarrow \emptyset$; they are said to be *disjoint* systems.

The *relative complement* of a system Φ_1 with respect to a system Φ , denoted by $\complement_{\Phi} \Phi_1$, or the *difference* of Φ and Φ_1 , denoted by $\Phi \setminus \Phi_1$, is the system

$$\Phi \setminus \Phi_1 \Rightarrow (\varphi \mid (\varphi \in \Phi) \vee (\varphi \notin \Phi_1))$$

The *complement* of a system Φ_1 , denoted by $\complement \Phi_1$, is the system

$$\complement \Phi_1 \Rightarrow (\varphi \mid (\varphi \in \Upsilon) \vee (\varphi \notin \Phi_1))$$

where Υ functions as a universal system. Evidently, $\mathbb{C}\Phi_1 = (\Upsilon \setminus \Phi_1)$. Usually, in a complex case, the formula system Φ has the role of the currently universal system to which its subsystems can be compared.⁸

4 Topological Informational Spaces

4.1 Definitions

4.1.1 Open Systems

DEFINITION 3 Let Φ mark a reasonable⁹ non-empty system of informational formulas. A class¹⁰ (short for informational class) \mathcal{D} of subsystems of Φ , $\mathcal{D} \subset \mathfrak{P}[\Phi]$, is a topology on Φ iff \mathcal{D} satisfies the following axioms:

- (T_I) The union of any number of systems in \mathcal{D} belongs to \mathcal{D} .
- (T_{II}) The intersection of any two systems in \mathcal{D} belongs to \mathcal{D} .
- (T_{III}) Systems Φ and \emptyset belong to \mathcal{D} .

The systems of \mathcal{D} are then called \mathcal{D} -open systems, or simply open systems, and Φ together with \mathcal{D} , i.e. the informational pair $\langle \Phi, \mathcal{D} \rangle$, is called the topological informational space. \square

As we see, a topological space is defined as an ordered pair between the carrier Φ and its topology \mathcal{D} .

Let us formulate Def. 3 in another way to get a different experience of the meaning of an informational topology. Topology can be determined by the following four steps too:

- 1⁰ A basic system Φ of formulas $\varphi_1, \varphi_2, \dots, \varphi_m$ exists.
- 2⁰ There exists a type characteristics¹¹ $\mathcal{D} \in \mathfrak{P}[\mathfrak{P}[\Phi]]$.
- 3⁰ The first axiom is:
For each system \mathcal{D}' , the informational implication $(\mathcal{D}' \in \mathcal{D}) \implies \left(\left(\bigcup_{\Xi \in \mathcal{D}'} \Xi \right) \in \mathcal{D} \right)$ and $\Phi \in \mathcal{D}$ hold.
- 4⁰ The second axiom is:
For each Ξ_1 and each Ξ_2 , the informational implication $(\Xi_1 \in \mathcal{D}; \Xi_2 \in \mathcal{D}) \implies ((\Xi_1 \cap \Xi_2) \in \mathcal{D})$ holds.

Such a structure family is called the topological structure, and the relation $\Xi \in \mathcal{D}$ can be expressed verbally as: system Ξ is open in topology \mathcal{D} .¹²

⁸A complex system is, for example, that of informational consciousness, in which several complex subsystems are imbedded. However, it does not mean that, in a specific case, a subsystem appears as a kind of universal to which its subsystems can be system-complementally compared.

⁹A reasonable system of informational formulas usually concerns a concrete, cyclically structured informational graph [32, 35].

¹⁰A class (family, collection) of subsystems means a system of subsystems.

¹²For more details see [6], p. 246. Evidently, the openness of a system concerns its topology.

EXAMPLE 1 Topologies Deducible from Standardized Metaphysicalism. Let the following classes of subsystems of the standardized metaphysicalistic system [32, 35] $\mathfrak{M} = (\varphi_1; \varphi_2; \dots; \varphi_6)$ of circular formulas¹³ be given:

- $\mathcal{D}_1 = (\mathfrak{M}; \emptyset; (\varphi_6));$
- $\mathcal{D}_2 = (\mathfrak{M}; \emptyset; (\varphi_5; \varphi_6); (\varphi_6));$
- $\mathcal{D}_3 = (\mathfrak{M}; \emptyset; (\varphi_4; \varphi_5); (\varphi_6));$
- $\mathcal{D}_4 = (\mathfrak{M}; \emptyset; (\varphi_4; \varphi_5; \varphi_6); (\varphi_6));$
- $\mathcal{D}_5 = (\mathfrak{M}; \emptyset; (\varphi_1; \varphi_2; \varphi_3; \varphi_6); (\varphi_4; \varphi_5; \varphi_6); (\varphi_6))$

For \mathcal{D}_3 , the union

$$((\varphi_4; \varphi_5) \cup (\varphi_6)) = (\varphi_4; \varphi_5; \varphi_6)$$

does not belong to \mathcal{D}_3 . Evidently, \mathcal{D}_3 is not a topology of \mathfrak{M} . \square

Formulas of a topological informational space are often called *points*. System Φ , in which a topology is defined, is said to be the *carrier* of the topological space $\langle \Phi, \mathcal{D} \rangle$. System Φ can be the carrier of more than one topological space. Thus, a system of different topological spaces for Φ is, for instance,

$$\langle \Phi, (\mathcal{D}_1; \mathcal{D}_2; \dots; \mathcal{D}_n) \rangle = \begin{pmatrix} \langle \Phi, \mathcal{D}_1 \rangle; \\ \langle \Phi, \mathcal{D}_2 \rangle; \\ \vdots \\ \langle \Phi, \mathcal{D}_n \rangle \end{pmatrix}$$

EXAMPLE 2 The carrier of a topological informational space can be expressed in different ways, with different structural and organizational¹⁴ consequences. Conceptualizing an IS, at the first glance, usually, a system of serial and serially circular informational formulas is determined. According to [35], p. 114, such a system includes serial formulas of the type¹⁵

$${}^n\varphi_{\rightarrow} [\alpha, \alpha_1, \dots, \alpha_n]; 1 \leq i \leq N_{\rightarrow}; N_{\rightarrow} = \frac{1}{n+1} \binom{2n}{n}$$

¹³In this case, the following is meant:

- φ_1, φ_2 , and φ_3 mark the short loops ($\ell = 2$) of informing, counterinforming and embedding, respectively;
- φ_4 and φ_5 denote the middle-size loops ($\ell = 4$) for informing-counterinforming and counterinforming-embedding, respectively; and
- φ_6 represents the long loop ($\ell = 7$).

¹⁴Informational organization always concerns the substantial, semantic, meaning-like structure, the choice of entities and their decomposition, the reasonable setting of parenthesis pairs (parenthesizing), operational connections and informational loops, particularization of informational operators, system parallelization (introduction of parallel interpretive, detailing, explaining, complementary formulas and systems), etc. It concerns the significance, distinguishability, and specificity of a concrete structure representing a conventional informational case.

¹⁵Here, the delimiters $[,]$ are introduced to replace the parentheses $(,)$ when for $\varphi[\alpha]$ the occurrence (and not the functional dependence—the informational Being-of [29]) of α in formula φ is meant. Expression $\varphi[\alpha_1, \dots, \alpha_n]$ is read as formula φ with operands $\alpha_1, \dots, \alpha_n$, or also, formula φ concerning operands $\alpha_1, \dots, \alpha_n$. The idea to introduce this sort of parentheses (floor parentheses) comes from Bourbaki [5, 6] where the vertical snake-form parentheses are used (not available in L^AT_EX2 ϵ).

and circular serial formulas of the type

$${}^{n+1}\varphi_{\rightarrow}^{\circ}[\alpha, \alpha_1, \dots, \alpha_n]; 1 \leq i \leq N_{\rightarrow}^{\circ}; N_{\rightarrow}^{\circ} = \frac{1}{n+2} \binom{2n+2}{n+1}$$

where n applies for different lengths of formulas entering into the system. Thus, this kind of the system includes the last two types of formulas.

Another possibility is to express the system in the form of primitive informational parallelism, corresponding to the expression of an informational graph (for example, in [32, 35]). Such a system is obtained by the parallelization (in fact, reduction of formulas into primitive informational transitions [33]) of serial and circular serial formulas given previously. Parallelization is denoted by

$$\begin{aligned} {}^n\varphi'_{\parallel}[\alpha, \alpha_1, \dots, \alpha_{n-1}, \alpha_n] &\equiv \Pi' [{}^n\varphi_{\rightarrow}[\alpha, \alpha_1, \dots, \alpha_{n-1}, \alpha_n]]; \\ {}^{n+1}\varphi'_{\parallel}[\alpha, \alpha_1, \dots, \alpha_{n-1}, \alpha_n] &\equiv \Pi' [{}^{n+1}\varphi_{\rightarrow}[\alpha, \alpha_1, \dots, \alpha_{n-1}, \alpha_n]] \end{aligned}$$

for the serial and circular serial case, respectively. The parallel system is obtained by setting all the occurring primitive transitions into the system. As we know, such a primitive formula system represents the graph, describing all the possible informational situations of a system.

Different topologies can be constructed and studied for both circular serial and circular primitive parallel systems. We shall learn the essential differences and possibilities for both types of systems. \square

4.1.2 Further Examples and Definitions

(1) **Discrete spaces.** Let Φ be any formula system. The system of all possible subsystems of system Φ , $\mathfrak{P}[\Phi]$, satisfies the axioms for the open systems and is, in this sense, the topology in Φ , called the *discrete topology* on Φ . If Φ is topologized by its discrete topology, it is called the *discrete space*.

(2) **Indiscrete spaces.** Let Φ be any formula system. A system (\emptyset, Φ) satisfies the axioms for the open systems and is, in this sense, the topology on Φ , called the *indiscrete* (or *trivial*) topology in Φ . If Φ is topologized by its indiscrete topology, it is called the *indiscrete space*.

(3) **Basis of a topology.** Basis \mathfrak{B} of a topology \mathcal{D} in Φ is such a subsystem of family \mathcal{D} ($\mathfrak{B} \subset \mathcal{D}$) that each open system \mathcal{U} in \mathcal{D} ($\mathcal{U} \in \mathcal{D}$) is a union of some open systems \mathcal{A}_i in \mathfrak{B} , that is, $\bigcup_{\mathcal{A}_i \in \mathfrak{B}} \mathcal{A}_i$. Said in

another way: for each $\mathcal{U} \in \mathcal{D}$ and each point (formula) $\varphi \in \Phi, \mathcal{U}$, there exists such a system $\mathfrak{W} \in \mathfrak{B}$ that

$$\varphi \in \mathfrak{W}; \mathfrak{W} \subset \mathcal{U}$$

Open systems of the given basis \mathfrak{B} are called *basic open systems* of space (Φ, \mathcal{D}) .

(4) **Subbasis of a topology.** Subbasis of a topology \mathcal{D} in Φ consists of finite intersections (intersections of finite families) of open systems in \mathfrak{S} in such a way that these intersections constitute the basis of topology \mathcal{D} . Then, for each $\mathcal{U} \in \mathcal{D}$ and each $\varphi \in \mathcal{U}$ there exists a finite number of systems in \mathfrak{S} , for instance, $\mathfrak{W}_1, \dots, \mathfrak{W}_n$, such that

$$\varphi \in (\mathfrak{W}_1 \cap \dots \cap \mathfrak{W}_n); (\mathfrak{W}_1 \cap \dots \cap \mathfrak{W}_n) \subset \mathcal{U}$$

Open systems in the given subbasis \mathfrak{S} are called the *subbasic open systems* of space Φ .

It is evident that a topology \mathcal{D} of space Φ is completely determined by the basis or subbasis of \mathcal{D} .

(5) **Topological cover.** In general, we say that a family of systems $(\Xi_i)_{i \in I}$ is a cover of system Ψ , if $\Psi \subset (\bigcup_{i \in I} \Xi_i)$.

Cover $(\Xi_i)_{i \in I}$ of subsystem Ψ of topological informational space Φ is called *open*, if all Ξ_i are open systems in Φ .

4.1.3 Interior, Exterior, and Boundary Formulas (Points)

According to [13, 20], some further definitions could be useful also for the purposes of informational topology.

DEFINITION 4 Let Φ be a subsystem of system σ in topological space (σ, \mathcal{D}) .

(1) A formula $\varphi \in \Phi$ is called *interior formula (point)* of Φ if φ belongs to an open system $\Psi \in \mathcal{D}$ contained in Φ , that is $\varphi \in \Psi$ and $\Psi \subset \Phi$, where Ψ is open. The system of interior formulas of Φ , denoted by Φ_{int} , is called the *interior* of Φ .

(2) A formula $\varphi \in \Phi$ is called *exterior formula (point)* of Φ if φ belongs to an open system $\Xi \in \mathcal{D}$ contained in the complement $\complement\Phi$, that is $\varphi \in \Xi$; $\Xi \subset \complement\Phi$, where Ξ is open. The system of exterior formulas of Φ , denoted by $\overline{\Phi}$, is called the *exterior* of Φ .

(3) A formula $\varphi \in \sigma$ is called *boundary formula (point)* of Φ if in each neighborhood of φ formulas of Φ and $\complement\Phi$ occur. The system of all boundary formulas is called the *boundary* and denoted by $\beta[\Phi]$. \square

The situation is presented diagrammatically in Fig. 1. For the boundary, there is, evidently,

$$\beta[\Phi] \equiv ((\overline{\Phi} \setminus \Phi) \cup (\Phi \setminus \Phi_{\text{int}}))$$

A SIGNIFICANT COMMENT A formula system Φ is open iff each of its formula is an interior formula (point). A formula system Ψ is closed iff its complement $\complement\Psi$ is an open formula system. \square

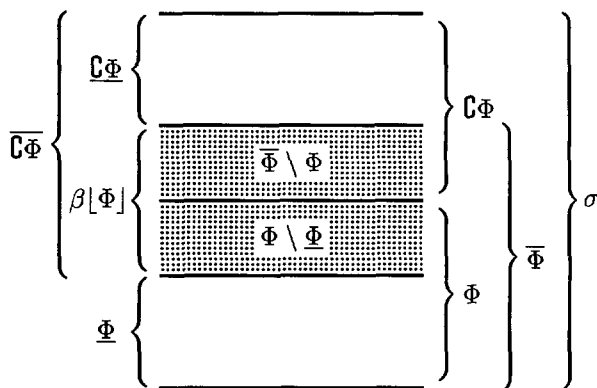


Figure 1: Diagram presentation of formula system Φ , its interior $\underline{\Phi}$, exterior $\overline{\Phi}$, and boundary $\beta[\Phi]$, within a system σ , where the complement of Φ , denoted by $\overline{C\Phi}$, appears together with the complement interior $\underline{C\Phi}$, exterior $\overline{C\Phi}$, and boundary $\beta[\overline{C\Phi}] = \beta[\Phi]$.

EXAMPLE 3 The connection of an informational loop and to it belonging system, that is, circular formula system, with the environment, can be realized by a special formula, usually a simple transition formula, e.g., $\alpha_j \models \xi_{ij}$ in Fig. 4. This formula belongs to Φ_φ , but is not an interior formula of Φ_φ , that is, of $\underline{\Phi}_\varphi$. □

EXAMPLE 4 Let systems Φ, B, Ψ form a topological space $((\Phi \cup B) \cup \Psi, \mathcal{D})$, as presented in Fig. 2. How

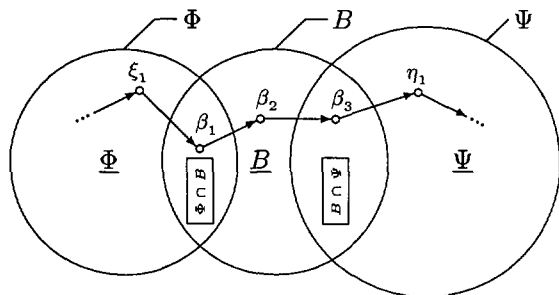


Figure 2: Graphical presentation of formula system $\sigma = (\Phi; B; \Psi)$, where each of the subsystems has its interior $\underline{\Phi}, \underline{B}, \underline{\Psi}$, respectively.

can this system informational graph be interpreted in different ways?

We must clarify more precisely what subsystems, marked by Φ, B, Ψ , might represent. The aim of the graph is to explicate the so-called interior and neighborhood regions of subsystems in respect to informational formulas in general, their operands, and the so-called primitive transitions, represented in the form of the graph route¹⁶

$$\dots \models \xi_1 \models \beta_1 \models \beta_2 \models \beta_3 \models \eta_1 \models \dots$$

¹⁶In literature, different names are given to the route. Informationally, the name informational scheme or, in short, scheme, is used. The name edge denotes the edge (representing an operator)

The route drawn in Fig. 2 runs through the operands

$$\xi_1, \beta_1 \in \Phi; \beta_1, \beta_2, \beta_3 \in B; \beta_3, \eta_1 \in \Psi$$

where

$$\beta_1 \in \Phi, B \text{ and } \beta_3 \in B, \Psi$$

According to Fig. 2, parts of system B function as neighborhoods of Φ and Ψ , respectively. System \underline{B} as the interior of B is represented by

$$\underline{B} = (B \setminus ((\Phi \cap B) \cup (B \cap \Psi)))$$

that is, considering both systems Φ and Ψ . On the other side, the exterior of $\Phi, \overline{\Phi}$, could roughly be understood as the union $\Phi \cup \underline{B}$ and, adequately, $\overline{\Psi}$ as $\Psi \cup \underline{B}$. In general, B is the neighborhood system for both Φ and Ψ . Further, evidently,

$$\begin{aligned} C\Phi &= \underline{B} \cup \Psi; C\Psi = \underline{B} \cup \Phi; \\ \underline{C\Phi} &= B \cup \Psi; \underline{C\Psi} = \Phi \cup B; \\ \overline{\Phi} &= \Phi \cup \underline{B}; \overline{\Psi} = \Psi \cup \underline{B} \end{aligned}$$

etc. This example shows the importance of distinguishing the three possible types of topological spaces: (1) a space of circular/serial informational formulas of arbitrary lengths, Φ_φ ; (2) a space of basic transition formulas of length 1, $\Phi_{\xi=\eta}$; and (3) a space of simple informational operands (formula length 0), Φ_ξ . □

Another example shows the questionableness of a unique determination of topological structures dealing with different types of informational formula systems ($\Phi_\varphi, \Phi_{\xi=\eta}$, and Φ_ξ).

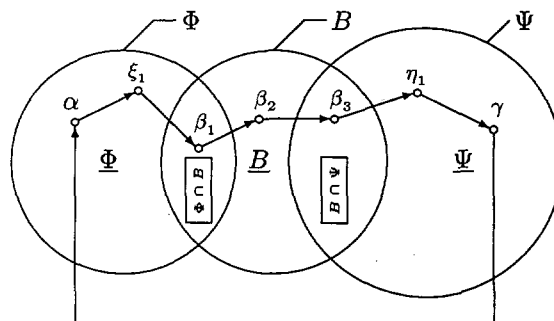


Figure 3: Graphical presentation of formula system $(\Phi \cup B) \cup \Psi$, including a circular path.

EXAMPLE 5 According to Fig. 3, let formula systems

of a graph polyhedron, to which a graph can be transformed. Path instead of the graph route sounds also adequately.

$$\begin{aligned}
 ((\Phi \cup B) \cup \Psi)_\varphi &= \left(\begin{array}{l} \alpha \models (\xi_1 \models (\beta_1 \models (\beta_2 \models (\beta_3 \\ \models (\eta_1 \models (\gamma \models \alpha)))))); \\ \vdots \end{array} \right); \\
 ((\Phi \cup B) \cup \Psi)_{\xi \models \eta} &= \left(\begin{array}{l} \alpha \models \xi_1; \xi_1 \models \beta_1; \beta_1 \models \beta_2; \\ \beta_2 \models \beta_3; \beta_3 \models \eta_1; \\ \eta_1 \models \gamma; \gamma \models \alpha; \dots \end{array} \right); \\
 ((\Phi \cup B) \cup \Psi)_\xi &= (\alpha; \xi_1; \beta_1; \beta_2; \beta_3; \eta_1; \gamma; \dots)
 \end{aligned}$$

be given, where $\alpha \in \Phi_\xi$ and $\gamma \in \Psi_\xi$. What is now evident (or not quite evident) from the topological graph in Fig. 3? Can basic transition formula systems be determined uniquely, and in which way?

The problem occurs at basic transitions crossing the boundaries of systems Φ, B and Ψ . In Fig. 3 such transitions are $\xi_1 \models \beta_1, \beta_3 \models \eta_1$, and $\gamma \models \alpha$. Evidently, in a strict situation, it would be not possible to express the basic transition systems rigorously. Thus, the compromise notation¹⁷

$$\begin{aligned}
 \Phi_{\xi \models \eta} &= (\models \alpha; \alpha \models \xi_1; \xi_1 \models \beta_1; \beta_1 \models); \\
 B_{\xi \models \eta} &= (\models \beta_1; \beta_1 \models \beta_2; \beta_2 \models \beta_3; \beta_3 \models); \\
 \Psi_{\xi \models \eta} &= (\models \beta_3; \beta_3 \models \eta_1; \eta_1 \models \gamma; \gamma \models)
 \end{aligned}$$

can be accepted, where the transition $\gamma \models \alpha$ comes additionally. Further,

$$\Phi_\xi = (\alpha; \xi; \beta_1); B_\xi = (\beta_1; \beta_2; \beta_3); \Psi_\xi = (\beta_3; \eta; \gamma)$$

In this situation, Fig. 3 shows $(\Phi \cap \Psi) = \emptyset$. □

The last example presents how basic transition formula systems and operand systems derived from general formula systems can offer various informational interpretations.

4.1.4 Informational Neighborhood

Informational neighborhood (neighborhood, for short) is both a metaphor and a formalistic structure concerning various possibilities of informational relationships between formulas and formula systems.

DEFINITION 5 *Neighborhood of system Ξ in a topological space $\langle \Phi, \mathcal{D} \rangle$ is called each system which includes an open system including Ξ . Neighborhoods of one-formula system (φ) are said to be also the neighborhoods of formula φ .* □

Let us present the last definition by other words.

(1) Let φ be a formula (point) of topological space $\langle \Phi, \mathcal{D} \rangle$, that is, $\varphi \in \Phi$. A subsystem Ξ of Φ ($\Xi \subset \Phi$) is a neighborhood of

¹⁷The compromise notation is, for example, $\models \alpha$ and $\gamma \models$. Each informational operator \models is a binary operator, dependent on both operands. If one side of the operator is open, it is meant, that the missing operand is not fixed yet.

φ iff Ξ is a supersystem of an open system Ψ ($\Xi \supset \Psi$) containing φ . Thus, $\boxed{\varphi \in \Phi, \Psi; \Psi \subset \Xi; \Xi \subset \Phi}$.

Other possible interpretations of the neighborhood definition are the following:

(2) A system Ξ in a topological space $\langle \Phi, \mathcal{D} \rangle$ is a neighborhood (\mathcal{D} -neighborhood) of a point (formula) φ iff Ξ contains an open system Ψ to which φ belongs.

(3) In a topological space $\langle \Phi, \mathcal{D} \rangle$, the neighborhood of the point $\varphi \in \Phi$ is called each subsystem $\Xi \subset \Phi$, including an open system Ψ such that $\varphi \in \Psi$ and $\Psi \subset \Xi$. Then, the neighborhood of subsystem $\Omega \subset \Phi$ is each subsystem $\Xi \subset \Phi$ which includes an open system Ψ , that is, satisfies $\Omega \subset \Psi$ and $\Psi \subset \Xi$. Thus, $\boxed{\varphi \in \Phi, \Psi; \Omega, \Xi \subset \Phi; \Psi \subset \Xi; \Omega \subset \Psi}$.

Evidently, each neighborhood of system Ξ in Φ is also a neighborhood of each system $\Psi \subset \Xi$ and, in particular, of each formula in Ξ . In turn, let Ξ be the neighborhood of each formula of system Ψ and Υ be the union of all open systems included in Ξ ; then $\Upsilon \subset \Xi$, as well as each formula of Ψ , belong to an open system, included in Ξ , that is, to $\Psi \subset \Upsilon$; but Υ is open according to (T_I) ; consequently, Ξ is the neighborhood of system Ψ . In particular, the following comes into the foreground:

SUPPOSITION 1 *That a system is the neighborhood of each its formula, it is necessary and sufficient for it to be open.*¹⁸ □

Let us mark by $\mathfrak{N}[\varphi]$ the system of all neighborhoods of formula φ . Systems in $\mathfrak{N}[\varphi]$ possess the following properties:

- (N_I) *Each subsystem of system Φ , including a system of $\mathfrak{N}[\varphi]$, belongs to $\mathfrak{N}[\varphi]$.*
- (N_{II}) *Intersection of a finite number of systems of $\mathfrak{N}[\varphi]$ belongs to $\mathfrak{N}[\varphi]$.*
- (N_{III}) *Formula φ belongs to each system in $\mathfrak{N}[\varphi]$.*

In fact, these three properties are a direct consequence of Def. 5 and Ax. (T_{II}) .

(N_{IV}) *For each system N belonging to $\mathfrak{N}[\varphi]$, there exists a system W belonging to $\mathfrak{N}[\varphi]$, such*

¹⁸In the informational sense, the word "neighborhood" has a meaning which points to the informational relationship, proximity, similarity, closeness and the like, coming to the surface as a sensefull informational formula of intuitively clear situations, facts and properties. In this way, the choice of this term has the advantage to make the speech figurative. For instance, Sup. 1 can be expressed in the following manner: for a system Ξ to be open, it is necessary and sufficient that for an arbitrary $\varphi \in \Xi$, all the formulas being sufficiently informationally close to φ , belong to Ξ . And, in general, if a property is true for all the formulas of a neighborhood of formula φ , it is said that this property holds for all the formulas, being sufficiently informationally close to φ . As we shall learn, informational closeness will concern, for instance, formulas of an entirely circularly connected system, with several loop formulas.

that N belongs to $\mathfrak{N}[\psi]$ for an arbitrary $\psi \in W$.

Evidently, considering Sup. 1, it suffices to take for W an arbitrary system, including formula φ , and being included in N .¹⁹

These four properties of system $\mathfrak{N}[\varphi]$ are said to be the neighborhood characteristics. Thus:

SUPPOSITION 2 *If to each formula φ of system Φ , a system $\mathfrak{N}[\varphi]$ of subsystems of Φ corresponds, and properties (N_I) , (N_{II}) , (N_{III}) and (N_{IV}) hold, then in Φ there exists a unique topological structure, for which $\mathfrak{N}[\varphi]$ serves as a system of all neighborhoods φ at an arbitrary $\varphi \in \Phi$.* \square

If there exists the required topological structure, then, by Sup. 1, as the system of all open systems of this topology necessarily serves the system \mathfrak{D} of all those systems Ξ in Φ , for which $\Xi \in \mathfrak{N}[\varphi]$ holds for each $\varphi \in \Xi$; this is the reason of a unique topology, if it exists.

4.1.5 Informational Bases and Subbases

The base \mathfrak{B} of a topological space $\langle \Phi, \mathfrak{D} \rangle$ is a system of formula systems B such that any formula φ of Φ belonging to subsystem B , B is a subsystem of \mathfrak{G} in \mathfrak{D} .

DEFINITION 6 *A class (system of subsystems) \mathfrak{B} of open subsystems of Φ , in topological space $\langle \Phi, \mathfrak{D} \rangle$, is a base for the topology \mathfrak{D} iff (i) every open system $\mathfrak{G} \in \mathfrak{D}$ is the union of members of \mathfrak{B} . Equivalently, $\mathfrak{B} \subset \mathfrak{D}$ is a base for \mathfrak{D} iff (ii) for any formula φ belonging to an open system \mathfrak{G} , there exists $B \in \mathfrak{B}$ with $\varphi \in B$ and $B \subset \mathfrak{G}$.* \square

Let us examine the base \mathfrak{B} as a system of singleton systems.

EXAMPLE 6 Let a topological formula space $\langle \Phi, \mathfrak{F} \rangle$ be given. Then the class $\mathfrak{B} = \{(\varphi) \mid \varphi \in \Phi\}$ of all singleton subsystems of Φ is a base for formula topology \mathfrak{F} on Φ . Because each system (φ) is \mathfrak{F} -open, each $\mathfrak{A} \subset \Phi$ is \mathfrak{F} -open too. Furthermore, every system is the union of singleton systems. Thus, any other class \mathfrak{B}^* of subsystems of Φ is a base for \mathfrak{F} iff it is a superclass of \mathfrak{B} , that is, $\mathfrak{B}^* \supset \mathfrak{B}$. \square

Which are the necessary and sufficient conditions for a class of systems to be a base for some informational topology?

THEOREM 2 *Let \mathfrak{B} be a system of subsystems of a non-empty system Φ . Then \mathfrak{B} is a base for topology \mathfrak{D} on Φ iff*

¹⁹This property can be additionally expressed by saying that a neighborhood of formula φ is, besides, a neighborhood of all formulas being sufficiently close to φ .

(i) $\Phi = \bigcup_{B \in \mathfrak{B}} B$ and

(ii) for any $B, B^* \in \mathfrak{B}$, $B \cap B^*$ is the union of members of systems of \mathfrak{B} , or, equivalently, if $\varphi \in (B \cap B^*)$ then there exists $B_\varphi \in \mathfrak{B}$ such that $\varphi \in B_\varphi$ and $B_\varphi \subset (B \cap B^*)$. \square

Informational subbasis is another notion which could become relevant in the topological investigation of informational formula systems.

DEFINITION 7 *In a topological space $\langle \Phi, \mathfrak{D} \rangle$, a class \mathfrak{S} of open formula subsystems of Φ , that is, $\mathfrak{S} \subset \Phi$, is a subbasis for the topology \mathfrak{D} on Φ iff finite intersections of members of \mathfrak{S} form a base \mathfrak{B} for \mathfrak{D} .* \square

Any class \mathfrak{A} of formula subsystems of a non-empty formula system Φ is the subbasis for a unique topology \mathfrak{D} on Φ . Intersections of members of \mathfrak{A} form a base for the topology \mathfrak{D} on Φ .

EXAMPLE 7 Let $\Phi = (\varphi_0; \varphi_1; \dots; \varphi_6)$, according to the formula system, belonging to the graph in Fig. 4, and $\mathfrak{A} = \{(\varphi_3; \varphi_4); (\varphi_4; \varphi_5); (\varphi_6)\}$. Finite intersections of members of \mathfrak{A} gives the base

$$\mathfrak{B} = \{(\varphi_3; \varphi_4); (\varphi_4; \varphi_5); (\varphi_6); (\varphi_4); \emptyset; \Phi\}$$

By definition, $\Phi \in \mathfrak{B}$ follows, since it is the empty intersection of members of \mathfrak{A} -system. Considering unions of members of \mathfrak{B} gives the family

$$\mathfrak{D} = \{(\varphi_3; \varphi_4); (\varphi_4; \varphi_5); (\varphi_6); (\varphi_4); \emptyset; \Phi; (\varphi_3; \varphi_4; \varphi_6); (\varphi_4; \varphi_5; \varphi_6); (\varphi_3; \varphi_4; \varphi_5)\}$$

Formula system \mathfrak{D} is the topology on Φ generated by formula system \mathfrak{A} . \square

4.1.6 Informational Accumulation Point

Accumulation point (also, limit point) is a well-known term in mathematical topology. We need the notion of informational accumulation point, for example, as a formula or formula system approaching as close as possible to the meaning of something. This means that the final meaning of something can never be reached, although the meaning of something can be expressed by a formula system as close as required.

DEFINITION 8 *If \mathfrak{A} is a subsystem of a formula system Φ , formula $\varphi \in \Phi$ is an accumulation point of \mathfrak{A} , iff every open system \mathfrak{G} containing φ contains a point (formula) of \mathfrak{A} different from φ . There is,*

$$(\mathfrak{G} \text{ is open; } \varphi \in \mathfrak{G}) \implies ((\mathfrak{A} \cap (\mathfrak{G} \setminus \{\varphi\})) \neq \emptyset)$$

The system of accumulation points of \mathfrak{A} , marked by \mathfrak{A}' , is called the derived system of system \mathfrak{A} . \square

It usually happens that an accumulation point is informationally inexpressible, although a formula system as a point comes close and closest to the accumulation point.

4.1.7 Informational Connectedness and Compactness

Most of topological investigation concerns certain topological properties as *connectedness* and *compactness*. Intuitively, the connectedness of an informational space is a consequence of an adequate operand distribution in informational formulas.

Two formula subsystems \mathfrak{A} and \mathfrak{B} of a topological space $\langle \Phi, \mathfrak{D} \rangle$ are separated if \mathfrak{A} and \mathfrak{B} are operand disjoint and neither contains an accumulation point of the other. This means $\mathfrak{A}_\xi \cap \mathfrak{B}_\xi = \emptyset$.

A topological informational space $\langle \Phi, \mathfrak{D} \rangle$ is *disconnected* iff Φ is the union of two open, non-empty, disjoint subsystems of formulas, i.e.,

$$\Phi = (\mathfrak{A} \cup \mathfrak{B}); \mathfrak{A}, \mathfrak{B} \in \mathfrak{D}; \mathfrak{A} \cap \mathfrak{B} = \emptyset; \mathfrak{A}, \mathfrak{B} \neq \emptyset$$

DEFINITION 9 *An informational graph is said to be connected graph if there is a path (informational scheme) between every pair of operands in the graph. An informational graph is said to be circularly connected graph if there is a circular path (circular informational scheme) between every pair of operands in the graph.* \square

A connected graph represents an operand connected formula system. How the connected formula systems can be recognized topologically?

Because informational topology deals with formula systems, the connectedness can be recognized by the properties among formulas of a formula system. Visually, connectedness can be inspected by a graph, using Def. 9. Because a graph is uniquely described by the formula system $\Phi_{\xi \models \eta}$ deduced from the original formula system Φ_φ , connectedness is determined by the adequate transitivity of formulas in the sense

$$((\xi_1 \models \xi_2) \wedge (\xi_2 \models \xi_3)) \Rightarrow ((\xi_1 \models \xi_2) \models \xi_3) \vee (\xi_1 \models (\xi_2 \models \xi_3))$$

If $\varphi_i, \varphi_j \in \Phi_\varphi$, formulas φ_i and φ_j are connected iff there exists an operand ζ such that

$$\zeta \in (\Phi_\xi^{\varphi_i} \cap \Phi_\xi^{\varphi_j})$$

This means that operand systems $\Phi_\xi^{\varphi_i}$ and $\Phi_\xi^{\varphi_j}$, derived from formulas φ_i and φ_j , respectively, have the common operand ζ . This property coincides with the concept of informationally linked formulas in a system, discussed in Sect. 3.1.

The concept of cover [Sect. 4.1.2/(5)] is needed in the definition of compactness in the following sense.

DEFINITION 10 *A subsystem Ψ of informational topological space $\langle \Phi, \mathfrak{D} \rangle$ is compact if every open cover of Ψ is reducible to a finite cover.* \square

In another topological interpretation, if Ψ is compact and $\Psi \subset (\bigcup_{i \in I} \Xi_i)$, where Ξ_i are open formula systems, then it is possible to select a finite number of the open systems, say $\Xi_{i_1}, \dots, \Xi_{i_m}$, so that $\Psi \subset (\Xi_{i_1} \cup \dots \cup \Xi_{i_m})$.

4.2 Topologies Concerning Meaning

Let us present a direct generalization of the informational topology used in concern with the phenomenon of meaning²⁰ μ , for instance, as it can emerge within the various forms of understanding, interpretation, conceptualization, perception, consciousness, and the like.

DEFINITION 11 *Let Φ be a system of formulas, and assume that there exists a meaning formula $\mu[\varphi, \psi]$ on pairs of formulas $\varphi, \psi \in \Phi$ satisfying the following conditions:*

1. $\mu[\varphi, \psi] \neq \emptyset$;
2. $\mu[\varphi, \psi] \models \mu[\varphi]$ if and only if $\varphi = \psi$;
3. $\mu[\varphi, \psi] \models \mu[\psi, \varphi]$;
4. $(\mu[\varphi, \psi] \models \mu[\psi, \omega]) \models \mu[\varphi, \omega]$
(the informational transitional consequence)

We say that Φ is a meaning space with meaning μ , or with meaning difference μ . \square

Condition 1 says that system $\mu[\varphi, \psi]$ does not inform to be empty. Thus, this condition has also the meaning $\mu[\varphi, \psi] \neq \emptyset$. Condition 2 says that meaning concerning operands φ and ψ informs to be $\mu[\varphi]$ only and only if φ is the same operand as ψ . In this case, also $\mu[\varphi, \psi] = \mu[\varphi]$. By Condition 3, since meaning concerning two operands, $\mu[\varphi, \psi]$, is a meaning difference between φ and ψ , such a difference informs to be $\mu[\psi, \varphi]$. Thus, $\mu[\varphi, \psi] = \mu[\psi, \varphi]$. It can certainly be introduced a difference between the symmetrical cases by the distinguishing $\mu[\varphi, \psi] \neq \mu[\psi, \varphi]$. Such a condition would ruin the traditional convenience of space metrication.

Meaning of something as an informational phenomenon follows the possibilities of informational decomposition and, in this sense, offers various possibilities for the play with meaning topologies in informational spaces. The diversity of decomposition is pointed out, for instance, by Haney [15], using the term deconstruction²¹. Meaning as informational decomposition makes an informational space a continuum, in which topological notions of open systems,

²⁰Informationally, meaning replaces the so-called metric in mathematics. The metric spaces (introduced by Fréchet in 1906 [12]) are based on the concept of distance [25]. Usually, a metric can be introduced in real numbers and in other kinds of mathematical spaces (e.g., in Hilbert space). A metric provides an easy way to define a topology in a metric space. To the topologist, the particular metric used on a space is merely a convenient way to define open sets [16].

²¹In Haney [15], the following is stated: ... it would be an overgeneralization to say, as the tendency is in 'American deconstruction', that all meaning is indeterminate, all presence illusory, all

interior, exterior, boundary, neighborhood, accumulation point, etc. become reasonable for a formalistic and artificial construction (composition) of formula systems.

4.3 Topologies Concerning Distributed (Parallel) Systems

DEFINITION 12 (Distributed System) *A parallel informational space \mathcal{D} with operands (points, a kind of vectors) δ being distributed by their components (point coordinates) $\xi_1^\delta, \dots, \xi_{n_\delta}^\delta$ is called the distributed informational space or \mathcal{D} -space if:*

1. A rule is given by which to each pair of points δ, ε of space \mathcal{D} the parallelism $(\delta; \varepsilon)$ corresponds.
2. This rule satisfies the following conditions:
 - a) $(\varepsilon; \delta) \equiv (\delta; \varepsilon)$ (displacement law);
 - b) $(\delta; \varepsilon \models \zeta) \equiv ((\delta; \varepsilon) \models (\delta; \zeta))$ (distributive law);
 - c) $(\mu\delta; \varepsilon) \equiv \mu(\delta; \varepsilon)$ for an arbitrary functional operand μ ;
 - d) $(\delta, \delta) \not\models \emptyset$ for $\delta \neq \emptyset$ and $(\delta, \delta) \equiv \emptyset$ for $\delta = \emptyset$.

By axioms b) and c) the general formula

$$(\varphi[\mu_1\delta_1, \dots, \mu_k\delta_k]; \psi[\nu_1\varepsilon_1, \dots, \nu_m\varepsilon_m]) \equiv \omega[\mu_1\nu_1(\delta_1; \varepsilon_1), \mu_1\nu_2(\delta_1; \varepsilon_2), \dots, \mu_1\nu_m(\delta_1; \varepsilon_m), \mu_2\nu_1(\delta_2; \varepsilon_1), \mu_2\nu_2(\delta_2; \varepsilon_2), \dots, \mu_2\nu_m(\delta_2; \varepsilon_m), \dots, \mu_k\nu_1(\delta_k; \varepsilon_1), \mu_k\nu_2(\delta_k; \varepsilon_2), \dots, \mu_k\nu_m(\delta_k; \varepsilon_m)]$$

is obtained which holds for arbitrary vectors $\delta_1, \dots, \delta_k, \varepsilon_1, \dots, \varepsilon_m$ and arbitrary meanings $\mu_1, \dots, \mu_k, \nu_1, \dots, \nu_m$. \square

To explicate the vector nature of points (formulas and/or formula systems) in a distributed informational space \mathcal{D} , let us introduce the vector notation (of the basic degree) of points in the form $\|\delta\|$ [32, 38]. The question is which formula components constitute the informational vector $\|\delta\|$? Evidently, the structure of an informational vector is not as simple as in a mathematical vector space. Let us discuss several formula notations constituting a vector.

The basic constituent is obviously the system of all simple operands appearing in a formula system δ and being denoted by

$$\rightarrow\delta_{\parallel}^0 \left[\xi_1^\delta, \dots, \xi_{n_\delta}^\delta \right] \equiv \left(\xi_1^\delta, \dots, \xi_{n_\delta}^\delta \right)$$

theme of intention irrelevant, all reference a fiction, etc. (see Attridge [1] p. 12). That a text for Derrida, especially a literary text, is always situated, read and re-read in a specific place and times makes it 'iterable' or repeatable, the same but always different, and therefore never reducible to an abstraction by theoretical contemplation (Derrida [11] pp. 172-97). A text is unique and repeatable, concrete and abstract simultaneously. This coexistence lies in the heart of deconstruction and reflects the connectedness of the subject and object in the experience of the self as pure consciousness.

Besides, parallel components $\xi_1^\delta, \dots, \xi_{n_\delta}^\delta$ can appear and be distributed within different kinds of formulas, or even form a serial or circular serial formula as a whole, that is

$$\begin{aligned} \rightarrow\delta_{\parallel}^{\ell \rightarrow} \left[\xi_1^\delta, \dots, \xi_{\ell \rightarrow}^\delta \right] &\equiv \rightarrow\delta_{\parallel}^{\ell \rightarrow} \left[\xi_1^\delta, \dots, \xi_{\ell \rightarrow}^\delta \right] \quad \text{and/or} \\ \circ\delta_{\parallel}^{\ell \circ} \left[\xi_1^\delta, \dots, \xi_{\ell \circ}^\delta \right] &\equiv \circ\delta_{\parallel}^{\ell \circ} \left[\xi_1^\delta, \dots, \xi_{\ell \circ}^\delta \right] \end{aligned}$$

respectively. Vector $\|\delta\|$ corresponding to system δ is determined by

$$\|\delta\| \equiv \left(\begin{array}{c} \rightarrow\delta_{\parallel}^0 \left[\xi_1^\delta, \dots, \xi_{n_\delta}^\delta \right]; \\ \rightarrow\delta_{\parallel}^{\ell \rightarrow} \left[\xi_1^\delta, \dots, \xi_{\ell \rightarrow}^\delta \right]; \ell \rightarrow \in \{1, 2, \dots, \ell_{\rightarrow}^{\max}\}; \\ \circ\delta_{\parallel}^{\ell \circ} \left[\xi_1^\delta, \dots, \xi_{\ell \circ}^\delta \right]; \ell \circ \in \{2, 3, \dots, \ell_{\circ}^{\max}\} \end{array} \right)$$

The structure of vector $\|\delta\|$ needs to be additionally explained. What does such a vector include and in which sense the difference between the mathematical and informational vector comes to the surface?

First, let us list all the components of vector $\|\delta\|$ in concern to the origin system δ . System δ is simply a parallel system of serial and/or circular serial formulas. But, in fact, this list is in no way a complete one in regard to the complex parallelism hidden in particular formulas of δ . The reader should remind the axiomatic approach of the informational where the fundamental axiom is expressed by the implication

$$(\alpha \models \beta) \implies \left(\begin{array}{c} \alpha; \\ \beta; \\ \alpha \models \beta \end{array} \right)$$

If this rule is recursively applied to a serial or circular serial formula $\rightarrow\varphi_{\rightarrow}^n [\alpha, \alpha_1, \dots, \alpha_n]$ or $\circ\varphi_{\rightarrow}^{n+1} [\alpha, \alpha_1, \dots, \alpha_n]$, respectively, then, evidently, the application of the last axiom delivers all the sub-formulas appearing in a serial and/or circular serial formula, that is, in the serial case,

$$\rightarrow\varphi_{\rightarrow}^n [\xi, \xi_1, \dots, \xi_n] \implies \left(\begin{array}{c} \rightarrow\delta_{\parallel}^0 [\xi, \xi_1, \dots, \xi_n]; \\ \rightarrow\delta_{\parallel}^{1*} [\xi, \xi_1, \dots, \xi_n]; \\ \vdots \\ \rightarrow\delta_{\parallel}^{(n-1)*} [\xi, \xi_1, \dots, \xi_n]; \\ \rightarrow\varphi_{\rightarrow}^n [\xi, \xi_1, \dots, \xi_n] \end{array} \right)$$

and in the circular serial case,

$$\circ\varphi_{\rightarrow}^{n+1} [\xi, \xi_1, \dots, \xi_n] \implies \left(\begin{array}{c} \rightarrow\delta_{\parallel}^0 [\xi, \xi_1, \dots, \xi_n]; \\ \rightarrow\delta_{\parallel}^{1*} [\xi, \xi_1, \dots, \xi_n]; \\ \vdots \\ \rightarrow\delta_{\parallel}^{n*} [\xi, \xi_1, \dots, \xi_n]; \\ \circ\varphi_{\rightarrow}^{n+1} [\xi, \xi_1, \dots, \xi_n] \end{array} \right)$$

where the asterisked markers $\vec{\delta}_{\parallel}^{1*}, \dots, \vec{\delta}_{\parallel}^{(n-1)*}, \vec{\delta}_{\parallel}^{n*}$ denote the systems of serial subformulas of lengths $1, \dots, n-1, n$ conditionally in respect to operands ξ, ξ_1, \dots, ξ_n in floor parentheses. Namely, a system $\vec{\delta}_{\parallel}^{1*}[\xi, \xi_1, \dots, \xi_n]$ includes only and only such basic transitions of the form $\alpha_i \models \alpha_j$ ($\ell = 1$) which appear in formula $\vec{\varphi}_{\rightarrow}[\xi, \xi_1, \dots, \xi_n]$ or formula $\vec{\varphi}_{\rightarrow}^{\circ}[\xi, \xi_1, \dots, \xi_n]$ (as a whole), respectively. Similar concerns lengths ℓ up to value n or $n+1$, respectively.

A short analysis shows that in the serial and circular serial case the number of all possible subformulas of a given length can be evaluated by simple formulas. Let ℓ_{sub} mark the length of a subformula in a serial formula with the length ℓ_{\rightarrow} . Then, evidently, the number of such subformulas in a formula is

$$n_{\text{sub}}^{\rightarrow} = \begin{cases} \frac{\ell_{\rightarrow}}{\ell_{\text{sub}}}, & \text{if } \ell_{\rightarrow} \text{ is even} \\ \frac{\ell_{\rightarrow}+1}{\ell_{\text{sub}}}, & \text{if } \ell_{\rightarrow} \text{ is odd} \end{cases}$$

In a circular case there is

$$n_{\text{sub}}^{\circ} = \begin{cases} \frac{\ell_{\circ}}{\ell_{\text{sub}}}, & \text{if } \ell_{\circ} \text{ is even} \\ \frac{\ell_{\circ}-1}{\ell_{\text{sub}}}, & \text{if } \ell_{\circ} \text{ is odd} \end{cases}$$

5 Variants of Informational Topologies

A topology \mathcal{D} depends on the carrier system Φ , that is, on the characteristic forms of its formulas. Which kinds of formulas in Φ can be distinguished?

The most usual system of formulas is composed of different serial and circular-serial formulas. These formulas emerge during the analysis of an informational case, usually in a kind of top-down and bottom-up decomposition of an initial (top) marker or an end (bottom) marker, carrying implicitly a yet-not-determined concept, proceeding stepwise into a more detail of the case—a progressive case decomposition from different points of view. This approach seems to be the most natural one, seen from the human point of consciousness. Just after of such a case identification more abstract and convenient approach with possibilities can be considered.

DEFINITION 13 *The constructed system of formulas, Φ , can take the following characteristic forms:*

$$\begin{aligned} \Phi_{\varphi} &= (\varphi[\dots \xi_1 \dots]; \varphi[\dots \xi_2 \dots]; \dots; \varphi[\dots \xi_n \dots]); \\ \Phi_{\xi} &= (\xi_1; \xi_2; \dots; \xi_n) \cup \Phi[\xi_i | \text{implicit operands}]; \\ \Phi_{\xi \models \eta} &= (\xi_1 \models \xi_2; \xi_2 \models \xi_3; \dots; \xi_{n-1} \models \xi_n) \cup \\ &\quad \Phi[\xi_i \models \xi_j | \text{implicit basic transitions}] \end{aligned}$$

The first system, Φ_{φ} , is an authentic, intuitively constructed representation of a real case. The second system, Φ_{ξ} , is strictly expressed by all the occurring system operands as the title operands of a circular formula

system, each operand in at least one circular formula. The third system, $\Phi_{\xi \models \eta}$, is the representative of all possible situations occurring by all possible parenthesis pairs displacements within the constructed (analyzed and synthesized) system. \square

As said, the originally conceptualized system (obtained by the top-down or bottom-up approach or from both of them) is Φ_{φ} . Thus, the remaining two systems, Φ_{ξ} and $\Phi_{\xi \models \eta}$, evidently emerge from Φ_{φ} , that is,

$$\Phi_{\varphi} \longrightarrow \Phi_{\xi} \quad \text{and} \quad \Phi_{\varphi} \longrightarrow \Phi_{\xi \models \eta}$$

where \longrightarrow denotes the corresponding derivation approach. On this basis, three different topologies can be determined, as formula, operand and basic-transition topology, respectively. $\Phi_{\varphi} \longrightarrow \Phi_{\xi \models \eta}$ is the formal representative of the corresponding informational graph [35].

Now, let us show, how different topologies can be defined on Φ_{φ} , Φ_{ξ} and $\Phi_{\xi \models \eta}$ in a concrete case, and how all they mirror one and the same informational graph, with different possibilities in regard to various parenthesis displacements in formulas of the system. As an example we choose the metaphysicalistic case.

5.1 Topologies of a Simple Metaphysicalism

Simple metaphysicalism is a basic scheme of informational invariance which can be further decomposed in greater details during identification of the involved entities, that is, a formula expressed in the metaphysicalistic form. Thus, the graph in Fig. 4 can be understood as a consequence of the circular metaphysicalistic formula system $\boxed{k_j \varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]}$. Subscript j concerns the formula system component $k_j \varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]$ of system Φ_{φ} , e.g., $j = 1, 2, \dots, n$. Subscript i concerns the operand component ξ_i of metaphysicalistic formula system $k_j \varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j] \in \Phi_{\varphi}$. Subscript k_j concerns the parenthesis-pair combination $1 \leq k_j \leq n_{\text{sub},j}^{\circ}$ of the formula subsystem system $k_j \varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]$, with altogether $\boxed{\prod_{\forall i,j} \frac{\ell_{ij}}{\ell_{ij}+1} (2\ell_{ij})}$ possibilities, considering serial (input) and circular serial formulas of a system of formula systems, where ℓ_{ij} denotes the length of the formula in a formula subsystem.

5.1.1 Topologies on the circular formula system $k_j \varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]$

According to the graph in Fig. 4, one of the possible formula systems can be constructed (reconstructed). Let it be the consequent observing type of metaphysicalism for which the extreme left-parenthesis heaping is characteristic, that is, $k_j = 1$. In this case, the graph

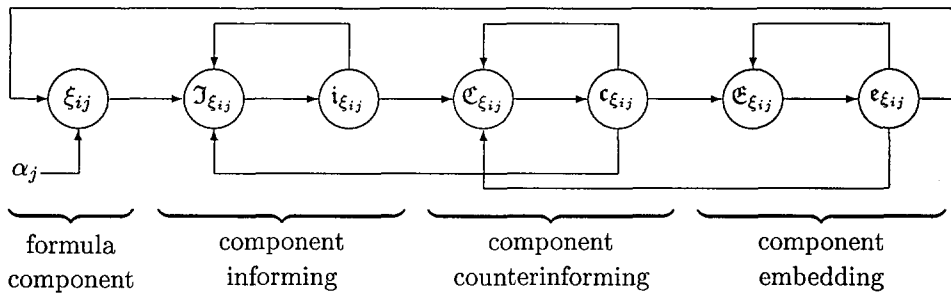


Figure 4: The graph representing the basic metaphysicalism of a formula system ${}_j\varphi_{\rightarrow}[\xi_{ij}, \alpha_j] \in \Phi_{\varphi}$ component ξ_{ij} , impacted by something (interior and/or exterior) α_j .

is interpreted by the one of possible formula systems, that is,

$${}_j\varphi_{\rightarrow}[\xi_{ij}, \alpha_j] = \left(\begin{array}{l} \alpha_j \models \xi_{ij}; \\ \left(\left(\left(\left(\xi_{ij} \models J_{\xi_{ij}} \right) \models i_{\xi_{ij}} \right) \models C_{\xi_{ij}} \right) \models c_{\xi_{ij}} \right) \models E_{\xi_{ij}} \right) \models e_{\xi_{ij}} \right) \models \xi_{ij}; \\ \left(\left(\left(J_{\xi_{ij}} \models i_{\xi_{ij}} \right) \models C_{\xi_{ij}} \right) \models c_{\xi_{ij}} \right) \models J_{\xi_{ij}}; \\ \left(\left(\left(C_{\xi_{ij}} \models c_{\xi_{ij}} \right) \models E_{\xi_{ij}} \right) \models e_{\xi_{ij}} \right) \models C_{\xi_{ij}}; \\ \left(J_{\xi_{ij}} \models i_{\xi_{ij}} \right) \models J_{\xi_{ij}}; \\ \left(C_{\xi_{ij}} \models c_{\xi_{ij}} \right) \models C_{\xi_{ij}}; \\ \left(E_{\xi_{ij}} \models e_{\xi_{ij}} \right) \models E_{\xi_{ij}} \end{array} \right) \quad \begin{array}{l} [\varphi_0] \\ [\varphi_1] \\ [\varphi_2] \\ [\varphi_3] \\ [\varphi_4] \\ [\varphi_5] \\ [\varphi_6] \end{array}$$

(0) According to the preceding notation, there is $\Phi_{\varphi} = {}_j\varphi_{\rightarrow}[\xi_{ij}, \alpha_j]$. To be more transparent, let us replace this system by the abbreviated one, in the form

$$\Phi_{\varphi} = \left(\begin{array}{l} \varphi_0; \\ \varphi_1; \\ \varphi_2; \\ \varphi_3; \\ \varphi_4; \\ \varphi_5; \\ \varphi_6 \end{array} \right)$$

representing the input transition formula and the six circular formulas, respectively. Which kind of topologies on Φ_{φ} can then be defined in a meaningful way?

(1) Let topology $\mathcal{D}_{\varphi,1}$, if possible, maintain the meaning of the original formula system Φ_{φ} in respect to the graph in Fig. 4. Let the meaningful condition be

$$(\varphi_1), (\varphi_2), (\varphi_3), (\varphi_4), (\varphi_5), (\varphi_6) \in \mathcal{D}_{\varphi,1}$$

by which all of the loops and the input transition enter the topology. What are the consequences of such a choice? First, the mutual intersections of formulas are empty systems, that is,

$$(\varphi_p \cap \varphi_q) = \emptyset; p \neq q; p, q = 0, 1, \dots, 6$$

However, all the possible unions of formulas $\varphi_0, \varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6$ must enter $\mathcal{D}_{\varphi,1}$. This condition

delivers together with \emptyset and Φ_{φ} a topology which is the power system of Φ_{φ} , $\mathfrak{P}[\Phi_{\varphi}]$, called the discrete topology $\mathcal{D}_{\varphi,1}$ (see Sect. 4.1.2). The precept of this example is that for all formulas $\varphi \in \Phi_{\varphi}$, $(\varphi) \in \mathcal{D}_{\varphi}$ implies that \mathcal{D}_{φ} is $\mathfrak{P}[\Phi_{\varphi}]$.

(2) A look to the graph in Fig. 4 brings to the surface another logic of topological understanding. Let us take the main loop φ_1 and one of the subloops, say φ_2 . In this case, further subsystems of topology $\mathcal{D}_{\varphi,2}$ are (φ_1) , (φ_2) , and $(\varphi_1; \varphi_2)$. Thus,

$$\mathcal{D}_{\varphi,2} = \left(\begin{array}{l} \emptyset; \\ (\varphi_1); \\ (\varphi_2); \\ (\varphi_1; \varphi_2); \\ \Phi_{\varphi} \end{array} \right)$$

One sees that this type of topology emerges independently on the chosen subloop. In case of two chosen subloops, say φ_2 and φ_3 , the topology becomes

$$\mathcal{D}_{\varphi,3} = \left(\begin{array}{l} \emptyset; (\varphi_1); (\varphi_2); (\varphi_3); \\ (\varphi_1; \varphi_2); (\varphi_1; \varphi_3); (\varphi_2; \varphi_3); \\ (\varphi_1; \varphi_2; \varphi_3); \Phi_{\varphi} \end{array} \right)$$

Let Φ_{φ} include n_{φ} formulas, $\varphi_1, \dots, \varphi_{n_{\varphi}}$. Then, for a topology $\mathcal{D}_{\varphi,i}$, implication

$$\left(((\varphi_1), (\varphi_2), \dots, (\varphi_i) \in \mathcal{D}_{\varphi,i}) \wedge (i < n) \right) \implies (n_{\mathcal{D}_{\varphi,i}} = 2^i + 1)$$

holds. Here, $n_{\mathcal{D}_{\varphi,i}}$ marks the number (cardinality) of subsystems in $\mathcal{D}_{\varphi,i}$.

(3) Other sensible topologies could consider specific situations in respect to the graph in Fig. 4. In constructing a topology, one can proceed from the other topological side, taking subsystems with more than one formula. Certainly, any other topology \mathcal{D}_{φ} on system Φ_{φ} is merely a subsystem of $\mathfrak{P}[\Phi_{\varphi}]$.

If the formula subsystems $(\varphi_0; \varphi_1; \varphi_2; \varphi_3)$ and $(\varphi_4; \varphi_5; \varphi_6)$ (the split of system Φ_{φ}) are joined to $(\emptyset; \Phi_{\varphi})$, no further subsystems are necessary for the

topology $(\emptyset; (\varphi_0; \varphi_1; \varphi_2; \varphi_3); (\varphi_4; \varphi_5; \varphi_6); \Phi_\varphi)$. Further, a characteristic implication is, for instance,

$$((\varphi_2; \varphi_4); (\varphi_3; \varphi_5); (\varphi_6) \in \mathfrak{D}_\varphi) \implies \left(\begin{array}{l} (\varphi_2; \varphi_3; \varphi_4; \varphi_5; \varphi_6); \\ (\varphi_2; \varphi_3; \varphi_4; \varphi_5); \\ (\varphi_2; \varphi_4; \varphi_6); \\ (\varphi_3; \varphi_5; \varphi_6) \end{array} \right) \in \mathfrak{D}_\varphi$$

The initial intention (premise of the implication) is to cover explicitly both subloops for $\mathcal{J}_{\xi_{ij}}$ by φ_2 and φ_4 , respectively, and both subloops for $\mathcal{E}_{\xi_{ij}}$ by φ_3 and φ_5 , respectively, including the covering of the main loop ξ by φ_1 . All these loops are implicitly covered by Φ_φ .

5.1.2 Topologies on the system of operands of the circular formula system

$${}_k \varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]$$

Any system operand as such (as an entity) can explicitly be extracted (expressed) by means of the other operands (including itself) of the system. Therefore, one can imagine the operand system as the one in which operands are representatives of their formula systems, that is, specific formula system markers. For instance, instead of the circular formula $\varphi_1 \in {}_1 \varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]$, we can put explicitly,

$$\xi_{ij} = \left(\begin{array}{l} ((((((\xi_{ij} \models \mathcal{J}_{\xi_{ij}}) \models i_{\xi_{ij}}) \models \mathcal{E}_{\xi_{ij}}) \models e_{\xi_{ij}}) \models \mathcal{E}_{\xi_{ij}}) \models e_{\xi_{ij}}) \models \xi_{ij} \end{array} \right)$$

etc. Thus, ξ_{ij} marks a formula of the kind

$${}_1 \xi_{ij \rightarrow}^{\circ}[\alpha_j, \xi_{ij}, \mathcal{J}_{\xi_{ij}}, i_{\xi_{ij}}, \mathcal{E}_{\xi_{ij}}, e_{\xi_{ij}}, \mathcal{E}_{\xi_{ij}}, e_{\xi_{ij}}]$$

In this sense, an operand system does not differ substantially from the formula system discussed in Sect. 5.1.1. The difference is that instead of formulas, the operands structuring them, come into the foreground. Thus, the basic system for the case in Fig. 4 is

$$\Phi_{\xi_{ij}} = \left(\begin{array}{l} \xi_{ij}; \\ \mathcal{J}_{\xi_{ij}}; \\ i_{\xi_{ij}}; \\ \mathcal{E}_{\xi_{ij}}; \\ e_{\xi_{ij}}; \\ \mathcal{E}_{\xi_{ij}}; \\ e_{\xi_{ij}} \end{array} \right)$$

In this situation, additionally, different operand roles can be explicated by means of different topologies. According to Fig. 4, some particularities can be stressed which do not proceed directly from the graph. For instance, topologically, a certain informational impact can be expressed between operands being not directly connected, for instance between $\mathcal{J}_{\xi_{ij}}$ and $\mathcal{E}_{\xi_{ij}}$, that is, the impacting of informing onto embedding. This

means, $(\mathcal{J}_{\xi_{ij}}, \mathcal{E}_{\xi_{ij}}) \in \mathfrak{D}_{\xi_{ij}}$. Such a topological condition could lead to the request that $\mathcal{J}_{\xi_{ij}}$ and $\mathcal{E}_{\xi_{ij}}$ must be explicitly expressed, for instance, applying the rotation principle of operands in the main loop, and then searching, how $\mathcal{J}_{\xi_{ij}}$ depends informationally on $\mathcal{E}_{\xi_{ij}}$, and vice versa. Namely, in a loop, cause and its consequence depend on each other.

In case of an operand topology, operands must be expressed explicitly, anyhow. The principle of operand rotation must be applied for operands which do not function as the main operands, that is, for $\mathcal{J}_{\xi_{ij}}$, $i_{\xi_{ij}}$, $\mathcal{E}_{\xi_{ij}}$, $e_{\xi_{ij}}$, $\mathcal{E}_{\xi_{ij}}$, and $e_{\xi_{ij}}$.

Besides, all the loops in the graph must be covered consequently. The rotation principle is one of the possibilities on this way. Let us rotate the embedding operand $e_{\xi_{ij}}$, considering system ${}_1 \varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]$, and looking into the graph in Fig. 4. The result is, from the consequent observational point of view,

$$e_{\xi_{ij}} = \left(\begin{array}{l} ((((((e_{\xi_{ij}} \models \xi_{ij}) \models \mathcal{J}_{\xi_{ij}}) \models i_{\xi_{ij}}) \models \mathcal{E}_{\xi_{ij}}) \models [\varphi_1]^{\circ}) \models e_{\xi_{ij}}) \models \mathcal{E}_{\xi_{ij}}) \models e_{\xi_{ij}}; \\ (((e_{\xi_{ij}} \models \mathcal{E}_{\xi_{ij}}) \models e_{\xi_{ij}}) \models \mathcal{E}_{\xi_{ij}}) \models e_{\xi_{ij}}; \\ (e_{\xi_{ij}} \models \mathcal{E}_{\xi_{ij}}) \models e_{\xi_{ij}}; \end{array} \right) \begin{array}{l} [\varphi_1]^{\circ} \\ [\varphi_3]^{\circ} \\ [\varphi_6]^{\circ} \end{array}$$

One can recognize how the rotation principle brings a new understanding of the metaphysicalistic system when new formula systems for metaphysicalistically interior operands come into consideration, complexing the system as a whole by detailing the before hidden additional possibilities of the interior operands. The last formula system is in its first part ($[\varphi_1]^{\circ}$, $[\varphi_3]^{\circ}$, and $[\varphi_6]^{\circ}$) informationally different to the adequate part of ${}_1 \varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]$.

The topological concept concerning operand system $\Phi_{\xi_{ij}}$ requests a more complex system in regard to the initial ${}_1 \varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]$. Substantially, the graph in Fig. 4 must be covered systematically irrespective of the operator rotation to the title (the leftmost) position of a circular formula.

5.1.3 Topologies on system of basic transitions of the circular formula system ${}_k \varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]$

Finally we come to the most significant and attractive form of topology determined on the system of basic transitions, that is, on basic serial formulas with the length $\ell = 1$ (e.g., of the form $\alpha \models \beta$). Why such a formula system could be of the primary interest, and why topologies on this formula system are informationally significant to a substantial extent?

To come into the course of the relevant discussion we have to remind on the informational equivalence existing between the so-called informational graph [35] and the corresponding parallel system of basic transitions. For instance, parallelizing a serial or circular

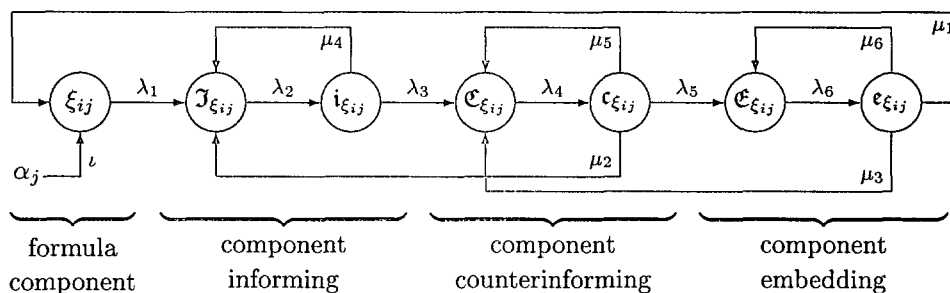


Figure 5: The graph representing the basic metaphysicalism of Fig. 4 in respect to primitive transitions ι , λ_1 , λ_2 , λ_3 , λ_4 , λ_5 , λ_6 , μ_1 , μ_2 , μ_3 , μ_4 , μ_5 , μ_6 .

serial formula means to decompose it into elementary transitions. Parallelization of a lengthy formula occurs by the following steps:

1. within a formula φ , all parenthesis pairs are omitted; what remains is called the route, sometimes scheme, or also framed scheme [37] (in Russian, *маршрут*, in [39], p. 85);
2. from the route, the system of basic transitions is constructed, moving from the left to the right along the route, building basic transition formulas by taking the two operands, connected by the operator between them in the route.

We see how the obtained paralleled system differs informationally from the original formula. The substantial difference occurs in the domain of informational operators. In a lengthy formula, an operator can connect two arbitrary subformulas which are not simple operands. However, an informational operator (a binary operator in any case) is always a product of its left and its right operand. Therefore the route must be understood as a frame scheme which only fixes the position of an operator (in the formula and then in the route), but does not definitely determine the subject of a concrete operator. This is the price which must be paid in any case of a formula reduction (in this case, in fact, generalization) where a concrete formula serves only as a sort of syntactical (structural, organizational) template.

A route $\xi_1 \models \xi_2 \models \dots \xi_{n-1} \models \xi_n$ is called *informational chain* (in Russian *цепь*, [39], p. 90) if operators \models in the route are mutually different. In an informational formula, informational operators are axiomatically mutually different. In mathematics, on contrary, equally denoted operators in the context of a mathematical formula (with a unique meaning, unique and firm definition) always represent equal operations. Routes or chains, respectively, are usually framed, to distinguish them clearly from formulas. For instance, $\boxed{\xi_1 \models \xi_2 \models \dots \xi_{n-1} \models \xi_n}$. Such a notation can be useful in cases where informational formulas and chains are combined, to enable the expression of

parts where parenthesizing is let open. A formula with framed routes is called framed informational formula.

The system from which the parallelization proceeds is ${}_k\varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]$, where k_j can be an arbitrary subscript in the interval concerning the formula system ${}_k\varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]$. Thus, one can take ${}_1\varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]$ and parallelize it according to the rules discussed in the previous text. Sometimes, the parallelization of ${}_1\varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]$ into the system of primitive transition formulas is marked by $\Pi'({}_1\varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j])$. The result is

$$\Pi'({}_1\varphi_{\rightarrow}^{\circ}[\xi_{ij}, \alpha_j]) = \varphi_{\parallel}^{\circ'}[\xi_{ij}, \alpha_j]$$

and, accordingly to Fig. 4, evidently,

$$\varphi_{\parallel}^{\circ'}[\xi_{ij}, \alpha_j] = \left(\begin{array}{l} \alpha_j \models \xi_{ij}; \\ \xi_{ij} \models \mathcal{J}\xi_{ij}; \mathcal{J}\xi_{ij} \models i\xi_{ij}; i\xi_{ij} \models \mathcal{C}\xi_{ij}; \\ \mathcal{C}\xi_{ij} \models c\xi_{ij}; c\xi_{ij} \models \mathcal{E}\xi_{ij}; \mathcal{E}\xi_{ij} \models e\xi_{ij}; \\ e\xi_{ij} \models \xi_{ij}; \\ c\xi_{ij} \models \mathcal{J}\xi_{ij}; \\ e\xi_{ij} \models \mathcal{C}\xi_{ij}; \\ i\xi_{ij} \models \mathcal{J}\xi_{ij}; \\ c\xi_{ij} \models \mathcal{C}\xi_{ij}; \\ e\xi_{ij} \models \mathcal{E}\xi_{ij} \end{array} \right) \begin{array}{l} [\varphi'_0] \\ [\varphi'_1] \\ [\varphi''_2] \\ [\varphi'_3] \\ [\varphi'_4] \\ [\varphi'_5] \\ [\varphi''_6] \end{array}$$

An informational transition formula can appear in the system only once. In this way, the last five rows include only the remaining feedback transitions.

(0) Let us introduce the notation $\Phi_{\xi \models \eta} \Rightarrow \varphi_{\parallel}^{\circ'}[\xi_{ij}, \alpha_j]$. For a better transparency, we replace the upper system $\varphi_{\parallel}^{\circ'}[\xi_{ij}, \alpha_j]$ by the abbreviated

$$\Phi_{\xi \models \eta} \Rightarrow \left(\begin{array}{l} \iota; \\ \lambda_1; \lambda_2; \lambda_3; \lambda_4; \lambda_5; \lambda_6; \mu_1; \\ \mu_2; \\ \mu_3; \\ \mu_4; \\ \mu_5; \\ \mu_6 \end{array} \right) \begin{array}{l} [\varphi'_0] \\ [\varphi'_1] \\ [\varphi''_2] \\ [\varphi'_3] \\ [\varphi'_4] \\ [\varphi'_5] \\ [\varphi''_6] \end{array}$$

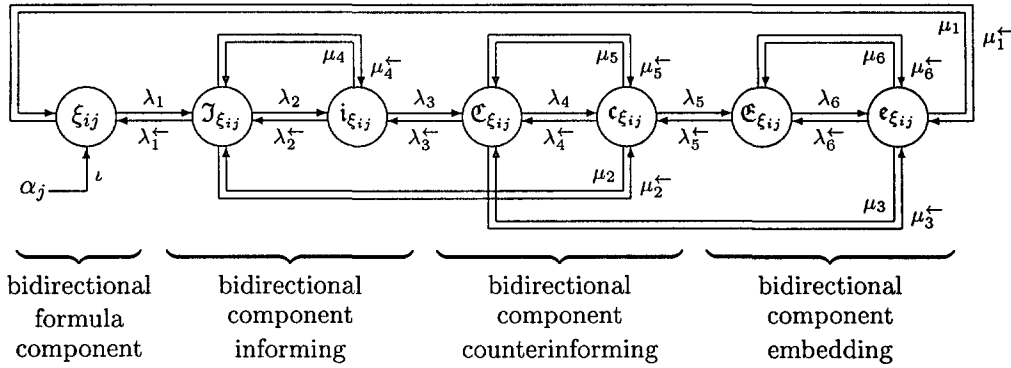


Figure 6: The bidirectional graph representing the metaphysicalism of Fig. 5 by considering the primitive onedirectional and counterdirectional transition pairs $(\lambda_1, \lambda_1^{\leftarrow}), (\lambda_2, \lambda_2^{\leftarrow}), (\lambda_3, \lambda_3^{\leftarrow}), (\lambda_4, \lambda_4^{\leftarrow}), (\lambda_5, \lambda_5^{\leftarrow}), (\lambda_6, \lambda_6^{\leftarrow}), (\mu_1, \mu_1^{\leftarrow}), (\mu_2, \mu_2^{\leftarrow}), (\mu_3, \mu_3^{\leftarrow}), (\mu_4, \mu_4^{\leftarrow}), (\mu_5, \mu_5^{\leftarrow}), (\mu_6, \mu_6^{\leftarrow})$.

respectively. The correspondence between transition formulas in $\varphi_{\xi}^{\prime}[\xi_{ij}, \alpha_j]$ and their abbreviated notations in $\Phi_{\xi=\eta}$ is evident. Notation ι marks the input transition, λ_p ($p = 1, \dots, 6$) the forward transition of the main loop, and μ_q ($q = 1, \dots, 6$) the feedback transition, corresponding to the graph in Fig. 4. The transitional situation is presented in Fig. 5. Systems $[\varphi_2''], \dots, [\varphi_6'']$ are already reduced by the common transitions within the main loop $[\varphi_1']$. Which kinds of sensible topologies can now be defined on $\Phi_{\xi=\eta}$?

(1) The basic topological question could concern the main loop in Fig. 5. The circular route of this loop is $\xi_{ij} \models J_{\xi_{ij}} \models i_{\xi_{ij}} \models C_{\xi_{ij}} \models c_{\xi_{ij}} \models \epsilon_{\xi_{ij}} \models \xi_{ij}$.

To this route, evidently, the subsystem $[\varphi_1']$, that is,

$$(\lambda_1; \lambda_2; \lambda_3; \lambda_4; \lambda_5; \lambda_6; \mu_1) \in \mathcal{D}_{\xi=\eta,1}$$

corresponds. If $[\varphi_1'] \subset \Phi_{\xi=\eta}$ is the only element besides \emptyset and $\Phi_{\xi=\eta}$ which has to enter in $\mathcal{D}_{\xi=\eta,1}$, topology $\mathcal{D}_{\xi=\eta,1}$ already satisfies the axioms (T_I), (T_{II}), and (T_{III}). Thus, $\mathcal{D}_{\xi=\eta,1} = \{\emptyset; \varphi_1''; \Phi_{\xi=\eta}\}$.

(2) Let us study topology $\mathcal{D}_{\xi=\eta,2}$ in which the basic transition systems, covering the loops in Fig. 4, are included. Thus,

$$\begin{aligned} &(\lambda_1; \lambda_2; \lambda_3; \lambda_4; \lambda_5; \lambda_6; \mu_1), \\ &(\lambda_2; \lambda_3; \lambda_4; \mu_2), (\lambda_4; \lambda_5; \lambda_6; \mu_3), \\ &(\lambda_2; \mu_4); (\lambda_4; \mu_5); (\lambda_6; \mu_6) \in \mathcal{D}_{\xi=\eta,2} \end{aligned}$$

This choice of topological subsystems causes the inclusion of further subsystems. By the intersection axiom (T_{II}), there is

$$(\lambda_2; \lambda_3; \lambda_4), (\lambda_4; \lambda_5; \lambda_6), (\lambda_2), (\lambda_4), (\lambda_6) \in \mathcal{D}_{\xi=\eta,2}$$

Evidently, these subsystems represent the common parts of the loops. By the union axiom (T_I), elements

like

$$\begin{aligned} &(\lambda_1; \dots; \lambda_6; \mu_1; \mu_2; \mu_3; \mu_4; \mu_5; \mu_6), \\ &(\lambda_1; \dots; \lambda_6; \mu_1; \mu_2; \mu_3; \mu_4; \mu_5), \\ &(\lambda_1; \dots; \lambda_6; \mu_1; \mu_2; \mu_4; \mu_5; \mu_6), \\ &(\lambda_1; \dots; \lambda_6; \mu_1; \mu_3; \mu_4; \mu_5; \mu_6), \\ &(\lambda_1; \dots; \lambda_6; \mu_2; \mu_3; \mu_4; \mu_5; \mu_6), \\ &(\lambda_1; \dots; \lambda_6; \mu_1; \mu_2; \mu_3; \mu_4), \\ &(\lambda_1; \dots; \lambda_6; \mu_1; \mu_2; \mu_3), \\ &(\lambda_1; \dots; \lambda_6; \mu_1; \mu_2), \\ &\vdots \\ &(\lambda_1; \dots; \lambda_6; \mu_5; \mu_6) \end{aligned}$$

etc. must additionally enter topology $\mathcal{D}_{\xi=\eta,2}$. Now, again axiom (T_{II}) has to be applied, etc. The number of elements in $\mathcal{D}_{\xi=\eta,2}$ becomes enormous.

5.2 Topologies of a Bidirectional Metaphysicalism

Bidirectionality in informational sense means introducing a strict counterdirectional path (reverse serial or circular serial formula) in regard to the existing path (initial formula). In a graph, this situation is evidently visible by the occurrence of counterarrows or, in some cases, by the operand connection lines with arrows on both sides of the line.

The graph in Fig. 7 represents a conceptually invariant shell of the possible bidirectional artificial consciousness. Bidirectionality is ensured in every point of the informational structure. Further, the graph can be used as a template for any formula system development on one side, and as an individual semantic approach to the choice of vertical components in several specific domains of the informational, that is, of the conscious individualism, its structure and organization on the other side. A suggestion for the choice of vertical components is given in [36, 38]. Thus, vertical components can fit best the specific field of research

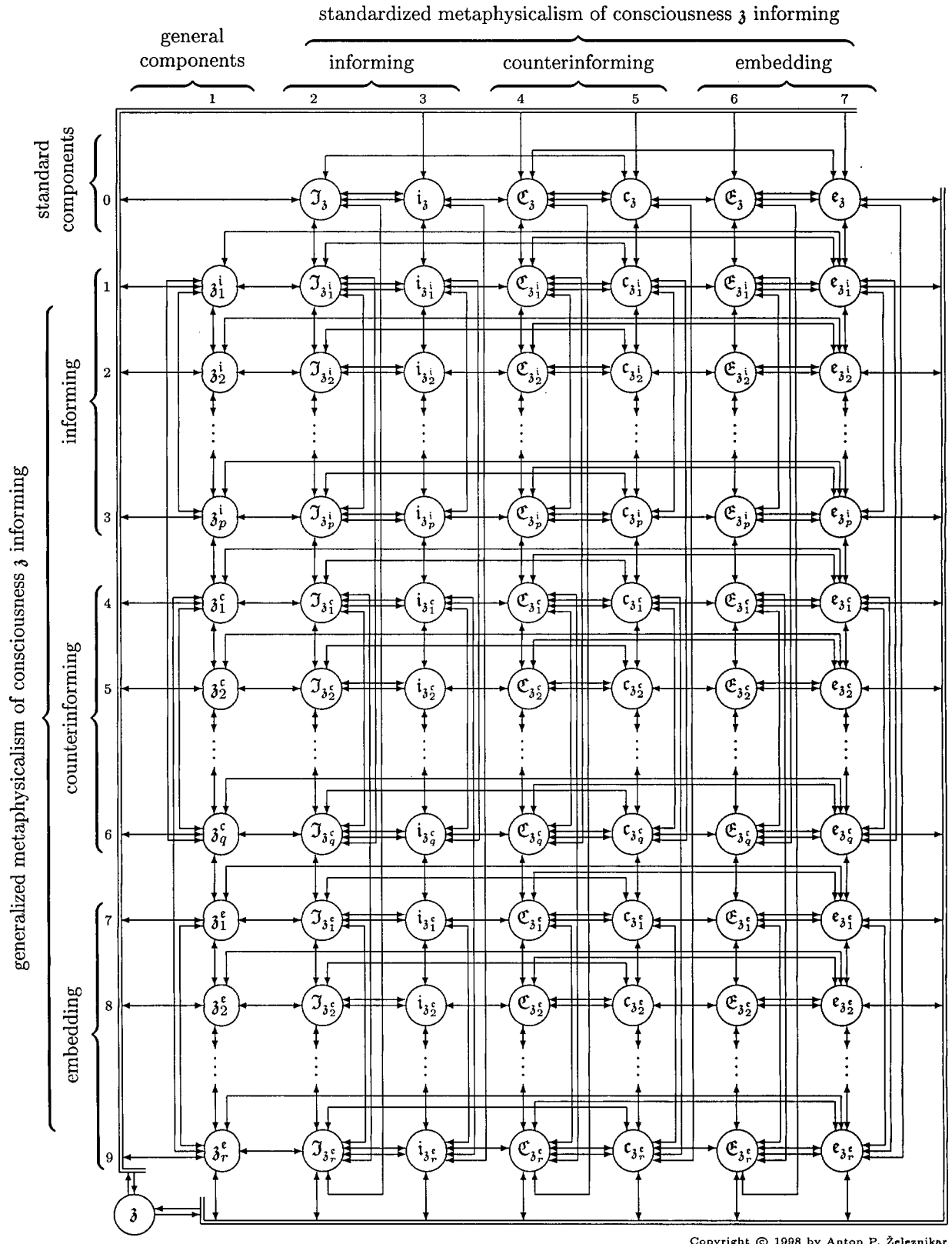


Figure 7: An initial informational shell of the generalized and standardized metaphysicalism of consciousness system \mathfrak{z} (a kind of pure consciousness), exploring the bidirectional metaphysicalism.

in respect to the function in the vertical metaphysical scheme. On the other side, chosen vertical components can be again metaphysically decomposed in the horizontal direction.

In the framework of consciousness circumstances, the stream of consciousness can be forced consciously into the opposite direction within informational cycles as shown in Fig. 6. A critical conscious informing must investigate its own conscious stream (of informing, counterinforming, and informational embedding) in one and the other direction, changing the causal conditions circularly in the opposite direction. In an unidirectional graph, each arrow, representing an operator, is replaced by the bidirectional arrow, representing two operators, the direct and the reverse one. Thus, in Fig. 7, a bidirectional arrow \longleftrightarrow , marks \rightleftarrows , meaning two separate and functionally (essentially) different operands.

Topologically, each concrete case concerning the graph in Fig. 6 can be informationally distinguished, for example, by a definite setting of the parenthesis pairs in formulas. As mentioned frequently before, the formula system Φ_φ is the originally conceptualized model of a real informational situation. In this sense, bidirectionality offers the possibility to investigate a loop in one and the opposite direction. For a loop, for instance, the principles of the pure observing and the pure informing can be applied in one and the opposite direction, simultaneously. For the case in Fig. 6, there is, for example,

$${}_{1,1}\varphi_{=}^{\circ}[\xi_{ij}, \alpha_j] \rightleftharpoons \left(\begin{array}{ll} \alpha_j \models \xi_{ij}; & [\varphi_0] \\ \((((\xi_{ij} \models \mathcal{J}_{\xi_{ij}}) \models i_{\xi_{ij}}) \models \mathcal{E}_{\xi_{ij}}) \models c_{\xi_{ij}}) \models & [\varphi_1] \\ \mathcal{E}_{\xi_{ij}} \models e_{\xi_{ij}}) \models \xi_{ij}; & \\ \xi_{ij} \models (e_{\xi_{ij}} \models (\mathcal{E}_{\xi_{ij}} \models (c_{\xi_{ij}} \models (\mathcal{C}_{\xi_{ij}} \models & [\varphi_1^{\leftarrow}] \\ (i_{\xi_{ij}} \models (\mathcal{J}_{\xi_{ij}}) \models \xi_{ij}))))); & \\ (((\mathcal{J}_{\xi_{ij}} \models i_{\xi_{ij}}) \models \mathcal{C}_{\xi_{ij}}) \models c_{\xi_{ij}}) \models \mathcal{J}_{\xi_{ij}}; & [\varphi_2] \\ \mathcal{J}_{\xi_{ij}} \models (c_{\xi_{ij}} \models (\mathcal{E}_{\xi_{ij}} \models (i_{\xi_{ij}} \models \mathcal{J}_{\xi_{ij}}))); & [\varphi_2^{\leftarrow}] \\ (((\mathcal{C}_{\xi_{ij}} \models c_{\xi_{ij}}) \models \mathcal{E}_{\xi_{ij}}) \models e_{\xi_{ij}}) \models \mathcal{C}_{\xi_{ij}}; & [\varphi_3] \\ \mathcal{C}_{\xi_{ij}} \models (e_{\xi_{ij}} \models (\mathcal{E}_{\xi_{ij}} \models (c_{\xi_{ij}} \models \mathcal{C}_{\xi_{ij}}))); & [\varphi_3^{\leftarrow}] \\ (\mathcal{J}_{\xi_{ij}} \models i_{\xi_{ij}}) \models \mathcal{J}_{\xi_{ij}}; & [\varphi_4] \\ \mathcal{J}_{\xi_{ij}} \models (i_{\xi_{ij}} \models \mathcal{J}_{\xi_{ij}}); & [\varphi_4^{\leftarrow}] \\ (\mathcal{C}_{\xi_{ij}} \models c_{\xi_{ij}}) \models \mathcal{C}_{\xi_{ij}}; & [\varphi_5] \\ \mathcal{C}_{\xi_{ij}} \models (c_{\xi_{ij}} \models \mathcal{C}_{\xi_{ij}}); & [\varphi_5^{\leftarrow}] \\ (\mathcal{E}_{\xi_{ij}} \models e_{\xi_{ij}}) \models \mathcal{E}_{\xi_{ij}} & [\varphi_6] \\ \mathcal{E}_{\xi_{ij}} \models (e_{\xi_{ij}} \models \mathcal{E}_{\xi_{ij}}) & [\varphi_6^{\leftarrow}] \end{array} \right)$$

This is the original (initial) formula system, a consequently observing case in each system formula, from which the graph in Fig. 6 was drawn, consistently following the rule of an arrow and its counterarrow. It is clear that according to a specific informational case, the parenthesis pairs can be set adequately (and dif-

ferently), following the realistic circumstances for each of the system formula. However, any other setting of the parenthesis pairs in the system formulas does not change the informational graph in Fig. 6. Maybe, in a specific case, some direct and/or reverse paths can even be omitted or left simply void for a later final decision.

(0) Let us denote $\Phi_\varphi^{\rightleftharpoons} \rightleftharpoons_{1,1} \varphi_{=}^{\circ}$. Now, for the sake of transparency, let be

$$\Phi_\varphi^{\rightleftharpoons} \rightleftharpoons \left(\begin{array}{l} \varphi_0; \\ \varphi_1; \varphi_1^{\leftarrow}; \\ \varphi_2; \varphi_2^{\leftarrow}; \\ \varphi_3; \varphi_3^{\leftarrow}; \\ \varphi_4; \varphi_4^{\leftarrow}; \\ \varphi_5; \varphi_5^{\leftarrow}; \\ \varphi_6; \varphi_6^{\leftarrow} \end{array} \right)$$

representing the input transition formula φ_0 (bringing into the system the exterior object α at point ξ_{ij}) and the twelve circular formulas.

(1) To represent the variability of a formula system rooting in the possibility of arbitrary parenthesis pairs displacements in formulas, we can use the formal expression of informational schemes (the so-called graph routes of graph paths) for Fig. 6, and write the graph equivalent scheme in regard to the initial system $\Phi_\varphi^{\rightleftharpoons}$ in the form

$$\Phi_\varphi^{\rightleftharpoons} \rightleftharpoons \left(\begin{array}{ll} \alpha_j \models \xi_{ij}; & [\varphi_0] \\ \xi_{ij} \models \mathcal{J}_{\xi_{ij}} \models i_{\xi_{ij}} \models \mathcal{E}_{\xi_{ij}} \models c_{\xi_{ij}} \models & [\varphi_1] \\ \mathcal{E}_{\xi_{ij}} \models e_{\xi_{ij}} \models \xi_{ij}; & \\ \xi_{ij} \models e_{\xi_{ij}} \models \mathcal{E}_{\xi_{ij}} \models c_{\xi_{ij}} \models \mathcal{C}_{\xi_{ij}} \models & [\varphi_1^{\leftarrow}] \\ i_{\xi_{ij}} \models \mathcal{J}_{\xi_{ij}} \models \xi_{ij}; & \\ \mathcal{J}_{\xi_{ij}} \models i_{\xi_{ij}} \models \mathcal{C}_{\xi_{ij}} \models c_{\xi_{ij}} \models \mathcal{J}_{\xi_{ij}}; & [\varphi_2] \\ \mathcal{J}_{\xi_{ij}} \models c_{\xi_{ij}} \models \mathcal{E}_{\xi_{ij}} \models i_{\xi_{ij}} \models \mathcal{J}_{\xi_{ij}}; & [\varphi_2^{\leftarrow}] \\ \mathcal{C}_{\xi_{ij}} \models c_{\xi_{ij}} \models \mathcal{E}_{\xi_{ij}} \models e_{\xi_{ij}} \models \mathcal{C}_{\xi_{ij}}; & [\varphi_3] \\ \mathcal{C}_{\xi_{ij}} \models e_{\xi_{ij}} \models \mathcal{E}_{\xi_{ij}} \models c_{\xi_{ij}} \models \mathcal{C}_{\xi_{ij}}; & [\varphi_3^{\leftarrow}] \\ \mathcal{J}_{\xi_{ij}} \models i_{\xi_{ij}} \models \mathcal{J}_{\xi_{ij}}; & [\varphi_4] \\ \mathcal{J}_{\xi_{ij}} \models i_{\xi_{ij}} \models \mathcal{J}_{\xi_{ij}}; & [\varphi_4^{\leftarrow}] \\ (\mathcal{C}_{\xi_{ij}} \models c_{\xi_{ij}}) \models \mathcal{C}_{\xi_{ij}}; & [\varphi_5] \\ \mathcal{C}_{\xi_{ij}} \models c_{\xi_{ij}} \models \mathcal{C}_{\xi_{ij}}; & [\varphi_5^{\leftarrow}] \\ (\mathcal{E}_{\xi_{ij}} \models e_{\xi_{ij}}) \models \mathcal{E}_{\xi_{ij}} & [\varphi_6] \\ \mathcal{E}_{\xi_{ij}} \models e_{\xi_{ij}} \models \mathcal{E}_{\xi_{ij}} & [\varphi_6^{\leftarrow}] \end{array} \right)$$

In this formula system scheme, some directed and counterdirected paths obtain equal formal expression, e.g., $[\varphi_4]$ and $[\varphi_4^{\leftarrow}]$, $[\varphi_5]$ and $[\varphi_5^{\leftarrow}]$, and $[\varphi_6]$

and $\boxed{[\varphi_6^{\leftarrow}]}$ seem to be equal. However, it is to understand that they originate from different informational situations and, according to the original circumstances, they have different operators between the equal operands²².

(2) An interesting case occurs in dealing with the operand system

$$\Phi_{\xi_{ij}}^{\rightleftarrows} = \left(\alpha; \xi_{ij}^{\rightleftarrows}; \mathcal{J}_{\xi_{ij}}^{\rightleftarrows}; i_{\xi_{ij}}^{\rightleftarrows}; \mathcal{C}_{\xi_{ij}}^{\rightleftarrows}; c_{\xi_{ij}}^{\rightleftarrows}; \mathcal{E}_{\xi_{ij}}^{\rightleftarrows}; e_{\xi_{ij}}^{\rightleftarrows} \right)$$

concerning the graph in Fig. 6 and with possible topologies on this system.

First, let us explain in which way the bidirectional operands $\xi_{ij}^{\rightleftarrows}; \mathcal{J}_{\xi_{ij}}^{\rightleftarrows}; i_{\xi_{ij}}^{\rightleftarrows}; \mathcal{C}_{\xi_{ij}}^{\rightleftarrows}; c_{\xi_{ij}}^{\rightleftarrows}; \mathcal{E}_{\xi_{ij}}^{\rightleftarrows}; e_{\xi_{ij}}^{\rightleftarrows}$ are formally and explicitly represented. Fig. 6 shows how many causal circular paths (loops) pass a certain operand. The following correspondence is evident:

$$\begin{aligned} \xi_{ij}^{\rightleftarrows} \blacktriangleright 2; \mathcal{J}_{\xi_{ij}}^{\rightleftarrows} \blacktriangleright 6; i_{\xi_{ij}}^{\rightleftarrows} \blacktriangleright 6; \mathcal{C}_{\xi_{ij}}^{\rightleftarrows} \blacktriangleright 8; \\ c_{\xi_{ij}}^{\rightleftarrows} \blacktriangleright 8; \mathcal{E}_{\xi_{ij}}^{\rightleftarrows} \blacktriangleright 6; e_{\xi_{ij}}^{\rightleftarrows} \blacktriangleright 6 \end{aligned}$$

Operator \blacktriangleright reads *directly informs the number of loops*.

How, for instance, operand $\mathcal{C}_{\xi_{ij}}^{\rightleftarrows}$ is expressed explicitly by means of the loops passing it, using the so-called operand rotation principle for each of particular loop, and the informational path (scheme) form? The advantage of the path formula is that the setting of the parenthesis pairs remains open, and in this case various possibilities of the final setting of parenthesis pairs can be considered. Evidently, the following comes out from the graph in Fig. 6:

$$\boxed{\mathcal{C}_{\xi_{ij}}^{\rightleftarrows}} = \left(\begin{array}{l} \mathcal{C}_{\xi_{ij}} \models c_{\xi_{ij}} \models \mathcal{E}_{\xi_{ij}} \models e_{\xi_{ij}} \models \xi_{ij} \models \mathcal{J}_{\xi_{ij}} \models i_{\xi_{ij}} \models \mathcal{C}_{\xi_{ij}}; \\ \mathcal{C}_{\xi_{ij}} \models i_{\xi_{ij}} \models \mathcal{J}_{\xi_{ij}} \models \xi_{ij} \models e_{\xi_{ij}} \models \mathcal{E}_{\xi_{ij}} \models c_{\xi_{ij}} \models \mathcal{C}_{\xi_{ij}}; \\ \mathcal{C}_{\xi_{ij}} \models c_{\xi_{ij}} \models \mathcal{J}_{\xi_{ij}} \models i_{\xi_{ij}} \models \mathcal{C}_{\xi_{ij}}; \\ \mathcal{C}_{\xi_{ij}} \models i_{\xi_{ij}} \models \mathcal{J}_{\xi_{ij}} \models c_{\xi_{ij}} \models \mathcal{E}_{\xi_{ij}}; \\ \mathcal{C}_{\xi_{ij}} \models c_{\xi_{ij}} \models \mathcal{E}_{\xi_{ij}} \models e_{\xi_{ij}} \models \mathcal{C}_{\xi_{ij}}; \\ \mathcal{C}_{\xi_{ij}} \models e_{\xi_{ij}} \models \mathcal{E}_{\xi_{ij}} \models c_{\xi_{ij}} \models \mathcal{C}_{\xi_{ij}}; \\ \mathcal{C}_{\xi_{ij}} \models c_{\xi_{ij}} \models \mathcal{C}_{\xi_{ij}}; \\ \mathcal{C}_{\xi_{ij}} \models c_{\xi_{ij}} \models \mathcal{C}_{\xi_{ij}} \end{array} \right) \begin{array}{l} \boxed{[\varphi_1^e]} \\ \boxed{[\varphi_2^e]} \\ \boxed{[\varphi_3^e]} \\ \boxed{[\varphi_4^e]} \\ \boxed{[\varphi_5^e]} \\ \boxed{[\varphi_6^e]} \\ \boxed{[\varphi_7^e]} \\ \boxed{[\varphi_8^e]} \end{array}$$

²²Operator \models denotes a general informational joker. In two cases, the equal transition formulas $\alpha \models \beta$ and $\alpha \models \beta$ can represent different transitions. For instance, between two substantives different verb forms can be set. It means that in virtually equal formal cases, the intention of α 's informing follows the first and then the second verbal form. Finally, the cases are resolved as being different by the particularization of operators.

The last two paths are virtually equivalent (see the footnote ²²). Similar schemata can be obtained from the graph in Fig. 6 for the remaining operand systems $\boxed{\xi_{ij}^{\rightleftarrows}}$, $\boxed{\mathcal{J}_{\xi_{ij}}^{\rightleftarrows}}$, $\boxed{i_{\xi_{ij}}^{\rightleftarrows}}$, $\boxed{c_{\xi_{ij}}^{\rightleftarrows}}$, $\boxed{\mathcal{E}_{\xi_{ij}}^{\rightleftarrows}}$, and $\boxed{e_{\xi_{ij}}^{\rightleftarrows}}$. The rule for an explicit expression of an operand out of given formula system is to collect all the formulas in which the operand occurs and then express these circular formulas, according to the principle of an operand rotation, in a way by which the operand comes to the title position (the most left and the most right position in a circular formula).

What can then be said to the topological outlook of the obtained framed operand (in fact, a system of informational paths) representing formula a system by each of the system path? It is to stress that the graph for the formula system scheme $\boxed{\mathcal{C}_{\xi_{ij}}^{\rightleftarrows}}$ (with 8 formula paths) is a subgraph of the graph in Fig. 6 (merely the local informing and embedding loops are missing).

In this sense we introduce a new concept of topology consisting of informational paths (routes, marked by ρ) instead of informational formulas φ , representing $\rho = \boxed{[\varphi]}$. Thus, instead of Φ_φ we introduce Φ_ρ or $\Phi_{\boxed{[\varphi]}}$, respectively. Each path (graph route) ρ represents potentially $\frac{1}{\ell_\rho} \binom{2\ell_\rho}{\ell_\rho}$ formulas if ℓ_ρ is the length of the path corresponding formula (number of the adequate formula binary operators). In this way a new sort of topological space is introduced, for instance pertaining to $\mathcal{C}_{\xi_{ij}}^{\rightleftarrows}$,

$$\left\langle \Phi_{\rho,1}^{\mathcal{C}_{\xi_{ij}}^{\rightleftarrows}}, \mathcal{D}_{\rho,1}^{\mathcal{C}_{\xi_{ij}}^{\rightleftarrows}} \right\rangle, \text{ where}$$

$$\Phi_{\rho,1}^{\mathcal{C}_{\xi_{ij}}^{\rightleftarrows}} = \left(\boxed{[\varphi_1^e]}; \boxed{[\varphi_2^e]}; \dots; \boxed{[\varphi_8^e]} \right), \text{ and, e.g.,}$$

$$\mathcal{D}_{\rho,1}^{\mathcal{C}_{\xi_{ij}}^{\rightleftarrows}} = \left(\begin{array}{l} \emptyset; \\ \boxed{[\varphi_1^e]}; \boxed{[\varphi_2^e]}; \\ \boxed{[\varphi_1^e]}; \boxed{[\varphi_2^e]}; \\ \Phi_{\rho,1}^{\mathcal{C}_{\xi_{ij}}^{\rightleftarrows}} () \end{array} \right)$$

Evidently, $\Phi_{\rho,1}^{\mathcal{C}_{\xi_{ij}}^{\rightleftarrows}} = \mathcal{C}_{\xi_{ij}}^{\rightleftarrows}$.

(3) One could construct other reasonable topologies being subsystems of $\mathfrak{P} \left[\Phi_{\rho,1}^{\mathcal{C}_{\xi_{ij}}^{\rightleftarrows}} \right]$. But, the next provoking question concerns a topology of formula systems Φ_φ (not just formulas φ) and topologies of topological spaces of the form $\langle \Phi, \mathcal{D} \rangle$.

Let Φ_{Φ_φ} mark a system of formula systems Φ_φ and $\Phi_{\langle \Phi_\varphi, \mathcal{D}_\varphi \rangle}$ a system of topological spaces $\langle \Phi_\varphi, \mathcal{D}_\varphi \rangle$, in general. Let

$$\Phi_\varphi \in \Phi_{\Phi_\varphi} \text{ and } (\Psi \in \mathcal{D}_{\Phi_\varphi}) \implies (\Psi \subset \Phi_{\Phi_\varphi})$$

This structure delivers a system topological space $\langle \Phi_{\varphi}, \mathcal{D}_{\varphi} \rangle$.

Another concept of topology of topological spaces follows the condition

$$\langle \Phi_{\varphi}, \mathcal{D}_{\varphi} \rangle \in \Phi_{\langle \Phi_{\varphi}, \mathcal{D}_{\varphi} \rangle} \text{ and } (\Psi \in \mathcal{D}_{\langle \Phi_{\varphi}, \mathcal{D}_{\varphi} \rangle}) \implies (\Psi \subset \Phi_{\langle \Phi_{\varphi}, \mathcal{D}_{\varphi} \rangle})$$

delivering a topology topological space of the form $\langle \Phi_{\langle \Phi_{\varphi}, \mathcal{D}_{\varphi} \rangle}, \mathcal{D}_{\langle \Phi_{\varphi}, \mathcal{D}_{\varphi} \rangle} \rangle$.

5.3 Topological Informational Spaces Possessing Informational Metrics

What kinds of informational metrics could come to the surface, could be considered, and finally theoretically (constructively) applied in artificial systems of consciousness and other cognitive models? Which are the possibilities of introducing various kinds of metrics concepts—the informationally static²³ and informationally dynamic²⁴ ones—into a topologically structured informational space?

Several candidates come into consideration as measures of the informational metrics. The properties of such measures could be, for instance, meaningfulness, understandingness, interpretativeness, perceptiveness, conceptiveness, determinativeness, and several others. If so, the corresponding decomposition and expressiveness of informational measures as entities must be available.

Where could these measures reside within a metaphysicalistic model? The answer is, anywhere. By the principle of operand rotation in a circular formula, any loop operand can be rotated to the initial (main) position of the loop and, by this, expressed by an adequate informational formula in respect to the parenthesis-pair setting in the formula. Meaning of something as an informational measure can usually appear in the embedding part of a metaphysicalistic loop. As a meaning of something it could represent the informational value (informational length) of something. In a similar manner, the informational distance between two informational operands could be determined, implicitly and explicitly, by a functionally inner and outer informational difference, respectively.

Various concepts of understanding, conception, perception, etc. can serve as special measures of meaning (metrics). They can be placed constructively in any part of the metaphysicalistic loop and, then, rotated to

²³By an informationally static metrics, the most common concepts of informing are meant, for instance, that of something's meaning. Typical, purely static metrics concerns numerical or any other value, distance, or any other geometrical measure.

²⁴By informationally dynamic metrics, the individually organized informational phenomena are meant, for instance that of an individual understanding structure, which has something in common with the individual structures of others, and which is to some extent structured invariantly (standardized) in concern to the meaning or understanding.

the main position of a formula and expressed explicitly [32]. This kind of constructive approach must remain within the reasonable limits, preserving the common logical principles or direction.

6 Possible Geometry and Topology of the Informational

That what will be stressed in this section concerns the interpretation possibilities of informational topologies by means of geometric bodies—their surfaces, intersections, volumes, and arbitrary substructures occurring interiorly, on the surface, and/or exteriorly of these bodies. Interpretation ideas can be found in several sources dealing with geometry [9, 22, 23, 26, 27]. Mathematica [9] seems to be the tool for an adequate graphical presentation.

By such an interpretation of systems of informational formulas, geometrical bodies become also a means for informationally semantic presentation of modeled entities. For instance, a sphere—its interior, surface and exterior—can be taken as a body of consciousness (or a body of any other informational entity). The surface of the sphere can represent topologically that which is potentially possible to become conscious, and a circle on the sphere surface can represent the currently conscious. Such circles can expand as parts of different toruses which intersect with the sphere. They can represent different intentional informings within the consciousness activity.

Further, the interior of the sphere can represent the subconscious which can come to the surface. On contrary, the exterior of the sphere can be grasped as the non-conscious and non-subconscious yet. Thus, a system of spheres and toruses intersecting each other can built a complex and to some degree globally transparent model of interacting consciousness systems.

Such a complex geometrical model can be particularly, that is, additionally, characterized with specific topologies, bringing into the modeling system an interaction of different topological spaces. In this context, both informational topologies and geometrical bodies can become a reasonable unit for complex informational investigation and experiments in the domain of the informational, and particularly in the domain of the conscious in an informational sense.

Geometrically interpreted, informational topological spaces of informational topological spaces could get a transparent view to an arbitrary (recurrent) depth. Further, such interpreting geometrical structures can behave variable in any possible aspect, for instance, in moving of geometrical body intersections together with bodies which can change also dimensions (volumes, radii, sides, surfaces) to follow the dynamic picture of informational circumstances emerging, changing, and vanishing. Such problems of informational

and consciousness geometry interpretation deserve a special attention and will be treated somewhere else.

7 Conclusion

We see how the concept of mathematical topology comes intuitively close to the informational topology. However, the substantial differences occurring between them, e.g. the nature of emergence of operands, operators, and formula systems, have to be stressed over and over again. Some of the differences are already recognized from the mathematical-informational dictionary in Sect. 2, and other follow from the discussion and examples in this paper. It is worth to refresh these differences by the following list:

1. A formula system is obviously a set of interdependent formulas, irrespective, how it is expressed; e.g., by (1) serial circular formulas of different lengths, (2) primitive transition formulas, or (3) informational operands that are in some way, by some specific formula systems given on some other places.
2. Formulas (elements) of an informational formula system are directly dependent on each other through the common operands. Thus, the change of an operand in a given formula changes the same operand in an other formula and, thus, changing the informing of the other formula. As said, the interdependence of formulas as system elements is a rule, that is, a consequence of their formal linkage through common formula operands. In this respect, informational formulas as system elements behave differently in respect to the elements of a mathematical set.
3. A consequence of the preceding item is that elements of a set are meant as a sort of constant determined entities, and are in this way represented as (fixed) set elements. On the other hand, formulas as system elements possess their emerging nature in any respect: in emerging operands and operators, in setting of parenthesis pairs in a formula, and, most significantly, in expanding or contracting a formula by the number of occurring operands and operators, that is, in spreading and narrowing the meaning power of a formula.
4. A concrete formula system can also emerge according to the circumstances of its informing, for instance, by adding the interpretational formulas concerning the occurring operands, expressing the operand properties by additional (new) formulas. On the other side, a concrete mathematical set is defined constantly, even its cardinality is infinite. The elements of a set are determined by an unchangeable rule (e.g., predicate) or by a sort of concrete or recursive enumeration.

By informational topology, a complex meaningfully structured grouping and coupling of formulas concerning substantial informational spaces can formally be expressed (implemented), keeping the entire, that is, a non-reductional informational nature of involved entities as they perform in their reality. In this respect, a topologized formula system is not a simplified model for real informational situations, for instance in the domain of cognitive science²⁵.

Tangled webs of causal influences are target phenomena in recent biology and cognitive science [10]. Such twisted influences include both internal and external factors as well as patterns of reciprocal (also bidirectional) interaction. The shell graph in Fig. 7 is a general scheme for the most pretentious informational modeling and experimenting, where the so-called reductionist approach can be entirely circumvented. Such an initial informational shell can be used as an informing model for any other problems beside consciousness (e.g., in philosophy, cognitive science, biology, psychology, psychiatry, language, on-line economic simulation, etc., as shown in [32, 37] where additional references are listed.). This points evidently to the applicability of informational topology with its deep intuitive background being appropriate for natural and artificial modeling of interactive philosophical and scientific problems.

An evident example of the informational metaphysicalism could be the so-called inner speech (talking to oneself) [3]. Such a speech is constituted by the experienced meaning (informing), emergence of speech (counterinforming), and logical articulation (informational embedding), respectively. But, all components of this sort can emerge in a distributed form across the inner speech informing. They can be treated (grasped, understood) topologically as certain informational or semantical unity through topological grouping by subsystems $\sigma_\varphi \in \mathcal{D}_{\Phi_\varphi}$, where $\sigma_\varphi \subset \Phi$ and $(\Phi, \mathcal{D}_{\Phi_\varphi})$ is the corresponding topological informational space.

References

- [1] ATTRIDGE, D. 1992. Derrida and the question of literature. *In* J. Derrida, *Acts of Literature*. Routledge. New York.
- [2] BALAKRISHNAN, V.K. 1995. *Combinatorics (including concepts of Graph Theory)*. McGraw-Hill. New York.
- [3] BLACHOWICZ, J. 1997. The dialogue of the soul with itself. *Journal of Consciousness Studies* 4:485-508.

²⁵In contrast to artificial neural networks (ANNs) being *simplified mathematical models for neural systems formed by massively interconnected computational units running in parallel* [28], informational topological systems can always fit adequately and arbitrarily precisely the point of meaning in a topological continuum.

- [4] BONNINGTON, C.P. & C.H.C. LITTLE. 1995. *The Foundations of Topological Graph Theory*. Springer-Verlag. New York, Berlin, Heidelberg.
- [5] BOURBAKI, N. 1960–1966. *Théorie des ensembles*. Chapitres 1,2 ,3 et 4. Hermann. Paris.
- [6] Бурбаки, Н. 1965. *Теория множеств*. Издательство МИР. Москва.
- [7] BOURBAKI, N. 1965. *Topologie générale*. Chapitres 1 et 2. Hermann. Paris.
- [8] Бурбаки, Н. 1968. *Общая топология. Основные структуры*. Наука. Физматгиз. Москва.
- [9] BOYLAND, P. 1991. *Guide to Standard Mathematica Packages*. Wolfram Research.
- [10] CLARK, A. 1998. Twisted tales: causal complexity and cognitive scientific explanation. *Minds and Machines* 8:79–99.
- [11] DERRIDA, J. 1988. Signature event context. *In* G. Graff, Ed. Limited Inc. Northwestern University Press. Evanston.
- [12] FRÉCHET, M. 1906. Sur quelque points du calcul fonctionnel. *Rend. Palermo* 22:1–74 .
- [13] FRANZ, W. 1960. *Topologie I. Allgemeine Topologie*. Sammlung Götschen. Band 1181. Walter de Gruyter & Co. Berlin.
- [14] GROSS, J.L. & T.W. TUCKER. 1987. *Topological Graph Theory*. J. Wiley. New York.
- [15] HANEY, W.S. 1998. Deconstruction and consciousness: the question of unity. *Journal of Consciousness Studies* 5:19–33.
- [16] HOCKING, J.G. & G.S. YOUNG. 1961. *Topology*. Addison-Wesley. Reading, MA. London.
- [17] HUSSERL, E. 1950. *Ideen zu einer reinen Phänomenologie und phänomenologischen Philosophie*. Husserliana III (W. Biemel). Martinus Nijhoff. Haag.
- [18] HU, SZE-TSEN. 1964. *Elements of General Topology*. Holden-Day. San Francisco. London. Amsterdam.
- [19] KELLEY, J.L. 1955. *General Topology*. Springer-Verlag. New York. Heidelberg. Berlin.
- [20] LIPSCHUTZ, S. 1964. *Set Theory*. Schaum. McGraw-Hill. New York.
- [21] LIPSCHUTZ, S. 1965. *General Topology*. Schaum. New York.
- [22] MAEDER, R.E. 1990. *Programming in Mathematica*. Addison-Wesley. Redwood City, CA.
- [23] PRASSOLOV, V. 1995. *Topologie in Bildern*. Verlag Harri Deutsch. Thun, Frankfurt am Main.
- [24] PRIJATELJ, N. 1985. *Mathematical Structures III. Neighbourhoods*. Državna založba Slovenije. Ljubljana. In Slovene.
- [25] THRON, W.J. 1966. *Topological Structures*. Holt, Rinehart and Winston. New York. London.
- [26] TÓTH, L.F. 1965. *Reguläre Figuren*. B.G. Teubner. Leipzig.
- [27] WELLS, D. 1991. *The Penguin Dictionary of Curious and Interesting Geometry*. Illustrated by J. Sharp. Penguin Books. London.
- [28] YANG, H.H., N. MURATA & S. AMARI. 1998. Statistical inference: learning in artificial neural networks. *Trends in Cognitive Sciences* 2:1:4–10.
- [29] ŽELEZNIKAR, A. P. 1994. Informational Being-of. *Informatica* 18:277–298.
- [30] ŽELEZNIKAR, A.P. 1995. Elements of metamathematical and informational calculus. *Informatica* 19:345–370.
- [31] ŽELEZNIKAR, A.P. 1996. Informational frames and gestalten. *Informatica* 20:65–94.
- [32] ŽELEZNIKAR, A.P. 1996. Organization of informational metaphysicalism. *Cybernetica* 39:135–162.
- [33] ŽELEZNIKAR, A.P. 1996. Informational transition of the form $\alpha \models \beta$ and its decomposition. *Informatica* 20:331–358.
- [34] ŽELEZNIKAR, A.P. 1997. Zum formellen Verstehen des Informationsphänomenalismus. *Grundlagenstudien aus Kybernetik und Geisteswissenschaft/Humankybernetik* 38:3–14.
- [35] ŽELEZNIKAR, A.P. 1997. Informational graphs. *Informatica* 21:79–114.
- [36] ŽELEZNIKAR, A.P. 1997. Informational theory of consciousness. *Informatica* 21:345–368.
- [37] ŽELEZNIKAR, A.P. 1997. Informationelle Untersuchungen. *Grundlagenstudien aus Kybernetik und Geisteswissenschaft/Humankybernetik* 38:147–158.
- [38] ŽELEZNIKAR, A.P. 1997. Informational consciousness. *Cybernetica* 40:261–296.
- [39] Зыков, А.А. 1969. *Теория конечных графов I*. Издательство *Наука*. Сибирское отделение. Новосибирск.

Control Mechanisms for Assuring Better IS Quality

Marjan Pivka
 University of Maribor
 School of Business and Economics Maribor
 Razlagova 14, 62000 Maribor, Slovenia
 Tel: ++ 386 62 2290247, Fax: ++ 386 62 26 681
 Email: pivka@uni-mb.si

Keywords: software quality, IS manager, business

Edited by: Janez Grad

Received: August 28, 1997

Revised: July 17, 1998

Accepted: August 25, 1998

The software domain is faced with a number of quality assurance and process improvement models. Business managers are under pressure from many different kinds of assessments for their operations, products and services. Accounting departments are audited by financial auditors. What about Information Systems? Do we have a universal model on how to achieve required IS quality? This paper deals with the definition of IS quality and the influence of different control mechanisms on IS. The results of this empirical research are several. First of all, none of the control mechanisms are universal and applicable to all IS resources. Application of more than one of them could be redundant. Mutual recognition of results between them is required. IS managers are responsible to understand them and use them with all limitations on specific IS resources.

1 Introduction

The computer based Information System (IS) uses hardware, software, telecommunications and other forms of Information Technology (IT) to transform data resources into a variety of information products. Enterprises need and use those products in their business processes to achieve business objectives. IT resources need to be managed in order to provide such information products to the enterprise.

Typical resources of IS are:

dataware computer data bases and other data resources

software computer programs, applications,...

lifeware human resources

hardware computers, communications and other office technology

orgware organisation, procedures etc.

In business we are constantly under pressure to reduce all kinds of expenses on the one hand and to improve quality on the other. One of these expenses are those for Information Systems and there are always some logical questions to be asked: is it good enough to justify the expense? Do we get what we need? Shall we invest in new IS? Is our IS reliable? Are results accurate? Those and other questions can be answered in a discussion on the quality of IS. Because quality

is what we all expect from IS: cover of all functional requirements, reliability, needed results given on time, usable for all users, maintainable etc.

It would be too easy to suppose that IS quality could be achieved by assuring a high quality of IS resources. IS is too complex and it grows with organisation and needs different control mechanisms in its maturity process. The high quality of technical resources of an IS, such as software or hardware, is by no means any guarantee for its high quality implementation in an improperly organized enterprise! And vice versa!

Different methods and principles (our term is *control mechanisms*) are known for IS quality assurance. Some of them control the IS development process, others IS resources and some may be used to control the development process and the implementation process. The best known control mechanisms today are:

- quality system standards (ISO 9000 family standards);
- software products standards;
- software process assessment models such as BOOTSTRAP [Haase et al 1993], CMM [Paulk et. all 1993] [CMM v2.0], ISO/SPICE [Rout P. Terence 1995] and many others [SPC 1997];
- IS Auditing.

It is difficult to understand those, and perhaps other, aggressively marketed control mechanisms.

Many have asked themselves: ISO 9000, CMM, SPICE, IS Auditing, BOOTSTRAP - what shall I do? Why does our financial auditor request an IS Audit, if we use a certified software product, or have an ISO 9001 certificate in software development? What shall I do? Which mechanism or model can help me? What are their strengths, what are their weaknesses? The answer is not universal or easy.

The aim of this paper is to analyse the influence of the most well known control mechanisms for IS (not just software!) quality. The paper is organized as follows. The second section introduces a formal definition of IS quality. Sections three to six deal with quality system standards, software product standards, software process assessment models and IS Auditing. Each control mechanism is briefly introduced and its strengths and weaknesses on IS quality are discussed. The discussion on collaboration, competition or conflicts between those control mechanisms is presented in section 7. Finally conclusions are given in section 8.

2 The formal definition of IS quality

The quality is defined as *totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs* [ISO/IEC 8402:1995]. According to this definition of IS, the quality of IS in general is totality of needed or implied quality characteristics of dataware, software, lifeware, hardware, and orgware. Or speaking more generally, the quality of IS is totality of quality of IS resources. Therefore, if we want to judge whether a given information system is reliable or not, accurate or not, efficient or inefficient, etc. we shall:

- define quality model i.e. quality characteristics based on required and implied needs,
- measure or assess each characteristic,
- compare the measured or assessed characteristics with the specific requirements, and
- validate the results.

In the field of engineering these procedures are well defined and known as evaluation models. In software engineering, such a model is defined in [ISO/IEC 9126:1991] and in [ISO/IEC DIS 14598:1996]. Real implementation of this evaluation model requires the practical solution of some very serious problems:

1. The definition of IS quality model. A quality model in general is a structure or composition of all quality characteristics of an entity. Thus, for IS quality model the quality characteristics and

their sub-characteristics shall be defined for each IS entity.

2. Each quality characteristic shall be decomposed to a measurable or assessable level. The metrics assessment method for each characteristic has to be defined.
3. User's, legal and or professional requirements (i.e. stated and implied needs) and their significance for (on) IS quality shall be defined for each quality characteristic.

The quality of IS is of course not a simple sum of the quality of each IS resource. The quality of an IS Q is a function of the stakeholders defined (stated and - or implied) IS characteristics (DC_i), actual values of those characteristics (AC_i), and by stakeholders defined influence of each characteristic (p_i) on IS:

$$Q = f(DC_i, AC_i, p_i).$$

Function f , and its arguments DC_i , AC_i and p_i depends on management & operations maturity of the organisation, type of organisation, its environment etc. for which the IS is intended. They all define general requirements such as the type of IS (Management IS, Decision support systems, Executive Support Systems, ...or any combination thereof), and of course very specific requirements such as Inputs, Outputs, Interfaces, Security requirements, Services etc.

This definition formally demonstrates, that quality of IS can not be achieved by high quality of the IS resources (AC_i). It can be achieved only if requirements (f , DC_i and p_i) are demonstrated with actual resources (AC_i). For instance: the implementation of a high quality and complex software product in an badly organized organization will result in a badly organized IS!

3 Influence of Quality System standards on IS quality

The quality system is defined as *organisational structure, procedures, processes and resources to implement quality management* [ISO/IEC 8402:1995]. The most popular international quality system standards are ISO 9000 family standards. ISO 9001:1994 defines a model for quality assurance in design, development, production, installation and servicing. It defines a number of quality requirements structured in 20 clauses, from management responsibility to statistical techniques. ISO 9000-3:1997 [ISO 9000-3:1997] are guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software. British TickIT [TickIT 1998] certification schema assures a thorough compilation of

ISO 9001 with ISO 9000-3 guidelines in software development processes.

The result of ISO 9001 application in software development is the defined, controlled and managed process of development, supply, installation and maintenance of computer software.

The strengths of implemented ISO 9001 in a software development process are:

- The management of software development process is focused on internationally acknowledged quality requirements defined by ISO 9001 which is also measurable. As such, management can monitor and compare quality characteristics of the software development process (productivity, efficiency, number of bugs, user complaints etc) with their plans and with their competitors.
- It covers the processes in a software development, acquisition and implementation: from requirements definitions to planning activities, Software Life Cycle (SLC) activities, configuration management, hardware purchasing, software maintenance and training.
- It assures constant improvements in SLC, with Quality Assurance and Quality Control activities such as corrective and preventive actions.
- It improves co-operation between all parts in the SLC processes.
- Its international recognition (national and international certification schemes) has a strong impact on the software industry.

The weaknesses of implemented ISO 9001 in a software development process are:

- The quality system is limited to the software development processes. IS resources such as life-ware, orgware, hardware and other IS resources are not directly considered;
- Implementation of activities such as security management, application control and technology specific controls, depend on the maturity of the implemented quality system. This may vary from not implemented at all to fully implemented procedures;
- Implementation of the ISO 9001 standard does not give clear answers to questions concerning productivity, functionality, usability, reliability, cost effectiveness etc. of an IS.
- A controlled software development process is no guarantee for quality solutions to business problems. If the user defines a bad or inadequate requirement, this will be with high quality built in a software, but the end product will be of no use.

The above discussion consider only the software development process. An interesting situation is, where IT resources in an enterprise are under the umbrella of ISO 900x requirements, but other business processes are not. The information products are faced with problems like functionality, availability or usability in such a cases. The root causes are in the different maturity levels of business orgware. A Data Base Administrator can not design a robust and accurate data model of an enterprise if the enterprise has no business vision or policy statement, long and short term business plans, and also in the case when he or she may or can communicate only with people from third level management. The business objectives of an enterprise are not achieved only with the IT resources. IT resources are the support to other processes and ideally, all of them must be on the same maturity level to achieve expected business results.

The implementation of the ISO 9001 standard in the software development process has no essential influence on a very important IS source, orgware. This means that a quality of IS can not be assured only by implementation of the ISO 9001 standard in the software development process, but it is also necessary to consider the maturity of the organisational and management level of the company. In other words, the impact of ISO 9001 in software development and on IS depend on the maturity of the environment for which the software development is intended.

However, it is to be expected that the gap between IT processes and business processes will be reduced in time if the quality system is implemented only in one of them. A quality system requires internal audits and corrective and preventive actions which assure improvements and growth of all involved.

4 Influence of software product standards on IS quality

There are several hundred software standards. Most of them are National or Multinational such as ANSI (American National Standard Institute), BSI (British Standard Institute) and DIN (Deutsches Institute für Normung) standards, and professional standards such as IEEE Standards, Defence standards etc. The majority of them deal with the results of software activities and tasks such as Management, Quality Assurance, Configuration Management, Safety, Design, Requirement Specification, Coding, Verification & Validation etc. and are mainly considered in the process of software development (discussed in sections 2,4, and 5). The applicability of them and their influence on IS quality therefore depends on the scope of the standard within the software development process.

Only a few of the software product standards deal with software products for end users or software pack-

ages. They are: ISO/IEC 12119 1995: *Information Technology - Software packages - Quality requirements and testing*, ISO/IEC 9126 1991: *Information Technology - Software product evaluation - Quality characteristics and guidelines for their use* and ISO 14598:1996 - Part 1 to Part 6: *Information Technology, Software product evaluation*.

The international standard ISO/IEC 12119 is applicable to software packages like accounting, payroll, data base programs etc. Sets of quality requirements based on this standard are: requirements on product description, documentation, programs and data, and testing procedure.

The strengths of software product standards are:

- Conformity according to those standards provide confidence that the product actually does what it claims to.
- They can be used in national or international software packages certification schemes.
- They can be used as a marketing advantage for off the shelf products;
- They can have a substantial impact in a software acquisition process.

The weaknesses of those standards generally deal with their scope and their importance in the IS. Existing software product standards cover only some parts of the software design process or software products. Its influence on IS quality is limited to the importance of the considered subject of software quality. It is obvious that safety standards (implemented in a software development process!) have a supreme influence on IS quality in safety critical systems, such as Nuclear Power Plant. And vice versa: a text processor package with a certificate of conformity with ISO/IEC 12119, has only little influence on IS quality.

5 Influence of software process assessment models on IS quality

It is very important to recognise, that any software process improvement program needs a sound understanding of the current status of the software development process. Software process assessment is the most common method used to achieve this understanding. Process assessment is defined as *The disciplined examination of the process used by an organisation against a set of criteria to determine the capability of those processes to perform within quality, cost and schedule goals. The aim is to characterise current practice, identifying strengths and weaknesses*

and the ability of the process to control or avoid significant causes of poor quality, cost and schedule performance. [ISO/IEC JTC1/SC7 1992]. This definition is also applicable to software process assessment.

The most popular approaches for software process assessment are the Software Engineering Institute's (USA) CMM - Capability Maturity Model [Paulk et. al 1993] [Paulk M.C. 1995] [CMM v2.0], ISO/SPICE (Software Process Improvement and Capability Determination) project [ISO 15504 (SPICE) PDTR Draft 1996], and BOOTSTRAP (European developed assessment method [Haase et al 1993]). Some of the others are: Capers-Jones software measurement model [Jones 1991] and Model-based Process Assessment [McGowan et al 1993]. SLC is also subject of standardisation: IEEE standard for software life cycle processes [IEEE 1988] and ISO/IEC12207: 1995 Information Technology - Software life cycle processes (SLC). Many other models and standards can be found in the Software Productivity Consortium WEB server [SPC 1997].

The CMM model provides a conceptual structure for improving the management and development of a software process in a disciplined and consistent way. The CMM model divides the software process into five maturity levels which highlight the primary process changes made at each level: Initial or basic level (*ad hoc* process), Repeatable (basic project management is established), Defined (process is documented and standardised), Managed (process and product measurements are established) and Optimised. Each level comprises of a set of process goals that, when satisfied, stabilise an important part of the software process. This in turn results in an increase of the software process capability. Each maturity level is composed of a number of key process areas. These key process areas are a set of activities that, when implemented, achieve a set of goals important for enhancing the process capability.

A European software process assessment and improvement method - BOOTSTRAP - has been developed in an ESPRIT (1990 - 1993) project. BOOTSTRAP has been built on the basis of the CMM and ISO 9000 series of standards. The basic concept underlying BOOTSTRAP requires first fulfilment of basic organisational requirements, such as process control, project management and risk management before any changes in methods and technology are made to improve the software process. An organisation and its processes are assessed with respect to organisation, methodology and technology. The result of BOOTSTRAP assessment is a capability profile showing maturity of an organisation against an ideal level of maturity, comparison with ISO 9001 requirements and recommendations for appropriate actions for further improvements.

ISO/SPICE is a project of the international Committee on Software Engineering Standards ISO/IEC JTC1/SC7. It synthesises above mentioned models and standards (CMM, Bootstrap, Trillium, ISO 9001, ISO 12207 and others). SPICE embodies a sophisticated model for software process management drawn from the world-wide experience of large and small companies. The architecture of the SPICE process assessment defines a two-dimensional view of software process capability: the process categories and capability level. Process categories are: Customer-supplier process category, Engineering process category, Project process category, Support process category and Organisation process category. Each category is a set of processes addressing the same general area of business. The result of SPICE assessment is a "software process profile" defining a capability level for a considered software process category. This model will have a significant influence on software domain in the near future: very probably as "the facto standard".

The strengths of software process assessment models are that:

- The Software Life Cycle processes are dealt with in full.
- They give a clear profile of the current capability of the software development and maintenance processes. This profile corresponds to the maturity level of the software development and maintenance process. Maturity levels or maturity profiles identifies current capabilities of the process and identify process areas for further improvements.
- They have a important influence on the software community.
- The results of an assessment can be used in a benchmarking process.
- The results can be used in national or international procurement activities.
- They can be used for self assessment, as a starting point into a quality improvement program.

The weaknesses of those models are:

- They are limited to the Software Life Cycle and do not consider other IS/IT resources such as orgware, peopleware, hardware, telecommunications etc.
- They are not known outside of the software community (to the IT users).
- They are neither national or international standards.

Companies where those models are applicable are software houses and Electronic Data Processing departments or software development departments within enterprises. They can be used as a self assessment tool for improvements plans or implemented by a third party as an independent assessment, if stakeholders require such an assessment.

6 IS Auditing

The EDP Auditor Foundation, Inc. (EDPAF) developed General standards for information system auditing [Dykman A.C.] and Control objectives as a model for IS audit procedure. By general standards for information systems auditing [IS Audit], the Information System Auditing is defined as *any audit that encompasses the review and evaluation of all aspects (or any portion) of automated information processing systems, including related non automated processes, and the interfaces between them.* Those aspects, defined by EDP auditor Control objectives [Dykman A.C.] are:

- Management control
- Information system development, acquisition, and maintenance
- Information system operations controls
- Application controls
- Database supported information system controls
- Distributed data processing and network operations controls
- Electronic data interchange controls
- Service bureau operations controls
- Micro computer controls
- Local area network controls
- Expert system controls and
- Joint application design controls.

Each general control is divided in to controllable and manageable units. From the definition of the IS, the IS audit definition by EDPAF, and the described control objectives it may be concluded that EDP audit procedures deals with all aspects of an IS. The result of the IS audit procedure is a set of documented facts obtained with interviews and questionnaires by a certified auditor on an audited IS entity.

The strengths of the IS auditing assessment model are:

- any IS/IT entity can be audited;
- it is a strong management tool;

- special considerations are pointed on technical aspects such as Data Bases, LANs, Micro Computer Control, application controls, IT/IS risk evaluation and data security etc.;
- it is usually used in collaboration with financial auditing;
- it can be used as a basis for an improvement program;

The weaknesses of this model are:

- it can not be used as a tool to find out the current maturity level of a software development process or to compare a project profile;
- it can not be used in an competition attaining work / business
- it is neither a national or international standard;
- it can not be used as a self assessment model;

This model is applicable mostly to the auditing of EDP departments and other IS resources within enterprises [Pivka M. 1998]. Owners of IS auditing are usually company's management and accounting auditors. IS auditing is generally not applicable to software houses. Detail informations on control objectives for information and related technology are in [CobiT 98] which is a registered trade mark of ISACA.

7 IS control mechanisms: collaboration, competition or conflicts?

Information Technology managers, software development managers, business managers and users are faced with aggressively marketed control mechanisms. The following questions are interesting from the IS management point of view:

1. Which model to choose?
2. Which IS resources are controlled by those mechanisms?
3. Are they redundant?
4. Do they compete?
5. Do they collaborate?

The answer to those and other questions on IS control mechanisms are not easy and universal. The following paragraphs describe the most general characteristics of control mechanisms and their influence on IS resources.

In table 1, the ranking of the influence of control mechanisms on some IS resources is defined. It is obvious, that assessment models have limited or nil influence on technology resources such as communications. But of course, they have a strong influence on the software development process.

Table 2 represents which model to choose for some most interesting business requirements.

Quality systems based on ISO 9001, BOOTSTRAP, SPICE, CMM and other assessment models, are **management tools** for improving the software development, maintenance and implementation process. They have an important influence on the software environment in defining their maturity level and in helping them to find and define the key management procedures to improve the software process. Assessment teams (first, second or third party teams) use BOOTSTRAP, SPICE or the CMM model to identify the maturity level of the software process. Models are not used in national or international certification schemes. On the other hand, the ISO 9001 certificate confirms at an international level that the software process is in compliance with internationally accepted quality requirements. The influence on IS of those models is therefore limited to the scope of the model! We may conclude, that there is some competition and conflicts between them, but also a possibility for collaboration. For instance: self assessment with a CMM model can be a good starting point for an improvement program with the aim of attaining an ISO 9001 certificate.

The influence of **software product standards** on IS quality is limited to the importance of the considered software package or the scope of the standard. Those standards shall therefore be recognised as helpful and useful in assessing aspects such as risk management, application control, data security, or any other IS/IT entity, where such standards exist and are implemented.

The **EDPAA IS audit model** is a tool for general management to evaluate the efficiency, security, productivity etc. of implemented IS/IT, or part of it, in a company. IS auditing does not deal with the natural growth of software processes as proposed by the BOOTSTRAP, SPICE, CMM, or with quality systems as defined in ISO 9000 family standards. IS audits are most usually ordered by top management or accounting auditors. There are some gaps between IS auditing and other models, which are for sure possibilities for collaboration. At least the following aspects are a matter of collaboration: application controls, risk management and IT assessment.

There is also some overlapping (and therefore, conflicts and competition) between those models, especially between assessment models. BOOTSTRAP, SPICE and CMM models overlapped each other and to conduct more than one of them in practice, is redundant. **Which of them to chose is in our opin-**

	ISO 9001	CMM, SPICE BOOTSTRAP	Product standards	IS Auditing
People	Strong	Strong	none	Medium
Application system	Strong	Medium	medium, depends on the standard and the application system	Strong
Software development	Strong	Strong	medium, depends on the standard and applied SLC	Medium
Technology of IS	Medium	Medium	none	Strong
Facilities	Medium	None	none	Strong
Data security	Depends on the maturity of the company: from medium to strong	depends on the maturity of the company: from medium to strong	standard dependable	Strong
Risk management	Depends on the maturity of the company: from medium to strong	depends on the maturity of the company: from medium to strong	standard dependable	Strong
Communications	Medium	Medium	standard dependable	Strong

Table 1: Control mechanisms and their influence on IS

Requirement: To assess specific IS/IT sources concerned productivity, security, usability, ...	Requirement: To assess software process as the basic for software improvement programme.	Requirement: To buy (or produce) SW product for market with legal or specific requirements	Requirement: To improve software quality and to get market Advantage
Possible solutions: IS auditing, combined with assessment methods if necessary for software process. Typical stakeholders: top management, financial auditors.	Possible solutions: Assessment methods: BOOTSTRAP, CMM, SPICE, ISO 9001. Depends on added value from assessor company. Typical stakeholders: SW managers, user requirement.	Possible solutions: Software product standards or ISO 9001. Typical stakeholders: legal requirements, market or contractor's demands.	Possible solutions: ISO 900x certification with one of the assessments methods. Usually influenced by market demands or by management.

Table 2: which model to chose.

ion the matter of added value, given from the method and assessor company (third party assessment). It is generally accepted that the software process which is certified to be in compliance with ISO 9001 is on the third capability level based on the CMM model. Overlapping between IS Auditing and other control mechanisms is obvious only with the software development and maintenance process and as that, SLC processes are an aspect of possible conflicts and competitions.

The responsibility for proper decisions between different models is unfortunately on the stakeholders side. **In situations where IS auditors, auditors for quality systems (ISO 9000 family) and software process assessors are hired, or some assessment results exist from the past, we recommend that company management demands from all parties collaboration and a mutual recognition of their results.** This shall have a substantial influence on the costs and added value of the company. Otherwise, a lot of people in the company will be interviewed several times with similar questions on the same subjects! Table 2 below shows an example of which model to choose for some business targets or strategies.

8 Conclusions

In this paper only the most popular and well defined control mechanisms to achieve better IS quality are briefly presented. Those control mechanisms are: ISO 9001 quality system standards, software process assessment models (CMM, BOOTSTRAP, SPICE), software product standards and IS Auditing.

The answer to the question which of them to choose is therefore not easy and it depends on the perspective of the stakeholder: enterprise management, IS management, buyer of software package, contractor for software or software services etc. Once the scope and required goals and expectations are defined and strengths and weakness of available control mechanisms understood then the right choice is not so difficult any more. This can be derived from the descriptions above. It is also obvious, that some competition, conflicts and possibilities for collaboration exist between the considered models.

It is the management's role and responsibility to understand those models and to avoid different audits on the same IS resource.

Acknowledgements

I would like to express my thanks to Gilles MOTET (DGEI/INSA Toulouse, France), Régis FLEURQUIN (IUT Informatique Vannes, France) and Dr. Walter WINTERSTEIGER (Management & Informatik

Dornbirn, Austria) for their important and valuable comments on the paper.

References

- [Haase et al 1993] V. Haase, R. Messnarz, G.Koch, H.J.Kugler, P. Decrinis: BOOTSTRAP: Fine-tuning process assesment, IEEE Software, July, 1994 pp 25-35.
- [Dykman A.C.] Dr. Charlene A. Dykman (editor): Control objectives. Controls in an Information systems Environment: Objectives, Guidelines, and Audit Procedures. EDP Auditors Foundation, Inc. Illinois, USA, 1992.
- [CMM v2.0] Draft version of CMM v 2.0. Internet web site <http://www.sei.cmu.edu>
- [CobiT 98] Control Objectives for Information & related technology. ISACA, Illinois, USA, 1998
- [IEEE 1988] Institute of Electrical and Electronics Engineers Inc., Standard of Software Life Cycle Processes, P1074/D2.1 Dec. 1988.
- [IS Audit] IS Audit & Control Journal: General standards for IS auditing. Vol. I. 1994, pp 60 - 66.
- [ISO 9000-3:1997] Quality Management and Quality Assurance Standards - Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software.
- [ISO/IEC JTC1/SC7 1992] The Need and Requirements for a Software Process Assessment Standard. Study Report Issue 2.0 JTC1/SC7 N944R.
- [ISO/IEC 8402:1995] Quality management and quality assurance - Vocabulary.
- [ISO/IEC 9126:1991] Information Technology - Software product evaluation - Quality characteristics and guidelines for their use.
- [ISO/IEC 12119:1995] Information Technology - Software packages - Quality requirements and testing.
- [ISO/IEC 12207:1995] Information technology - Software life cycle processes
- [ISO/IEC DIS 14598:1996] Information technology - Software product evaluation. Part 1: General overview. Part 5: Process for evaluators.
- [ISO 15504 (SPICE) PDTR Draft 1996] Software Process Improvement and Capability determination (Software Process Assessment Standard). SPICE web site: <http://www-sqi.cit.gu.edu.au/spice/suite.shtml>

- [Jones 1991] T. Capers Jones, Applied Software Measurement, McGraw Hill, 1991.
- [McGowan et al 1993] C.L.McGowan and S.A.Bohner, Model Based Process Assessment, Proc. Int. Conf. Software Engineering IEEE CS Press, Los Alamitos, 1993.
- [O'Brien J. A. 1994] Introduction to Information Systems. IRWIN 1994.
- [Paulk et. al 1993] M. Paulk, Bill Curtis, M.B. Chriss&C Webber, Capability Maturity Model, Version 1.1. IEEE Software, July 1993, pp18-27.
- [Paulk M.C. 1995] Paulk M.C. How ISO 9001 Compares with the CMM. IEEE Software, January 1995, pp. 74-83.
- [Pivka M. 1998] Pivka Marjan. A Comparison of Control Mechanisms to Help Achieve Better IS Quality. ISAudit & Control Journal Volume II 1998.
- [Rout P. Terence 1995] SPICE: A Framework for Software Process Assessment. Software Process Improvement and Practice. August 1995.
- [SPC 1997] Software Productivity Consortium: <http://www.software.org/quagmire>.
- [TickIT 1998] The TickIT Guide. Issue 4.0. DISC TickIT Office. London 1998.

Authentic and Functional Intelligence

Mario Radovan

University of Rijeka, Faculty of Philosophy, Department of Information Science,
Omladinska 14, 51000 Rijeka, Croatia
E-mail: mradovan@mapef.pefri.hr

Keywords: mind, computation, subjectivity, understanding, thinking, intelligence, three worlds, care thesis, coping-with

Edited by: Rudi Murn

Received: April 15, 1998

Revised: May 21, 1998

Accepted: June 16, 1998

Philosophical discussions about the aims, possibilities and limitations of Artificial Intelligence (AI) can shed light on the plausibility of different approaches to cognition and computation, and with that, they can have great impact on the future development of computer technologies. However, we argue that such discussions are often based on the vague concepts or on the unsafe assumptions. Intelligence and understanding are usually observed on the level of behaviour; we argue that these phenomena should be considered in terms of motivations; in that context, we hold it necessary to differentiate between authentic and functional cognitive abilities. Computation does not seem to be a plausible way toward authentic understanding and intelligence; however, computational systems do offer virtually unlimited possibilities to replicate and exceed human cognitive abilities on the functional level.

1 Introduction

Discussions about the aims, possibilities and limitations of Artificial Intelligence (AI) take place between two basic and mutually opposed theses. The first one claims that the goal of AI is not only to produce systems which mimic some specific intelligent behaviour but to create *machines with minds*, in the full sense of the word. On such a view, human beings are biologically created computers. The other thesis claims that the very idea that the human mind could be artificially replicated by computational systems of any kind is radically wrong. In this paper, we argue that neither of these thesis has so far offered a conclusive argument in its favour. In light of our present knowledge, however, we hold the second thesis more plausible. But this thesis cannot be actually proved, not only because present knowledge about the human brain is not sufficient to provide such a proof, but primarily because the basic features of human cognition - such as awareness, understanding, thinking, and intelligence - are not defined well enough. Arguments about the relationship between human cognition and computation are almost always based on such concepts, without it being mentioned that there is any problem with their meaning. Although often interesting, such arguments are vague (or simply invalid), or they can be reduced to some rather obvious and trivial assertion. In section (2) we discuss a few typical examples of such arguments and their drawbacks. In section (3) we deal with the perennial problem of the subjectivity of mental states. We

introduce the idea of *three worlds* - the physical world, the world of subjective states, and the world created by humans. We argue that such a three-world ontology offers a basic conceptual framework within which problems involving cognition and computation can be expressed and discussed in the most appropriate way. In section (4) we discuss the basic problems and limitations of AI in terms of this three-world ontology. In this context, we introduce the distinction between *authentic* and *functional* intelligence. We claim that computation is not a plausible way towards an artificial replication of subjectivity, and hence towards authentic understanding and intelligence. However, we recognise that AI does hold unlimited possibilities for replicating human cognitive abilities on the functional level. We argue that authentic intelligence is out of reach of purely computational systems, though on the level of functional intelligence, performances of computational systems will be more and more out of reach of human capacities. A recent example of this trend comes from computer chess, where machine functional (performatonal) intelligence surpassed human capacities. Chess programs are often criticised to rely mainly on the “brute force” of the machine. I do not think that the qualities of such programs could be reduced to “brute force”; however, even if they could be so reduced in part, that would not be unusual when qualities of machine performance are concerned. On the other hand, if the main goal of AI is to create a machine with an authentic mind, then we must face the fact that we do not actually have any clear idea what

direction of research could lead to such an artificial mind. In this context, we claim that an authentic artificial mind is not possible without a nearly authentic *sentient* life, and that leads us out of AI, perhaps towards biology.

2 Simplistic Solutions

In this section, we discuss some typical arguments based on vague concepts and unsound assumptions, or trivially true results. Thus, such arguments can tell us far less than they pretend to do. The victory of the chess-playing system Deep Blue over the world's chess champion offers an illustrative example of the tone which dominates in such discussions. As a first, the media presented the event as a *threat to human dignity*, as if the designers and creators of Deep Blue were not human beings. On the other hand, Searle offers a calming explanation: "The computer knows nothing of chess ... It just manipulates meaningless formal symbols according to the instructions we give it" (Searle '97, p. 59). Such an explanation of machine abilities seems to be more misleading than informative. Namely, it implicitly assumes that the skill of a chess master consists of something radically *different* than the ability to "manipulate meaningless formal symbols". But let us suppose that Searle knows how to play chess, and that he is no less intelligent than the world champion. Does this mean that he would have equal chances of winning a match with the world champion? Probably not. But isn't that simply because the world champion knows far more "meaningless" rules about which move seems best in which position than Searle does? Searle gives no explanation of the nature of a chess master's "meaningful" way of reasoning (or computation). Hence, it is not obvious why the claim stated for a computer - that it "knows nothing" except to "manipulate meaningless formal symbols" - could not be equally stated about the world chess champion himself.

2.1 Replicating the Mind

Let us start with arguments about the theoretical possibility of complete artificial replication of the human mind. Although computers vastly exceed humans in performing various sorts of computations, it seems that no machine could ever appreciate a wine, or a sonata, for example. Hence, it seems that no machine could fully replicate a conscious human mind. However, Dennett, who advocates "a version of functionalism", says: "if all the control functions of a human wine taster's brain can be reproduced in silicon chips, the enjoyment will *ipso facto* be reproduced as well" (Dennett, p. 31). Such a claim seems to be true by definition. Namely, if you reproduce all causes (and that is what "all the control functions" should be),

you would also reproduce all effects. However, such a claim leaves open at least two essential questions: *what* are the "control functions" of the brain, and *how* (if at all) could these functions be reproduced in a lifeless system. If we would know the answers to these two questions, we would hardly need anything more. But there are no clear indications that the functionalist approach could ever lead to the answers to such questions. Hence, the trivial (even if true) conclusion of the argument tell us very little about the real problems concerning the possibility of fully replicating the human mind.

The problem of replicating the mind is usually posed in a more precise manner, as the question whether the human mind could be fully replicated by a *computational* system. In that context, there are claims that the *universal Turing machine* and the *Church-Turing thesis* offer the theoretical grounds for a positive answer to this question. Roughly speaking, the universal Turing machine is a symbol system which consists of a set of symbols, operations, internal states and state change instructions by means of which algorithms can be defined and performed. The Church-Turing thesis says that every computable system can be simulated by the universal Turing machine. Such a machine can be implemented on a digital computer (neglecting the rather theoretical limitation that the universal Turing machine has an unlimited input tape). Hence, the Church-Turing thesis opens the following possibility: if the human brain is a computable system, then a suitably programmed computer could replicate all its features, and with that fully replicate the mind. Thus, Goertzel claims that (1) humans are "systems governed by the equations of physics", and (2) "the equations of physics can be approximated, to within any degree of accuracy, by space and time discrete iterations that can be represented as Turing machine programs" (Goertzel, p. 22). Consequently, according to Goertzel, brain activities - and with that, human cognitive abilities - can be fully replicated (at least in principle) by implementation on the universal Turing machine.

On the other hand, on the basis of Goedel's theorem, Penrose claims that human cognitive abilities cannot even in principle be replicated by a computational system of any kind (Penrose, p. 116). Roughly speaking, Goedel's theorem shows that there are sentences in a system which are not provable in that system, but humans can see they are true (under a certain interpretation). Any computational replication of the mind could "see" only those things which *are* computable: therefore, *less* than a human mathematician can see. Hence, independently of the "technical" problems, the mind is not even in principle computationally replicable. Arguing against Penrose's position, Searle claims that from the fact that computational simulation of some human ability cannot be done on *some* level of

description (i.e., on the level of the language of mathematics), it does not follow that the same ability cannot be simulated on some *other* level of description (Searle '97). Hence, he holds that Goedel's theorem does not prove that a full simulation of human brain processes is not possible. Searle holds that those brain processes which take part in a mathematician's brain when he sees the truth of an unprovable sentence could be (at least in principle) completely simulated on the level of neurons, so that the computational system (which performs the simulation) would have the same knowledge as the mathematician. Penrose admits that a computational simulation of the brain on the neural level seems possible; however, on the basis of Goedel's theorem, he insists that human cognitive abilities cannot be fully replicated by *any* sort of computation. Hence, Penrose speculates that the source of human cognitive abilities (which transcend the computable) should be looked for, not on the level of neurons (which seems to be computable), but on some *lower* level for which there would not exist any computable description.

All these positions concerning the computability of the brain take as obvious something that is of essential importance, but is not known. For Goertzel, it is "a physical fact" that the brain is a computer, and that it "deals only with computable functions". Searle assumes that the existence of truths which are not computable on the level of conscious thought does not say anything relevant about the possibility that the system itself (i.e. brain) could be non-computable. And, granted that the system (i.e. brain) can see something that is not computable (on some level of description), Penrose takes it for granted that the system itself is not computable. However, it seems that the question of computability in the human brain is an open question. Copeland gives an analysis of that problem, based on the fact that there exist *undecidable sets* and that human cognitive system is *productive*, on the basis of which he claims that "for all we know many aspects of brain function may be non-computable". However, Copeland concludes that this problem is still *terra incognita* and that "in point of fact" we do not have any definite answer concerning the computability of the human brain (Copeland, p. 233). Let us add here that computability is an abstract category, and it is not obvious how to apply such a category to an organic system, at least not as long as that system has not been formally described. And an organic system can be described in many different ways, which renders the problem even more difficult. Hence, the basic drawback of the arguments and positions listed above is not that they are wrong, but rather that they do not pay enough attention to that problem which is essential for their very coherence.

2.2 Causal Powers

The same kind of neglect and simplification is peculiar to arguments concerned with the necessary conditions which an artefact must fulfil to be (or to become) conscious or to have a mind in the full sense of the word. It seems that the simplest way to replicate the conscious mind by artificial means would be to replicate the "neurobiological basis" from which conscious mind emerges in organisms like ourselves. On the other hand, it seems logically possible to obtain the same effect by means and methods radically different from the ones in human brains. However, Searle emphasises that from the fact that the human brain causes consciousness it follows that "anything else capable of causing consciousness would have to have the relevant *causal* powers at least equal to the minimal powers that human and animal brains have for the production of consciousness" (Searle '97, pp. 158-59). In other words, an artificial product made from different matter and in a structurally different way than the human brain might "cause consciousness" if and only if its structure shared with the brain the "causal powers" to get it over the threshold of consciousness. In short, to be conscious, an artificial system (of whatever kind) "must be able to cause what brains cause" (Searle '97, p. 191).

The problem with Searle's claims (as well as with Dennett's) is that their negation would lead to contradiction. If we want an artefact with consciousness like the one "caused" by a brain, the artefact must be of such a "structure" to have sufficient "causal powers" to "cause what brains cause". Indeed, to deny such a claim would mean to contradict the principle of causality, which is the cornerstone of scientific explanations. Hence, although such claims leave an impression that we know something essential about the way an artificial mind could be created, they tell us very little. Namely, the real problem is *what* are these "causal powers" which get us over the threshold of consciousness. And that is an empirical question.

We encounter the same kind of problems with "proofs" that the human mind is (or is not) a computational system, or that computer can (or cannot) be conscious, think and understand. The essence of the problem with such proofs can be reduced to the following form of reasoning. There are entities of type H (humans) which manifest a property P (consciousness, intelligence, etc.). To prove that a computer cannot have property P, one tries to show that there is nothing in it that *could* cause P. However, as long as we do not know *what* causes P in H, there is not much sense in trying to *prove* that entities different from H cannot have property P. The claim that in order to have property P a thing would have to have the "causal power" to produce P does not say anything substantial. It seems reasonable to suppose that, for example, a thermostat does not have any conscious mental

states, but it would be less reasonable to try to prove it. As Chalmers puts it, there may exist some “crucial ingredient in processing that the thermostat lacks and that a mouse possesses, or that a mouse lacks and a human possesses”; however, Chalmers holds that there is “no such ingredient that is *obviously* required for experience, and indeed it is not obvious that such an ingredient must exist” (Chalmers, p. 259). In his recent book, Searle explains that he did not intend to prove that computers cannot be conscious, as his arguments were often interpreted. After all, if consciousness can emerge in a human brain (a lump of grey matter), why could the same (or similar) state not emerge in some other sort of system? What he intended to prove, says Searle, was that “computational operations by themselves”, as symbol manipulations, are “not sufficient to *guarantee* the presence of consciousness” (Searle ’97, p. 209). He claims that his proof is based on the fact that “syntax by itself has no mental content”, and on the fact that “the abstract symbols have no causal powers to cause consciousness because they have no causal powers at all” (Searle ’97, p. 210). In this, Searle is right. However, such an argument could hardly be called a proof, since these truths hold by definition: syntax is not, has not, and can not cause, semantics. On the other hand, nothing can be said to “guarantee the presence of consciousness”, as long as we do not know *what* it is that guarantees its presence in the human brain.

2.3 Solution by Definition

The main problem for those who argue that computational systems can (at list in principle) fully replicate the human cognitive abilities, represent the consciousness which, it seems, computers neither have nor could create it by computation. The typical approach to conscious mental states in such arguments reduces to finding a way to exclude them from the scope of discourse. As an example, let us take Copeland’s argument about the machine thinking. Copeland claims that the questions “Could a computer literally think?” can be settled “only by a *decision* on our part”, as is always the case when old concepts are used in new situations (Copeland, p. 53). In other words, although we may never manage to build machines that think, we can nevertheless “settle the question of whether or not it is a conceptual mistake to say that an artefact might literally think”. Copeland argues that there is nothing wrong with applying the term ‘thinking’ in its literal sense “to an artefact of the right sort” (Copeland, p. 33).

On the basis of the fact that “we are not consciously aware of all, or even most, of our mental processes”, Copeland assumes that consciousness is not of essential importance for the human cognition (Copeland, p. 34). He claims that human mental activities, such

as understanding speech and perceiving the external world, can be performed non-consciously; and since humans can perform these activities non-consciously, Copeland claims that the question whether an artefact could be said to perform such activities can be discussed “without considering whether or not an artefact could be conscious” (Copeland, p. 37). However, we hold that such way of reasoning is wrong. Namely, from the fact that *some* human thoughts and acts are not conscious it does not follow that an artefact which is *never* conscious can think in the literal sense of the word. After getting rid of consciousness, Copeland introduces the notion of “massively adaptable” internal processes. The internal processes of a system are said to be massively adaptable if the system itself can “analyse situations, deliberate, reason, exploit analogies, revise beliefs in the light of experience, weigh up conflicting interests, ... and so forth” (Copeland, p. 79). Now, suppose that one day such an artefact with massively adaptable inner processes is produced. Would you say that it thinks? Of course, yes. However, Copeland does not tell us *how* could an artefact “revise beliefs in the light of experience” if that artefact *never has* any conscious experience. But instead of dealing with such questions, Copeland introduces a robot which eats, writes poetry, “and so forth”. Now, when we are “confronted” with such a robot, “we ought to say that it thinks”, says Copeland, because “the contrary decision would be impossible to justify” (Copeland, p. 132). The main (and fatal) drawback of Copeland’s argument consists in the fact that he completely neglects consciousness, without which there is no coherent way to speak about the mental states at all. On the other hand, his definition of massive adaptability is such that it literally *implies* the ability to think; hence, to conclude that a “massively adaptable” artefact *should* be said to think means only to explicate a direct consequence of the definition. In section (4) we argue that such a definitional approach to the problem of thinking is wrong, and we propose an opposite, *motivational*, approach to the problem of defining the basic cognitive categories for humans and machines.

3 The Problem of Subjectivity

Science assumes that reality is objective in the sense that neither its existence nor its structure depend on the viewpoint of a particular observer. Scientific knowledge deals with things which could exist without being known, and is expressed in a way which is equally accessible to every (sufficiently qualified) human. We say that science speaks of phenomena from the *neutral* (or third-person) point of view. On the other side, subjective mental states seem to be a different kind of phenomena in an ontological and epistemic sense. Namely, mental states can exist only as *some-*

one's states, and they are known only to the subject whose states they are. For example, neural activities that cause a pain could (in principle) exist and be observed without being experienced, but the *pain itself* could not: it exists only if and as experienced. In other words, it seems that conscious mental states cannot be described in an objective fashion for the simple reason that they are *always and only* someone's *subjective* states. Churchland says that the taxonomy of physical science has some limits, and that it reaches them "at the subjective character of the contents of consciousness" (Churchland, p. 196). In this context, positions about the nature of mental states are divided into two basic views (which appear in various variants): *physicalism* and *property dualism*.

3.1 Physicalism and Property Dualism

Physicalism claims that mental states can be ontologically reduced to physical states of the brain, and thus expressed in the objective (third-person) language of science - firstly, in the language of neurology, and then in the language of biology and the language of physics. Copeland, who considers himself a physicalist, admits that physicalism is "supported only by faith", or more precisely, by the fact that other theories seem less plausible or at least less "comfortable". He says: "The anti-physicalist alternative - an irreducibly mental dimension that sticks out of an otherwise physical universe like a sore thumb - is unpalatable to us. It offends against our expectation that nature is a harmonious, integrated affair" (Copeland, p. 179). In essence, physicalism excludes the subjective from the scope of the discussion, but in doing so it does not solve the problem of how to deal with the "offensive" fact that the universe *does* contain subjective states of people like you and me. On the other side, property dualism holds that the attributes 'mental' and 'physical' designate two ontologically different kinds of properties, and that the same entity can have both kinds of properties. Mental is caused by the physical, but it cannot be expressed by the taxonomy of physics, since subjective states (qualia) lie beyond the reach of the objective language of science. There is a special feeling, a *quale*, to each of the conscious states; property dualists hold that we would be none the wiser about qualia even if neuroscience were completed and everything would be known about the biology and physics of the brain. For, to know everything about the physical processes going on in the brain does not mean to know anything about the hurtfulness of pain or the bliss of joy. However, by accepting the irreducibility of conscious mental states, property dualism faces the problem of explaining the *relation* between the physical and the mental "dimensions" of reality. That is, to say that the mental is caused by the physical but

is not reducible to it only explicates the problem, and does not solve it.

It has been suggested that some neural activities in specific regions of the brain (in the networks connecting the thalamus and the cortex) might be the source of conscious mental states. However, even if some neural "firing" could explain the *how* of conscious states, it would not explain the *what* of these states. No explanation on the physical (neural or subneural) level can express a subjective state as experienced, since the taxonomy of science does not have terms which could express a feeling (*quale*) which are intrinsically subjective and ontologically irreducible to the physical. We lack not only an explanation of the relationship between subjective mental states and physical activities of the brain, but we cannot even conceive the sort of taxonomy which could allow an explanation of the relationship between *subjective* and *objective*. Hence, the basic question is not whether machines can think - and therefore *be*, that is, whether *to be* means to be a *computer* - but is instead the question how to speak about subjective phenomena in the scientific fashion at all, since the scientific taxonomy is intrinsically objective. The mystery of the subjective dimension of reality is primarily of a conceptual nature, and we don't really know how to solve it.

3.2 The Three Worlds Framework

The basic aim of ontology is to define a conceptual system which makes it possible to speak about that which exist. In other words, a minimal requirement which an ontology should satisfy is to provide a conceptual framework within which we can appropriately speak about known (or, in principle, verifiable) phenomena. We claim that an adoption of the three-world ontological framework - which emphasises the difference between *the physical world*, *the world of subjective states*, and *the world created by humans* - opens the possibility to express and discuss the problems of human cognition and computation in the most appropriate way. The idea of the "three worlds" is attributed to Popper, but it has not been widely accepted by other philosopher, especially not in the context of the discussions about the relationship between human cognition and computation. Popper's basic idea about the three worlds is clear, but he treats many problems only barely or unsuitably, so that they can be considered open.

We assume that to the physical world (*world1*) belong all natural phenomena up to but not including the phenomena of consciousness as a boundary case. Although consciousness is a natural phenomenon, it is that peculiar and unique natural phenomenon which at the same time also comprises a new world for itself (*world2*): the world of subjective states, such as pain, desire, love or anger. The world created by humans

(*world3*) contains everything that has been created by conscious human beings (even though not all entities from *world3* have to be created consciously). *World3* contains abstract entities such as numbers and theorems as well as *all forms* created by humans. We assume here that forms have an autonomous existence; however, they are not eternal (in the Platonic sense), but created by humans and (often) imposed on the physical world. For example, a desk conceived as a piece of wood belongs to the physical world; however, conceived as a *desk*, in a functional or aesthetic sense, it is a creation of *world2* (the conscious mind) and belongs to *world3*, the world of the created by humans. The same holds for a computer: the computer on my desk is a physical object, but what makes that “heap of atoms” on my desk a *computer* is the *form* imposed on it by humans.

In the context of the unsolved problem of subjectivity, the question of the mutual impacts between the three worlds does not have a simple solution. According to Popper, consciousness “plays the main role in the causal chains” that lead from *world3* to *world1* (Popper, 24). Popper’s arguments are not precise enough to be simply accepted or to allow a clear refutation. However, it seems clear that the three worlds ontological frame does not, by itself, solve the problem of “causal chains” between the three worlds, nor does it seem that this problem has some obvious solution. Furthermore, Popper does not pay enough attention to the problem of delineation between the three worlds; he says: “reality consists of three worlds, which are interconnected and act upon each other in some way, and also partially overlap each other” (Popper, p. 8). Penrose says that the three worlds are mutually “profoundly dependent”, but that “there is something distinctly mysterious about the way that these three worlds inter-relate with one another” (Penrose, p. 139). Let us try to express our basic positions in a more precise way.

As a first, we assume that the three worlds are disjunctive; they do not overlap, nor are they mutually ontologically reducible. Second, we assume that there are no causal relations between the three worlds. The realm of the physical is *causally closed*, and we know no other causal relations than the physical ones. Hence, it is possible to speak coherently of causal relations only on the level of the physical world. It could seem that such a position implies that, for example, music cannot affect my physical behaviour; or perhaps that my conscious deliberation about my next move cannot have any effect on my choice. However, the proposed position does not imply such consequences; all it claims is that causal effects take part only in the physical world. Music can have an impact on behaviour, but only when instantiated by means of the physical world, and only through the physical impact of its instantiation on the neural system (through listening, or

thinking to it). Being conscious of some situation does impact a decision. However, the conscious state *itself* emerges from some neural processes, which were triggered and are influenced only by physical causes. And from these processes new mental states are constantly emerging. In short, all causal impacts take part inside *world1*.

Such a position could be said to belong to *epiphenomenalism*, which is often said not to be a plausible theory. According to epiphenomenalism, mental states are products and manifestations of brain activities, but mental states (as subjective phenomena) do not have any causal impact on these activities. Flanagan claims that it is “extremely implausible” that subjective awareness “plays no significant causal role”, although he admits that it is hard to say “exactly what role it plays” (Flanagan, p. 151). We argue that a conscious mental state (as subjective and irreducible to the physical) cannot be said to cause anything by itself; what causes is the *brain state* whose feature and manifestation the conscious state is. The main thesis of this paper is that there can be no understanding and intelligence without *awareness* as a form of consciousness. Furthermore, conscious mental states are considered a world in themselves (*world2*), and the source of the world of creation (*world3*). However, we claim that it is not a *conscious mental state* which really acts (or causes), but the *neural system* whose emergent feature the mental state is. When we say that the conscious mind creates or imposes a structure, we mean that the causal impacts take part on the level of the physical. In other words, it is not consciousness as a subjective state that creates by itself, but the physical system *must have* consciousness as a property to be *motivated and able* to fear, desire, think, understand and create.

We assume that *world3* has been entirely created by the human mind. For example, scientific theories are essentially creations of the human mind; we create a theory, and then we evaluate it on the basis of the effects we manage to obtain with its use. In *world3* belong not only mathematics, science, and technical ideas, but everything which has been created by humans by means of some *system of representation*, as well as the systems of representation themselves. Examples of such are various social structures, customs, works of art and social symbols, which are created by humans and exist on the basis of some system of representation. Humans not only create new entities but also *assign* functions and values to entities in the physical world as well as to created entities. For example, rivers, fruits and flowers belong to *world1*; but their functions and values belong to *world3*. We argue that entities of *world3* cannot create; when instantiated by *world1*, they only explicate that which has been implicitly created by their very creation. For example, an implemented program can produce symbols, shapes, colours or sounds, but its outputs were implicitly cre-

ated by the very creation (and implementation) of the program itself, although many of the outputs may not be created with an intention or a purpose.

4 Computation and Understanding

Discussions about the relationship between computation and understanding from the *Turing test* onwards tend to remain at the level of *effects* (behaviour or functioning). We argue that the questions of understanding, thinking and intelligence should be considered in terms of *motivation*. In other words, an attempt to define or describe these phenomena should start from the plausible sources (causes) of the phenomena themselves. In this connection, we claim that an appropriate definition of *authentic understanding* and *authentic intelligent behaviour* should be based on the concept of *awareness*, without which there are neither valid reasons nor suitable ways to speak of authentic intelligence or understanding at all.

4.1 Understanding as Problem Solving

No behaviour can be said to be “intelligent” independently of its motivations and aims. Hence, the question of the relationship between behaviour and intelligence cannot be resolved in terms of outside effects, but must take into account the motivations and aims. According to Popper, all living beings are constantly preoccupied with solving problems; they are trying to improve their situation, or at least to avoid its deterioration. Popper claims that consciousness was a *problem-solving* consciousness right from its very beginning (Popper, p. 17). Conscious beings encounter and create: they encounter successes and failures, satisfactions and frustrations, joys and sorrows; they create myths, religions, arts, and science in order to explain, solve or celebrate the mysteries of existence. A conscious being creates in order to reduce fears, produce pleasure, add values, communicate feelings, overcome loneliness, relieve tension, establish wholeness (Organ '88, p. 130). It is *tension* that triggers acting and creating. This may be only a vague feeling “that things are not as they ought to be”; but it can also be an awareness of the “fundamental dissatisfaction”, uneasiness, finitude, lostness, loneliness, anxiety, meaninglessness. On the other hand, there is an “incessant hope” for integration, satisfaction, meaning, and harmony (Organ '88, p. 171). Human intelligence has been motivated by the *internal necessity* to solve the problems which it encounters; it has been driven by the impulses to avoid the painful and reach happiness.

According to Penrose, it is not necessary to have the precise definitions of the notions such as conscious-

ness, understanding and intelligence, to be able to observe or define the basic *relationships* between these notions. In this connection, Penrose holds that “intelligence is something which requires understanding”, and that “understanding requires some sort of awareness”; therefore, *intelligence requires awareness*. He holds that to speak of understanding - and hence, also of intelligence - without awareness is “a bit of a nonsense” (Penrose, p. 100). Although Penrose does not elaborate his position in details, it seems that his attitude supports the motivational approach to the phenomena of understanding and intelligence which we propose here. The very existence of aware-less *functional intelligence* depends on the authentic intelligence that *is* aware, as the only *source and measure* of every other quality. For the same reasons, we cannot coherently speak of the creativity of aware-less intelligence, since such an intelligence does not and cannot have any motivation to create: it does not have any aim nor any criterion on the basis of which it could evaluate its own behaviour *as* creating.

4.2 The Care Thesis

The motivational approach to cognition implies that thoughts and intelligent behaviour are not independent of the subjective states (moods), neither in the generative nor in the evaluative sense. A mood opens a specific way for things to show up as mattering, and with that opens new ways of understanding and new possibilities of reasonable (i.e. intelligent) action. This also means that aware-less cognitive abilities are qualitatively limited and fragmentary, and that they exist only as interpretations made by an aware cognitive system. There are many successful expert systems, and we hold that there is no clear limit to the further improvement of machine abilities and skills within the realm of functional replication of human cognitive abilities. However, aware-less machines cannot reach authentic intelligence since for such systems there exist neither problems nor motivations to create. Artificial intelligence is a human creation, and as such it belongs to world3. On the other hand, human intelligence is a feature of world2: a feature which could neither exist nor be evaluated without the existence of the *other* features of world2, such as desire, fear, love and anger. Let us sum up the essence of the motivational approach we proposed, in the following *Care thesis*: human beings, as aware systems, are essentially determined by anxiety and desires: by care. Human cognitive abilities are features of essentially *caring* systems; they spring from care and are shaped by it. Hence, they cannot be fully replicated by a *care-less* (aware-less) system. Computers, as symbol manipulating devices, can be said (although not proved) to be *care-less* systems; hence, such systems cannot understand, be intelligent and creative in the sense in which humans

are.

Discussions about the possibilities and limitations of AI often refer to Heidegger's views of human cognition and behaviour. Heidegger's approach to cognition could be said to be in accordance with the motivational approach proposed here, except for one essential difference. Namely, we hold that awareness is of the fundamental importance for human cognition and behaviour, while Heidegger builds his position about cognition (and about being human, in general) not on awareness but on something he calls *Dasein*. According to Dreyfus, *Dasein* "can mean 'everyday human existence', and so Heidegger uses the term to refer to human being"; however, "we are not to think *Dasein* as a conscious subject"; "*Dasein* must be understood to be more basic than mental states and their intentionality" (Dreyfus, p. 13). In this context, the "understanding of being" is something which "is contained in our knowing-how-to-cope in various domains rather than in a set of beliefs that such and such is the case". In essence, Heidegger's basic position can be reduced to the claim that "we embody an understanding of being that no one has in mind" (Dreyfus, p.18).

We hold that to speak of understanding as aware-less coping does not make much sense, since everybody and everything "copes" somehow. Furthermore, the fact that human understanding does not consist only (or primarily) "in a set of beliefs that such and such is the case" does not imply that: (1) understanding is not related to an aware mind, and (2) that "being there" (of human beings) is not essentially an *aware* state. And if we reject these two Heidegger's assumptions, his sophisticated elaboration of "Daseining" reduces to nearly nothing. In fact, Heidegger himself admits that aware-less coping-with can only last as long as some *problem* ("breakdown", "obstinacy") appears; and then, *Dasein* becomes a *conscious subject* confronted with an obstinate and obtrusive world. Namely, "if the going gets difficult, we must pay attention and so switch to deliberate subject/object intentionality" (Dreyfus, p. 69). But, as we stated by the Care thesis, sentient beings are *always* confronted with "difficulties" (explicit or implicit), so that Heidegger's *Daseining* hardly ever exists; and when it does exist, it seems to be irrelevant. Indeed, human understanding and intelligence can be said to *begin* as awareness that "things are not as they should be". Finally, though in his peculiar terminology, Heidegger says the same thing: "[Care] is to be taken as an ontological structural concept. It has nothing to do with 'tribulation', 'melancholy', or the 'cares of life' ... These - like their opposites, 'gaiety' and 'freedom from care' - are ontically possible only because *Dasein*, when understood *ontologically*, is care" (Heidegger '62, p. 84). We hold that to *care* (or to *be care*) means to be a subject that is aware.

5 Summary

Authentic cognitive abilities are features of an essentially *caring* system: they originate from care and are shaped by it; hence, they cannot be fully replicated by a *care-less* (aware-less) system. We assume that computers are care-less systems, and that they cannot reach authentic understanding and intelligence. On the other hand, we hold that computational systems open virtually unlimited possibilities of replicating human cognitive abilities on the functional (behavioural) level. To develop an artificial system with authentic intelligence, we would first have to create a system which would be "alive" in the sense of being aware and having its own motivations (cares). However, developing such a system should not be considered as a real goal for current AI research. For, in that case, AI could actually not offer anything that could be considered a first step toward such a goal. All too optimistic announcements have been more damaging than beneficial to AI. Instead, machine intelligence and AI achievements should be defined and evaluated in functional terms. On the other hand, if we hold that, for example, an expert system with as much reasoning power in some domain as a human expert, is nevertheless "not really intelligent", then not only do we not have intelligent systems, but we also have no clear idea how such a system could be developed.

References

- [1] Chalmers, J. D.: *The Conscious Mind: In Search of a Fundamental Theory*, Oxford University Press, 1996.
- [2] Churchland, M. P.: *The Engine of Reason, the Seat of the Soul*, The MIT Press, 1995.
- [3] Copeland, J.: *Artificial Intelligence: A Philosophical Introduction*, Blackwell, 1993.
- [4] Dennett, D. C.: *Consciousness Explained*, Penguin Books, 1993.
- [5] Dreyfus, L. H.: *Being-in-the-World: A Commentary on Heidegger's 'Being and Time', Division I*, The MIT Press, 1991.
- [6] Flanagan, O.: *Consciousness Reconsidered*, The MIT Press, 1992.
- [7] Goertzel, B.: Self and Self-Organisation in Complex AI Systems, in Gams, M., Paprzycki, M., Wu, X. (eds): *Mind Versus Computer*, IOS Press / Omsha, 1997.
- [8] Heidegger, M.: *Being and Time*, Harper and Row, 1986.

- [9] Nagel, T.: *The View from Nowhere*, Oxford University Press, 1986.
- [10] Organ, T.: *Philosophy and the Self: East and West*, Susquehanna University Press, 1987.
- [11] Organ, T.: *The Self in Its Worlds: East and West*, Susquehanna University Press, 1988.
- [12] Penrose, R.: *The Large, the Small and the Human Mind*, Cambridge University Press, 1997.
- [13] Popper, K.: *In Search of a Better World*, Routledge, 1992.
- [14] Radovan, M.: Intelligent Systems: Approaches and Limitations, *Informatica*, Vol. 20 (1996), pp. 319-330.
- [15] Radovan, M.: Computation and Understanding, in Gams, M., Paprzycki, M., Wu, X. (eds): *Mind Versus Computer*, IOS Press / Omsha, 1997.
- [16] Searle, R. J.: *The Rediscovery of the Mind*, The MIT Press, 1992.
- [17] Searle, R. J.: *The Construction of Social Reality*, The Free Press, 1995.
- [18] Searle, R. J.: *The Mystery of Consciousness*, The New York Review of Books, 1997.
- [19] Winograd, T.: Preface in Gams, M., Paprzycki, M., Wu, X. (eds): *Mind Versus Computer*, IOS Press / Omsha, 1997.

LFA+: A Fast Chaining Algorithm for Rule-Based Systems

Xindong Wu and Guang Fang
 Department of Software Development, Monash University
 900 Dandenong Road, Melbourne, VIC 3145, Australia
 AND

Matjaž Gams
 Jozef Stefan Institute
 Jamova 39, 1000 Ljubljana, Slovenia
 xindong@insect.sd.monash.edu.au; matjaz.gams@ijs.si

Keywords: expert systems, rule-based systems, fast chaining, conflict resolution

Edited by: Rudi Murn

Received: May 20, 1997

Revised: April 8, 1998

Accepted: July 14, 1998

A significant weakness of rule-based production systems is large computational requirement for performing matching. Time complexity of algorithms is generally still NP-hard (non-polynomial) to the number of rules in a rule base. LFA is a linear-chaining algorithm for rule-based systems which does not require a specific conflict resolution step for chaining. However, its applications are still restricted, e.g., it cannot process first-order rules efficiently.

This paper reviews the design of chaining algorithms for rule-based systems, and analyses some well-known chaining algorithms such as RETE and LFA. The central contribution is the design of a robust LFA algorithm, LFA+, which can process first-order logic rules.

1 Introduction

1.1 Rule-based systems

Rule-based systems (RBSs) are an important type of pattern-directed inference systems. They consist of three basic components as follows:

1. A set of rules, which can be activated or fired by patterns in data.
2. One or more data structures (data bases), which can be examined and modified.
3. An interpreter or inference engine that controls selection and activation of the rules.

A rule includes a left-hand side, LHS, which is responsible for examining items in the data structures, and a right-hand side, RHS, which is responsible for modifying data structures. Data examination consists of comparing patterns associated with the LHSs with elements in the data structures. The patterns may be defined in many ways, such as simple strings, complex graphs, semantic networks, tree structures, or even arbitrary segments of code which are capable of inspecting data elements. Data modification can involve firing actions to modify data, rules, or even the environment. Information in the data can be in the form of lists, trees, nets, rules, or any other useful representation.

The organisation of rule-based systems is modular, and the characteristics of them are as follows [Waterman & Hayes-Roth 78]:

- RBS modules¹ separate permanent knowledge (rules in the rule base) from temporary knowledge (data in the working memory).
- RBS modules are structurally independent. They facilitate incremental expansion of the system and massive code understanding (Modules can be dealt with one by one).
- RBS modules facilitate functional independence. It is generally useful to distribute different functions to different modules.
- RBS modules may be processed by using a variety of control schemes, i.e. different modules may have different control structures.
- RBSs separate data examination from data modification because of the separation of LHSs and RHSs of rules.
- RBSs use rules with a high degree of structure, and are a natural knowledge representation (the natural “IF ... THEN ...” structure).

In the light of problem-solving methods, rule-based systems can be divided into two classes, namely forward-chaining systems and backward-chaining systems. Forward-chaining systems are antecedent-driven, while backward-chaining systems

¹A RBS module is a bundle of mechanisms for examining and modifying one or more data structures.

are consequent-driven. Forward-chaining systems are commonly known as rule-based systems.

Rule-based systems are a well-known type of system in which the control structure can be mapped into a relatively simple recognise-act paradigm. A typical interpreter of a rule-based system performs the following operations in each ‘recognise-act’ cycle:

1. Match

Find out the rule set in the rule base whose LHSs are satisfied by the existing contents of the working memory.

2. Conflict resolution

Select one rule with a satisfied LHS; if no rule has a satisfied LHS then stop.

3. Act

Perform actions in the RHS of the selected rule and go to step 1.

By using suitable interpretations of each of the above actions, the operation of a chaining-based inference engine can be readily described as iteration of such actions. A forward-chaining engine regulates the rules of new databases, while a backward-chaining engine controls the verification of hypothetical information. Another view of the inference engine is that it generates one or more inference nets linking the initial system state to a goal state [Schalkoff 90].

The fundamental operation of the inference engine is the process of matching. Partial matches or complete matches often involve matching with variables for which a suitable unification algorithm which ensures that variable bindings are consistent is necessary. This procedure may require many tests and comparisons. So it is usually difficult to design a fast-chaining algorithm for a large rule-based system.

There are three basic approaches to the problem of conflict resolution in a rule-based system as follows [Rich & Knight 91]:

- Assign preference based on matched rules in the rule base.
- Assign preference based on matched objects in the working memory.
- Assign preference based on actions that matched rules would perform.

1.2 Problems with the ‘Recognize-Act’ Paradigm

For naive rule-based systems, all but the smallest systems are computationally intractable because of the complexity of matching in the 3-phase cycles. The successful match of a rule in the rule base with the working memory does not always mean that the rule will be

fired. A rule may fail to match with the working memory in an overall problem-solving process, but it probably needs to be tested in each 3-phase cycle when the working memory is changed. Meanwhile, some other rule may be successful in matching with the working memory from the very beginning of a problem-solving process, but may fail to receive enough priority to fire in each conflict resolution phase. When there are changes in the working memory, the rule needs to be tested again and again. It has been observed that some systems spend more than nine-tenths of their total run time performing pattern matching in large rule-based systems [Forgy 82]. As a result of these problems, efficiency is a major issue in large rule-based systems.

Since rule-based systems may be expected to exhibit a high standard performance in interactive domains or in real-time domains, many researchers have worked towards improving the efficiency of such systems. As yet, the most significant results have been the RETE algorithm (See Section 4.4) and other RETE-like algorithms such as TREAT (See Section 4.5). These algorithms are match algorithms which avoid matching all rules with the working memory in order to find appropriate rules on each 3-phase cycle so that efficiency can be improved. However, the following two problems still exist in all known rule-based systems except KShell [Wu 93a]:

1. All complete chaining algorithms are exponential in time complexity. Non-worst-case sub-exponential algorithms are not possible for general cases.
2. Chaining in rule-based systems is a much more complicated process than testing the satisfiability of individual propositional formulae. It is not possible to know in advance precisely how many 3-phase “match — conflict resolution — act” cycles are needed for each problem solving task.

In KShell, a new algorithm called LFA (See Section 4.6) has been designed. LFA is a linear forward-chaining algorithm for rule-based systems. The most significant advantages of LFA are that its time complexity is $O(n)$ where n is the number of rules in the rule base, and that it does not need an independent conflict resolution step. By using a two-level “rule schema + rule body” structure (See Section 3.2.2), knowledge representation in KShell can explicitly express numeric computation and inexact calculus in the same way as inference rules in rule bodies. As long as knowledge representation has an applicable extension, and processing measures show further improvements, LFA should achieve a wider range of applications. However, its knowledge representation cannot represent first-order logic rules efficiently. This is a significant restriction for applications.

The research objective of this paper is to relax the above mentioned limitation of LFA so that it can effi-

ciently process first-order logic rules. We will present the design of a robust LFA algorithm, LFA+, based on LFA [Wu 93a], which has the following components:

- Extended knowledge representation for first-order logic rules, which includes specific representations of recursive rules and rules with negative condition elements.
- Sorting measures, for ordering the knowledge in a knowledge base.
- Linear forward chaining.

The paper is organised as follows. Section 2 introduces expert system principles and some concepts of the first-order logic language, and explains one definition for describing the LFA+ algorithm. In Section 3, knowledge representation issues are addressed, and two languages for rule-based systems — OPS5 and *rule schema + rule body* are described and compared. Section 4 discusses algorithm design issues and techniques, and analyses the RETE, TREAT and LFA algorithms. In Section 5, knowledge representation measures, sorting strategies, the chaining procedure and analyses of the LFA+ algorithm are presented in detail. Finally, Section 6 outlines conclusions and future research. Definitions are listed in the Appendix.

2 Background in Expert Systems and First Order Logic

2.1 Expert systems

An "expert system" is a computer program which uses knowledge and inference procedures to solve problems that are difficult enough to require human expertise for their solutions [Raeth 90]. Expert system technology aims at improving qualitative factors and can provide expert-level performance to complex problems. A typical expert system consists mainly of the following parts:

- A working memory/data base, which stores the evidence and intermediate results of problems during the chaining process.
- A knowledge base (KB) or knowledge source.
- An inference/chaining engine for solving users' problems by applying the knowledge encoded in the knowledge base.
- An explanation engine or tracing engine for telling the users how the solutions were obtained.
- A knowledge acquisition engine for acquiring knowledge or modifying the knowledge base when necessary.

- A knowledge base management subsystem that detects inconsistencies in the KB.

Their relationships are shown in Figure 1.

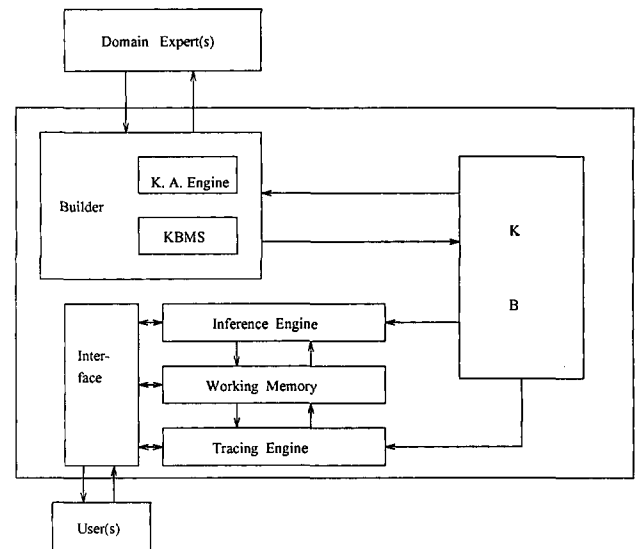


Figure 1: An expert system structure

Conventional software programs are designed to control computers algorithmically and tell the computers exactly what to do in problem-solving. These programs are usually procedural. Once a program system has been encoded, it is difficult to change the system design. On the other hand, expert systems excel at encoding knowledge declaratively, and they can be modified flexibly because of the separations of knowledge from expert system shells and knowledge from data, and their modular structures. Furthermore, expert systems have the following features [Pedersen 89]:

- They use symbols to encode the world which can be used in varied ways.
- Most expert systems support uncertainty representation.
- Expert systems can handle unknown cases of a problem by applying the knowledge in the knowledge base.
- They can explain their reasoning.
- They can make multiple conclusions.
- They can tailor conclusions.

2.2 Language of first-order logic

A first-order language is identified by a triple $\langle V, F, P \rangle$:

- V is a set of variables.

- F is a set of functors, each of which has an arity.
- P is a set of predicate symbols, each of which has an arity.

The *terms* (See the definition below) of the language are built from variables and functions (Constants are viewed as functors of arity 0), while *predicates* are built from terms and predicate symbols (Propositions are viewed as predicate symbols of arity 0).

Definition: A *term* is defined inductively as follows:

- A variable is a term.
- A constant is a term.
- If f is an n -ary function symbol and t_1, \dots, t_n are terms, then $f(t_1, \dots, t_n)$ is a term.

Interpretation

Truth value interpretation of a first-order logic language is a triple $\langle D, F, R \rangle$:

- D is the domain.
- F is a mapping from functions of domain elements to the domain.
- R is a mapping from predicates of domain elements to truth values.

Horn clauses are a subset of first-order logic languages, but the subset is powerful enough to encode Turing machines. A Horn clause has the following form:

$$p(t) :- q_1(t_1), q_2(t_2), \dots, q_n(t_n).$$

where p and q_1, q_2, \dots, q_n are predicate letters, $n \geq 0$, and all variables which occur in the terms t, t_1, t_2, \dots, t_n are universally quantified at the front of the clause (implicitly). If n is 0 then the clause is referred to as a *fact*, otherwise, it is called a *rule*.

The atom $p(t)$ is referred to as the head of the clause, and $q_1(t_1), q_2(t_2), \dots, q_n(t_n)$ as the body of the clause. The terms t, t_1, t_2, \dots, t_n may be arbitrary terms, and hence may contain variables and/or functions.

A logic program is a set of Horn clauses. However, it is often useful to consider sub-classes of this class of programs in rule-based systems. One type of these programs is a Datalog program, in which terms are only allowed to be either variables or constants.

2.3 Unification and Match

Unification

Unification is the basis of the uses of logical inference in artificial intelligence. It is a method of finding such variable bindings for two predicates or terms that they can be identical [Sterling et al. 86].

Match

Two terms *match* if [Bratko 90]:

- they are identical, or
- the variables in both terms can be instantiated to objects in such a way that after the substitution of variables by these objects the terms become identical.

The following is an extended definition, *partial match*, for describing LFA+ (See Section 5.2).

Definition — Partial match

Given a premise factor, *p-factor*, of one rule schema and a conclusion factor, *c-factor*, of another rule schema, partial match of *p-factor* with *c-factor*, written as $\text{partial-match}(p\text{-factor}, c\text{-factor})$, has the following meanings:

- If *p-factor* is a variable then *c-factor* is the same variable.
- If *p-factor* is a proposition p or $\text{not}(p)$ then *c-factor* is p or $\text{not}(p)$.
- If *p-factor* is a predicate $p(\dots)$ or $\text{not}(p(\dots))$ then *c-factor* is $p(\dots)$ or $\text{not}(p(\dots))$.

3 Knowledge representation

3.1 Introduction

In order to solve complex problems encountered in AI, a considerable amount of knowledge, as well as some mechanisms for manipulating knowledge, are necessary. Barr and Feigenbaum identify four types of knowledge as follows [Miranker 87]:

- Objects, i.e. nouns and adjectives that describe them.
- Events: Object interaction.
- Performance: How to do something, also known as procedural knowledge.
- Meta-knowledge: Knowledge about knowledge.

Knowledge plays two roles in AI programs as follows:

- It may define the search space and the criteria for determining a solution to a problem.
- It may improve the efficiency of a reasoning procedure by informing an inference procedure of the best places to look for a solution.

Knowledge representation occurs at two levels [Rich & Knight 91]:

- Data level, at which facts are described.
- Symbol level, in which representations of objects at the data level are defined in terms of symbols that can be manipulated by programs, such as PROLOG rules.

Knowledge representation and search are the two main themes of AI problem solving but they are not independent issues. If a particular search method is applied, and a method of knowledge representation may represent the problem more easily and it more efficiently supports the operations required by the search strategy, a particular problem may be more easily solved. For a particular problem, different combinations of knowledge representation methods and search may yield more or less effective means for solving the problem. The next section will focus on representing knowledge by using rules.

3.2 Knowledge representations for rule-based systems

The use of rules for encoding knowledge is a particularly important issue because rule-based reasoning systems have played a very important role in AI evolution from a purely laboratory science into a commercially significant one. This section outlines two representation methods namely OPS5 and "rule schema + rule body", which have been applied in some rule-based systems.

3.2.1 OPS5

OPS5 [Forgy 82] is a rule-based system language. An OPS5 rule comprises the following:

1. Symbol P.
2. Rule name.
3. Left-hand side (LHS).
4. Symbol \rightarrow
5. Right-hand side (RHS).

All of these are enclosed in parentheses.

A typical LHS structure is as follows: { <object> (computer \uparrow Name <name> \uparrow price <cost>)}. This structure is used to describe computer objects; it includes computer name and price.

RHS structure is similar to LHS structure, but contains an action, e.g. 'modify' before <money> (See below).

The \uparrow is the OPS5 operator that distinguishes attributes from values.

A variable is a symbol beginning with the character '<' and ending with the character '>', e.g. <object>.

The predicates in OPS5 include =, <>, <, >, <=, >=. A predicate is placed between an attribute and a value.

The following is a typical rule from [Brownston et al. 86]:

```
(P have-enough-money-to-buy-computer
 { <object> (computer  $\uparrow$ name <name>
  $\uparrow$ price <cost>)}
 { <money> (saving-account  $\uparrow$ balance
 {<balance>  $\geq$  <cost> })}
  $\rightarrow$ 
 (modify <money>  $\uparrow$ balance
 (computer <balance> - <cost>)))
```

where the meanings are apparent.

3.2.2 Rule schema + rule body

"Rule schema + rule body" [Wu 93a] is a 2-level method of knowledge representation. A rule schema is used to describe the hierarchy among factors or nodes in a reasoning network. A rule body consists of computing rules and/or inference rules and is used to express specific evaluation methods for factors and/or certainty factors in corresponding rule schemata.

A rule schema has the general form:

IF E_1, E_2, \dots, E_n THEN A,

where E_1, E_2, \dots, E_n is a conjunction (AND) of all premise factors and A is a predicate or variable called a conclusion factor.

Each rule schema has a corresponding rule body. In a rule body, there are one or more inference rules such as production rules and/or computing rules for computation.

A rule schema with its corresponding rule body is called a *rule set*. A rule set is an independent knowledge unit in the "rule schema + rule body" representation and can be described in *Backus Naur* form as follows:

```
<rule set> := <rule set number><rule
 schema><rule body>
<rule set number> := <integer>
<rule schema> := 'IF' <premise
 factors> 'THEN' <conclusion factor>
<premise factors> := <premise factor>
 {','<premise factors>}
<premise factor> := <factor>
<conclusion factor> := <factor>
<factor> := <logic
 assertion>|<variable name>
<logic assertion> := <predicate(object)>
<rule body> := (<C-rule>|<I-rule>)
 {<rule body>}
<C-rule> := (<factor>|
 (CF('<factor>')))'='
 <assignment expression>
<assignment expression> :=
 <value>|<algebraic expression>
<I-rule> := 'IF' <antecedents> 'THEN'
 <conclusion>
<antecedents> := <antecedent> {'and'
 <antecedents>}
<antecedent> := (<factor>|
```

```

CF('(<factor>'))
<relational symbol>
<assignment expression>
<relational symbol> :=
'>' | '<' | '=' | '<>' |
'>=' | '<='
<conclusion> := <C-rule>
<value> := <integer> |
<real> | <symbolic value> |
<probability> | <fuzzy value>

```

The terms <variable>, <predicate(object)>, <algebraic expression> and different kinds of values above have the standard interpretations.

3.3 Comparison between OPS5 rules and "rule schema + rule body" representation rules

There are a number of advantages with the "rule schema + rule body" representation [Wu 93a]. Firstly, rule schemata in a knowledge base provide a way of describing meta-knowledge about concrete rules in rule bodies which facilitate sorting rule sets in a rule base. Secondly, it expresses computing rule sets in the same form as inference rule sets. Further, it provides natural "IF-THEN" expertise expression in two-level structures, and also provides flexible processing of inexact reasoning in rule bodies, and so forth.

However, the negative aspect of these advantages is that "rule schema + rule body" representation is not so powerful as OPS5 rules. For example, it cannot efficiently represent first-order logic rules.

4 Design of rule-based system algorithms

Usually an algorithm refers to a method of solving a well-specified computational problem for a system. With the development of rule-based systems, efficiency has been a major consideration up to this stage. A rule-based algorithm is considered more efficient than others if its cost for per working memory change is lower. This section addresses some measures for improving the efficiency of rule-based system algorithms and analyses some good algorithms [Fang & Wu 94].

4.1 The knowledge

For the purpose of obtaining efficiency, three types of knowledge or state information may be incorporated into a rule-based system algorithm as follows [McDermott et al. 78]:

1. *Condition membership*, which provides knowledge about the possible satisfaction of each individual

condition element. An algorithm that uses condition membership can ignore further processing of those rules which are not active, i.e. those rules that one or more positive condition elements are not partially satisfied.

2. *Memory support*, which provides knowledge about which working memory elements individually partially satisfy each individual condition element. Associated with each condition element in rule-based systems is a memory which indicates precisely which subset of working memory elements partially match the condition element.
3. *Condition relationship*, which provides knowledge about the interaction of condition elements within a rule, and partial satisfaction of rules. The process for condition relationship is similar to that of maintaining the results of intermediate joins in database systems.

Two further types of knowledge can be identified as follows:

4. *Conflict set support*, which provides knowledge about which rule has consistent variable bindings between its condition elements. The conflict set is retained across each 3-phase cycle, and the contents of the conflict set are used to limit search in a rule-based system during the chaining process [Miranker 87].
5. *Premise-conclusion relationship*, which provides knowledge about which premise factor of a rule schema is the conclusion factor of another rule schema or which premise factor of a rule schema partially matches (See the definition in Section 2.3) with the conclusion factor of another rule schema. This knowledge is used to arrange rule sets in the rule base into order [Wu 93a].

4.2 Matching techniques

In each 3-phase cycle, matching is a crucial step. This section introduces four techniques for matching, namely indexing, filtering, and decision tree and decision table methods.

4.2.1 Indexing

In the matching process, the current state can be used as an index for immediate selection of matching rules, provided that the rule preconditions are stated as exact descriptions of the current state [Rich & Knight 91]. The simplest form of indexing for rule-based systems is that the interpreter begins the match process by extracting one or more features from each working memory element and uses these features to hash into the rule collection. This obtains a set of rules that might have satisfied LHSs. A more efficient form of indexing adds memory to the process. For example, one scheme involves storing a count with

each condition element. The counts are all set to zero when the system starts the execution. When a data element enters the working memory, all condition elements matching the data element have their counts increased by one. When a data element leaves the working memory, all condition elements matching the data element have their counts decreased by one. The interpreter deals with those LHSs that have non-zero counts for all their positive condition elements. This scheme has been combined into a few algorithms with other efficient measures [Forgy 82].

4.2.2 Filtering

Filtering is a method which uses a filter namely a body of code that uses the knowledge sources (KSs) introduced in the last section to reduce the number of rules tested by a rule-based system. If a filter contains enough information, a significant number of rules can be excluded from consideration. A filter admits to further testing of any subset of rules that may be unsatisfied by its KSs.

Filters usually are in the form of discrimination nets of which the famous RETE and TREAT algorithms are the best examples. RETE incorporates memory-support and condition-relationship; whereas TREAT takes one more knowledge source into account— conflict set support (See Section 4.4). RETE and TREAT will be further analysed in the following sections.

4.2.3 The decision tree method

The decision tree method compiles the set of condition elements which are defined in the form of lists into a near-optimal decision tree [Malik 81]. Firstly, it restricts a segment variable (e.g. @), which represents list fragments of an undefined length, to the tail of a sublist, and assumes that any datum in the working memory necessarily matches some condition elements of rules. And it treats two kinds of element features: those which compute the length of a list or a sublist, and those which extract an atom from some specified position. These tasks are done by the internal nodes of the tree. A leaf of the tree is a pointer that points to a stack which contains all data that match the corresponding condition elements. However, sometimes there exist overlapping cases in which some leaves contain more than one condition element. These considerations lead to an algorithm. It starts by selecting an 'efficient' feature which is defined everywhere as root of the tree from the discriminating feature table (built from condition element features). In the table '-' and '@' denote undefined values and variable feature values respectively. And then it recursively lets each branch have a label which corresponds to a subtable of the identification table. The recursion stops if some subtable has an empty set. An example from [Malik 81] is as follows:

Assume five rule antecedents are given:
 R1: (A(B C x)) (D x) (x F y) (E F y) --> ...
 R2: (A(B C x)) ~(D x) (x F y) (B F y) ~(C B @) --> ...
 R3: (A(x F y)) (H x) (x F z) --> ...
 R4: (A(B F @)) (B F y) (Z C B)} --> ...
 R5: (A(x B)) (x C B)} --> ...

where x, y and z are variables, and capital letters represent propositions. An identification table which plots these feature values v.s. the set of condition elements is constructed in Figure 2.

	(A(BCx))	(A(xFy))	(A(BF@))	(A(xB))	(Dx)	(Hx)	(BFx)	(xCB)	(EFx)	(CB@)	(xFy)
b	2	2	2	2	2	2	3	3	3	@	3
2	3	3	@	2	@	@	0	0	0	0	0
<1>	A	A	A	A	D	H	B	@	E	C	@
<2>	-	-	-	-	@	@	F	C	F	B	F
<2.1>	B	@	B	@	-	-	-	-	-	-	-
<2.2>	C	F	F	B	-	-	-	-	-	-	-

Figure 2: An Identification Table

After non-discriminating features (e.g. |2| with 0 values) are deleted from the table, a discriminating decision tree can be built in Figure 3.

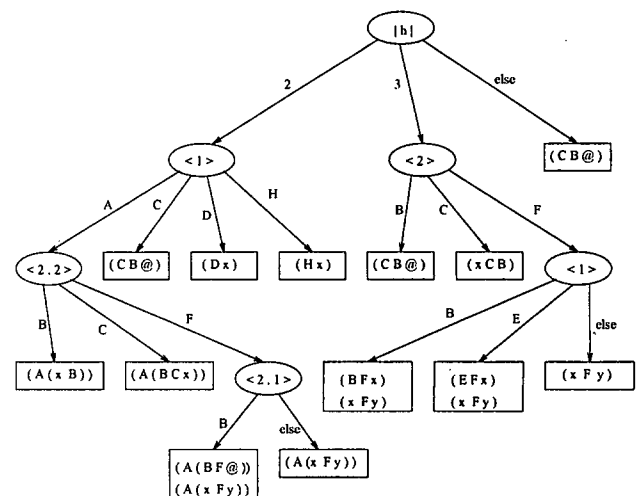


Figure 3: A Discriminating Decision Tree

At the beginning of each "match - conflict resolution - act" cycle, the interpreter traverses the tree with each modified datum in the working memory, computing a feature value at each tree node and selecting the branch corresponding to the value. So a leaf containing the condition element can be reached. Characteristic of the decision tree method is that it avoids the redundant computations.

This method is suited to rule-based systems where condition elements of rules are represented in the form of lists. Taking into account the tree optimisation possibilities, the potential performance of the method seems to be promising.

4.2.4 The decision table method

This method is based on the table knowledge representation in [Colomb 89], and it is mainly suited to propositional rule-based systems. The transformation of the rule set in a rule base to a table representation involves a transformation algorithm which eliminates all rows in the table which are inconsistent and all rows which are subsumed by other rows. Its chaining procedure can be described as in Figure 4.

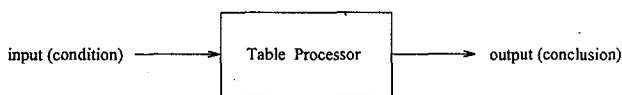


Figure 4: The chaining procedure of the decision table method

A table consists of rows which can be viewed as assignments of values to corresponding variables. A simple example of a table is as follows:

```

it-has-two-legs, fly, it-is-a-bird
it-has-four-legs, it-is-an-animal
  
```

When the input is *it-has-two-legs* and *fly*, the first row in the table will fire, and the output of the table processor (i.e. conclusion) will be *it-is-a-bird*; similarly, when the input is *it-has-four-legs*, the second row will fire, and the output of the table processor will be *it-is-an-animal*. Usually, rows in the decision table have interpretations as consequents such as the case of Garvan ES1 system.

Because of the replacement of intermediate assertions with expressions which imply them and the negation processing in the transformation procedure, the size of a knowledge table may be explosively large. In order to solve this problem, a few algorithms have been designed for reducing ambiguity and redundancy in [Colomb & Chung 90]. These algorithms can greatly reduce the response time of a system. In addition, an unambiguous table can be further transformed into a decision tree by ID3-like algorithms [Quinlan 86, Wu 93b]. A decision tree executes a number of nodes logarithmic in the number of rows in the decision table. Therefore, the decision table method can process large propositional rule-based systems efficiently.

4.3 Conflict resolution

The output from the matching process, and the input to the conflict resolution, is a set referred to as conflict set. All rules in which LHSs have been satisfied by

working memory elements can be identified by conflict set elements, which are termed instantiations. An instantiation is an ordered pair of a rule name and a list of working memory elements matching the condition elements of the rule. It is the job of conflict resolution to find an instantiation which will be executed in the act phase of a cycle.

A conflict-resolution strategy is a coordinated set of principles for making selections among competing instantiations. A rule-based system's performance depends on its conflict-resolution strategy for both sensitivity and stability [Brownston et al. 86]. Sensitivity is the fast degree by which a system responds to the dynamically changing demands of its environment, while stability is its continuity of behaviour. The following principles can be applied for any conflict resolution strategy [Brownston et al. 86]:

– Refraction

Refraction prevents rules from firing on the same data more than once. The intention is to avoid the trivial form of infinite looping which might occur when a rule does not change the working memory contents.

– Data ordering

Data ordering which orders data by recency or activation is a basic principle of conflict resolution and a powerful way of adding sensitivity to a conflict-resolution strategy. It gives preference to rules that match those elements most recently added to working memory or that are strongly related to recently-added data. This principle is usually combined with other principles to narrow down the selection of one instantiation to fire next.

– Specificity ordering

The specificity principle gives preference to rules that are more specific according to some standard which can be measured in a variety of ways. For example, one specificity principle depends on a specificity function that is correlated with the complexity degree of rule condition elements.

– Rule ordering

Rule ordering (which tends to be less sensitive) provides static ordering of a rule set independent of the way in which rules are instantiated by data. The ordering may be computed by using some rule feature/features. Either total or partial ordering can be given by a relation on rules. If total rule ordering has been provided, the rules can be stored in the order and scanned linearly until a matching one is found. LFA [Wu 93a] is a successful example of using the rule ordering strategy.

– Arbitrary choice and parallel selection

None of the above principles can guarantee that only a single instantiation will remain in the conflict set. If single firing is required for each cycle, an arbitrary decision referred to as arbitrary choice ordering can be made after all conflict-resolution principles have been applied; however, in some systems especially parallel systems, all the remained instantiations can be fired in one cycle, which is called parallelism in firing.

The following are two alternative conflict-resolution strategies for OPS5 systems — LEX and MEA [Forgy 81]:

LEX

The LEX conflict-resolution strategy includes four steps which are applied in order to find an instantiation:

1. Discard from the conflict set those instantiations that have already fired. If there are no instantiations that have not fired, conflict resolution fails and no instantiation is selected.
2. This step partially orders the remaining instantiations in the conflict set on the basis of recency of working memory elements by using the following algorithm to compare pairs of instantiations: Compare the most recent elements from two instantiations. If one element is more recent than the other, the instantiation containing that element dominates. If the two elements are equally recent, compare the second most recent element from the instantiations. Continue in this way until either one element of one instantiation is found to more recent than the corresponding element in the other instantiation, or no element remains for one instantiation. If one instantiation is exhausted before the other, the other dominates. If the two instantiations are exhausted at the same time, neither dominates.
3. If no one instantiation in particular dominates all others under the previous step, this principle is necessary for comparing the dominant instantiations on the basis of the specificity of the LHSs of the rules. Count the number of tests (for constants and variables) that have to be made in finding an instantiation for the LHS. The LHSs that require more tests dominate.
4. If no single instantiation dominates after the previous step, make an arbitrary selection of one instantiation as the dominant instantiation.

MEA

The MEA strategy differs from that of LEX in that another step has been added after the first step in

LEX. It places extra emphasis on the recency of a working memory element matching the first condition element of a rule. If no single instantiation dominates, then the remaining set is passed through the same sequence of orderings as in LEX.

4.4 RETE

The RETE match algorithm [Forgy 82] is an algorithm for computing the conflict set. It improves matching efficiency by incorporating memory-support and condition-relationship to avoid iterating computations over the working memory and the rule base; its concrete measure is using a sorting network (Subparts of which can be shared) which is compiled from the condition element patterns of rules to test features of data elements, and to store information. The structural form of the sorting network is given in Figure 5.

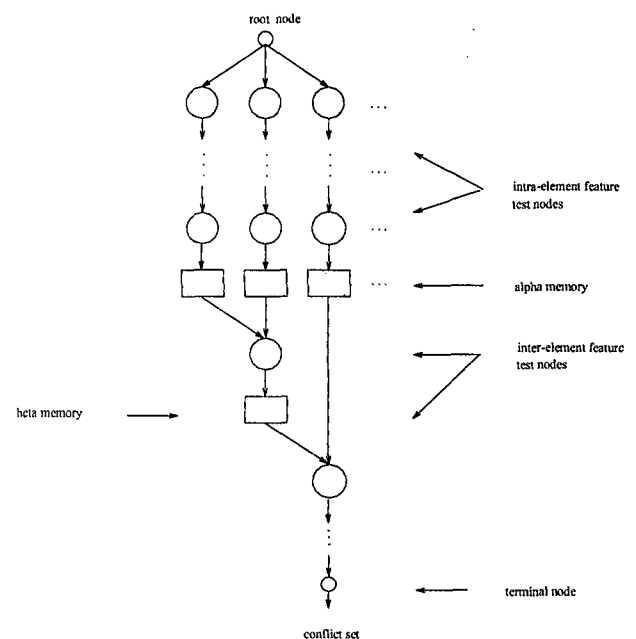


Figure 5: The Structural Form of RETE's Sorting Networks

RETE deals with two types of element features i.e. intra-element features and inter-element features. Intra-element features are the features that involve one working memory element. For example, the class of an element must be 'Expression'; the value of an 'OP' attribute must be '+'. However, inter-element features result from having a variable occur in more than one pattern. For instance, the value of an attribute of an element must be equal to the value of an attribute of another element.

As shown in Figure 5, when the pattern compiler processes an LHS, it builds a chain of intra-element feature test nodes, which are one-input, for each condition element pattern of the LHS based on the intra-element features which are required by the condition

element pattern. And then it builds inter-element feature test nodes for testing the inter-element features of the LHS. The inter-element feature test nodes are two-input and left-associative [Ho & Marshall 92]. Finally it builds a terminal node to represent the production rule.

The match procedure is as follows: The root node receives a token, which is state information (The tags '+' and '-' in a token indicate how the state information is to be changed), and then passes the copy of the token to all its successors i.e. the intra-element feature test nodes. A '+' token that has satisfied the intra-element feature tests is added to the alpha memory. A '-' token that has satisfied the intra-element feature tests has a corresponding '+' token that already presents in the alpha memory. The corresponding '+' token is removed. Once a token updates an alpha memory, it continues to go through the network and the next node is an inter-element feature test node. Inter-element feature test nodes store the first token and wait until the second one arrives, and then compare them. If they find that the variables between the two tokens are bound consistently, they join the two tokens into a bigger one. The bigger token is stored in the beta memory, and then sent to another inter-element feature test node for other consistent tests of variable bindings (if possible). If all the variable bindings are consistent for an LHS, the final token is sent to the terminal node. The terminal node receives the token, and adds the rule instantiation (Which is an ordered pair of the form <rule, list of data elements matched by the LHS of the rule>) of the LHS to the conflict set.

Over the naive matching algorithms, advantages of the RETE match algorithm can be summarised as follows:

- It does not need the interpretive step by using the sorting network.
- Sufficient state is maintained so that it can avoid many iterated computations.
- The subparts of the network for similar condition element patterns can be shared.

There are also some disadvantages inherent to the RETE match algorithm:

- It is just a matching algorithm.
- Time complexity of RETE is NP-hard to the number of rules in the rule base.
- The removal of data elements performs such operations for adding the data elements that the deletions of working memory elements are expensive.
- It is inefficient when most of the data changes in each cycle, because in that case RETE needs to

maintain its state between cycles i.e. it cannot efficiently process non-redundant rule-based systems.

4.5 TREAT

The TREAT match algorithm [Miranker 87] is a RETE-like algorithm. It not only makes use of condition membership and memory support knowledge sources, but also combines them with a new source of information, *conflict set support*. Its significant features are that (1) in some cases it performs much better than the RETE algorithm, and (2) it can be used in parallel systems.

The TREAT algorithm constructs a sorting network from the condition patterns of the rule set. But no subpart of the network can be shared by more than one condition pattern. Furthermore, it adopts the following measures:

– Conflict Set Support

TREAT retains the conflict set across system cycles and uses its contexts to reduce the number of comparisons required to find consistent variable bindings [Miranker 87]. As a result, it reduces the computations between beta memories that the RETE algorithm needs.

– Handling Negated Condition Elements

When a data element which partially matches a positive condition element is added into the working memory, the conflict set remains the same, except that the addition of the working element results in new instantiations. If a rule is active (See *Condition support*) and the new instantiations contain the new working memory element, then the instantiations are added into the conflict set.

When a working memory element which partially matches a negative condition element is deleted, no new rules will be instantiated. In that case, the instantiations that contain the removed working memory element will be invalidated and are removed from the conflict set.

When a rule firing adds a working memory element that partially matches a negated condition element, there may be some rule instantiations that are invalidated and will have to be removed from the conflict set. In this case, the invalidated instantiations will not contain the working memory element. To find the instantiations which must be removed from the conflict set, the negated condition element which is partially matched is temporarily transformed to be positive to form a new rule. The working memory element is used as a seed to build instantiations of this new rule. Then the new instantiations are compared with the conflict set. If any instantiation exists in the conflict set, then remove them.

When a working memory element is removed and it partially matches a negated condition element, if there is no other similar data element whose variable bindings are consistent with those of the removed one in the working memory, it may cause some rule instantiations to enter the conflict set.

– Memory Support

The alpha memories forming the memory support part of the TREAT match algorithm are the same as those of the RETE match algorithm. Information related to each condition element is stored in arrays indexed by CE-num's (condition element numbers). All the condition elements in a rule-based system are numbered with CE-num's. The alpha memories are partitioned into *old-mem*, *new-delete-mem* and *new-add-mem* as three separate vectors. The addition and deletion of a working memory element are different from those of the RETE match algorithm. This can be seen from the algorithm illustration below.

– Condition Support

Associated with each rule is a rule-active property. A rule is active if each of its positive condition elements is partially matched by some working memory elements. The rule-active property of a rule is affected by updating the contents of the *old-mem* for the rule.

The advantages of the TREAT algorithm are as follows:

- The deletion of elements is simpler than in RETE.
- Inactive rules are ignored.
- It can handle both temporally redundant and non-redundant rule-based systems.
- It can be easily implemented in parallel rule-based systems.

However, TREAT also has some disadvantages:

- No computing results can be shared by condition patterns or rules.
- The time complexity for matching is still NP-hard as RETE.
- It is also a matching algorithm.

4.6 LFA

Unlike RETE-like algorithms, the LFA algorithm [Wu 93a] is a linear forward-chaining algorithm. It adopts a 2-level "rule schema + rule body" knowledge representation outlined in Section 3.2.2. The major features of the LFA algorithm are that chaining is carried out in 2-phase "match-act" cycles instead of the 3-phase "match – conflict resolution – act" cycles, and

it can choose one rule set in each cycle without any specific conflict resolution.

The following outlines the concrete measures that the LFA algorithm has adopted:

– "Rule Schema + Rule Body"

"Rule schema + rule body" represents knowledge in two levels. Rule schemata describe the hierarchy among factors (include premise factors and conclusion factors) or nodes in a reasoning network. Rule bodies, which consist of computing rules and/or inference rules, are used to express specific computing methods for the factors and/or certainty factors in their corresponding rule schemas. This 2-level structure facilitates sorting the knowledge base and avoids matching all the rules in a knowledge base with the working memory when some piece of data is not available.

– Sorting the Knowledge Base [Wu 93a]

At the end of knowledge acquisition or knowledge modification, the knowledge base is sorted or compiled into a partial order: If rule schema N is *if factor-1, factor-2, ..., factor-n, then factor*, then all the schemas with *factor-1, factor-2, ..., factor-n* as their conclusion factors have rule-set numbers smaller than N.

Other sorting measures are as follows:

1. Processing dead cycles.

A cycle like *if A then B, if B then C, and if C then A* in a domain reasoning network is called a dead cycle if none of A, B, and C is a leaf node in the domain reasoning network and there is no other rule schema whose conclusion factor is one of them. A dead cycle cannot be numbered and has to be changed to a live cycle or removed. Figure 6 is an example of dead cycles.

2. Renumbering schemata.

Renumber all rule schemata which have all of their premise factors being leaf nodes in the domain reasoning network. For any factor F, if all schemata with it as their conclusion factor have been renumbered, it is treated as a leaf node for further renumbering. If all rule schemata in a knowledge base have been renumbered then stop.

3. Resolving live cycles.

A cycle like *if A then B, if B then A, and if C then A* is called a live cycle and A is called a live node in the live cycle if C is not involved in any dead cycle (Figure 7 shows a live cycle). A live cycle can be resolved by treating one of its live nodes as a leaf node for further renumbering. Resolve all live cycles and goto 2.

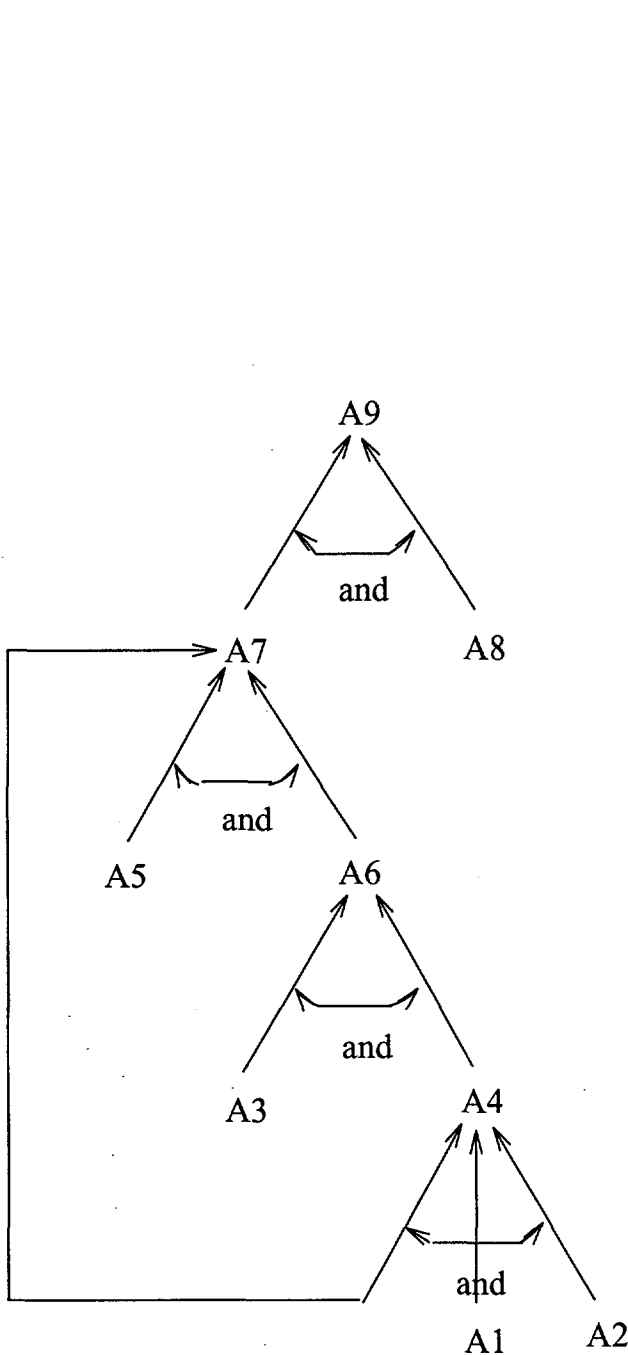


Figure 6: A dead cycle

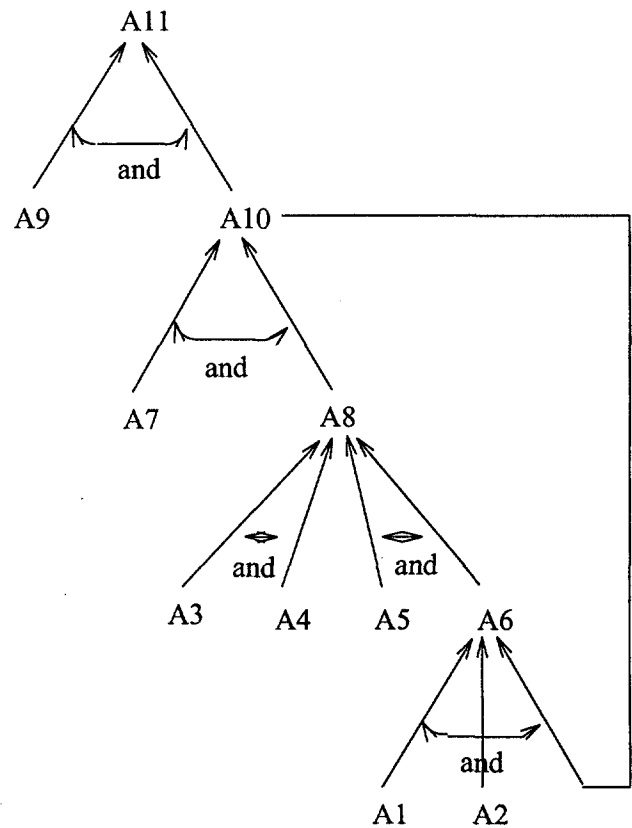


Figure 7: A live cycle

- Linear Forward Chaining

After knowledge compilation, chaining is performed as follows:

```

FOR the first TO the last renumbered
  schema in the knowledge base
DO
  IF there exist data in the working
    memory for each of the
    condition factors of the schema
  THEN fire the corresponding
    rule body of the schema.
ENDFOR.
    
```

Advantages of the LFA algorithm:

- Time complexity is $O(n)$ where n is the number of rules in a knowledge base.
- It is a complete forward-chaining algorithm, in which the conflict resolution step is unnecessary.
- Natural knowledge representation.
- Computational knowledge and uncertainty calculus are integrated with logic inference in the 2-level knowledge representation.
- Intermediate computing results can be shared.

Disadvantages of the LFA algorithm:

- First-order rules can not be efficiently processed.
- Its chaining is in a fixed order, and thus all problem evidence needs to be provided at the beginning of chaining in order to ensure that the inference can be accomplished.

5 LFA+: A robust LFA algorithm

For rule-based systems, LFA is the best forward-chaining algorithm to date in terms of time complexity. However, as with other algorithms, it also has its own limitations. The significant drawback which greatly restricts application is that LFA cannot efficiently deal with first-order logic rules. Based on the original LFA algorithm, this section describes a robust LFA algorithm — LFA+, which mainly tackles this problem and efficiently deals with recursive rules and negation.

The idea to transform first-order logic rules into simpler domains was used before in inductive logic programming [Lavrac & Dzeroski] and in empirical learning in order to transform more complex expressive mechanisms into e.g. attribute-value descriptions [Gams et al. 91].

5.1 Knowledge representation

Knowledge representation adopts the basic “Rule Schema + Rule body” structure (See Section 3.2.2), but expressiveness is extended to include first-order logic rules. In addition, some other measures are presented for the following purposes:

1. It facilitates avoiding matching all the rules in a rule base with the working memory at run time when some piece of data is not available.
2. It supports ordering the knowledge.

- Predicate representation

A predicate can be represented in the following form:

`<predicate.symbol>(object-list).`

In the object-list, an object is a constant or a variable.

- Rule schemata representation

A rule schema takes the general form:

IF *factor1*, *factor2*, ..., *factorn* THEN *factor*

where *factor1*, *factor2*, ..., *factorn*, *factor* may be variables for computing rules or logic assertions for inference rules. But any variable in logic assertions is replaced by the ‘_’ notation which means “don’t care” in rule schemata. The *factor* can also be an algebraic function or an action defined to modify the working memory.

- Negation representation

Negation representation takes the form: `not(p)`, where `p` represents a predicate. Its meaning is defined as that of [Bratko 90] — If `p` cannot be proven to be TRUE, then `not(p)` is TRUE.

Two constraints are given to this representation:

1. ‘not(p)’ and ‘p’ cannot appear in two different rules respectively within a rule set. For example, for

`p(X,Y) :- s(X,Y), not(r(X)).`
`p(X,Y) :- s(X,Y), r(Y).`

there are two rule sets:

Rule schema: IF `s(-,-)`, `not(r(-))` THEN `p(-,-)`
 Rule body: IF `s(X,Y)` and `not(r(X))` THEN `p(X,Y)`

and

Rule schema: IF `s(-,-)`, `r(-)` THEN `p(-,-)`
 Rule body: IF `s(X,Y)` and `r(Y)` THEN `p(X,Y)`

Thereby, the confused representation that `not(r(-))` and `r(-)` appear in the same schema can be avoided, and when data do not exist for `r(-)`, the matching procedure for the rule body in the second rule set is unnecessary at run time.

2. ‘not(p)’ and ‘p’ may appear in the same inference rule in a rule set, but only the ‘p’ is included in the corresponding premise factors of the rule schema. For instance, the rule set for

`p(X,Y,Z) :- s(X,Y,Z), not(r(X,Y)), r(Y,Z).`

is

Rule schema: IF `s(-,-,-)`, `r(-,-)` THEN `p(-,-,-)`
 Rule body: IF `s(X,Y,Z)` and `not(r(X,Y))` and `r(Y,Z)` THEN `p(X,Y,Z)`

This knowledge representation can be described in extended *Backus Naur form* as follows:

```

<rule set> := <rule set number><rule
  schema><rule body>
<rule set number> := <integer>
<rule schema> := ‘IF’ <premise
  factors> ‘THEN’ <conclusion factor>
<premise factors> := <premise
  factor> {‘,’ <premise factors>}
<premise factor> := <factor>
<conclusion factor> := <factor>
<factor> := <logic
  assertion>|<variable>|< algebraic
  function>|<action>
<logic assertion> :=

```

```

<predicate>|(not('<predicate>'))
<predicate> := <predicate
  symbol> {'('<object-list>')}
<object-list> := <object> {','
  <object-list>}
<object> := <constant>|<variable>|'_'
<algebraic function> :=
  (functor('<variable-list>'))
<variable-list> := <variable>
  {','<variable-list>}
<rule body> := (<C-rule>|<I-rule>)
  {'<rule body>}
<C-rule> := (<factor>|
  (CF('<factor>')))'='
<value>|<algebraic expression>
<I-rule> := 'IF' <antecedents> 'THEN'
  <conclusion>
<antecedents> := <antecedent> {'and'
  <antecedents>}
<antecedent> :=<logic
  assertion>|<relation expression>
<relation expression> :=
  (<factor>|(CF('<factor>'))|
  <value>|
  <algebraic expression>
  <relation symbol>
  (<factor>|(CF('<factor>'))|<value>|
  <algebraic expression>)
<value> := <integer>
  |<real> |
  <symbolic value> |
  <probability>|<fuzzy value>
<relation symbol> :=
  '>','<','='| '<>' | '>='| '<='
<conclusion> :=
<logic assertion>|<action>|<C-rule>

```

- Recursive rule representation

In the above form, the further specific representation for recursive rules may be described as follows:

```

<I-rule> := <I-rule-1><I-rule-2>
<I-rule-1> := 'IF' <antecedents 1> 'THEN'
  <conclusion 1> {'<I-rule-1>}
<I-rule-2> := 'IF' <antecedents 2> 'and'
  <antecedent 2> 'THEN'
  <conclusion 2> {'<I-rule-2>}

```

Here, <conclusion 1>, <antecedent 2> and <conclusion 2> have the same predicate symbol.

In rule schemata, if <antecedents 1> and <antecedents 2> become the same, and so do <conclusion 1>, <antecedent 2> and <conclusion 2>, and there is only one rule for <I-rule-1>, then <I-rule-1> and <I-rule-2> can be put into one rule set, and the rule schema is as follows:

IF <antecedents 1> THEN <conclusion 1>
 Otherwise they have to be broken into different rule sets, but these rule sets as a whole take part in the numbering process. Their schemata together are referred to as a recursive schema set.

For example, the rule set for

```

ancestor(X,Y) :- father(X,Y).
ancestor(X,Y) :- father(X,Z),
  ancestor(Z,Y).
ancestor(X,Y) :- father(Z,Y),
  ancestor(X,Z).

```

is as follows:

```

rule schema:
IF father(.,.) THEN ancestor(.,.)
rule body:
IF father(X,Y) THEN ancestor(X,Y)
IF father(X,Z) and ancestor(Z,Y) THEN
  ancestor(X,Y)
IF father(Z,Y) and ancestor(X,Z) THEN
  ancestor(X,Y)

```

However, the rule sets for

```

parent(X,Y) :- mother(X,Y).
parent(X,Y) :- father(X,Y).
parent(X,Y) :- sister(Z,Y), parent(X,Z).
parent(X,Y) :- brother(Z,y),
  parent(X,Z).

```

are

```

rule schema:
IF mother(.,.) THEN parent(.,.)
rule body:
IF mother(X,Y) THEN parent(X,Y)

```

```

rule schema:
IF father(.,.) THEN parent(.,.)
rule body :
IF father(X,Y) THEN parent(X,Y)

```

```

rule schema:
IF sister(.,.), parent(.,.) THEN
  parent(.,.)
rule body:
IF sister(Z,Y) and parent(X,Z) THEN
  parent(X,Y)

```

and

```

rule schema:
IF brother(.,.), parent(.,.) THEN
  parent(.,.)
rule body:
IF brother(Z,Y) and parent(X,Z) THEN
  parent(X,Y)

```

As long as the <object> is defined as follows:

```

<object> := <object
  function>|<variable>|'_'
<object function> := <object functor>
  {'('<variable list>')}

```

clearly, this knowledge representation can represent any Horn clause rules.

5.2 The LFA+ algorithm

This algorithm consists of two procedures namely sorting the knowledge in a knowledge base and linear-forward chaining.

• Sorting knowledge in a knowledge base

This procedure aims at placing rule sets in the rule base in order. The order of the rule sets is the order of the rule schemata. All measures adopted are described as follows:

– Processing of negations

Some schemata may have been separated due to the constraints of negation representation. These schemata together are called a schema set. In this case, the schema set as a whole takes part in the numbering process (See next step). Inside the schema set, all schemata take the order of the rule-body rules in situations that it is unnecessary to distinguish not(p) and p and in the order all rule-body rules can be put into a rule set.

– Schema numbering

If rule schema #N is

IF *factor1*, *factor2*, ..., *factorn* THEN *factor*

then all schemata whose conclusion factors partially match (See Section 2.3) with any of the *factor1*, *factor2*, ..., *factorn* have rule set numbers smaller than N. This process puts the schemata into partial order.

– Processing of live cycles

A cycle such as “IF a1 THEN b1, IF b2 THEN a2, and IF c THEN a3”, in which a1, a2 and a3 partially match with each other, and b1 partially matches with b2, is referred to as a live cycle and a1, a2 and a3 become a live node in the live cycle if c is not involved in any dead cycles (e.g. a live cycle in Figure 8). Schemata in a live cycle can be numbered by starting from one of the live nodes of the live cycle.

– Processing of dead cycles

A cycle such as “IF a1 THEN b1, IF b2 THEN c1, and IF c2 THEN a2”, in which a2 partially matches with a1, b1 partially matches with b2 and c1 partially matches with c2, is referred to as a dead cycle if none of a1, a2, b1, b2, c1 and c2 can be instantiated (Such as in Figure 9). A dead cycle cannot be numbered — it has to be changed into a live cycle or removed.

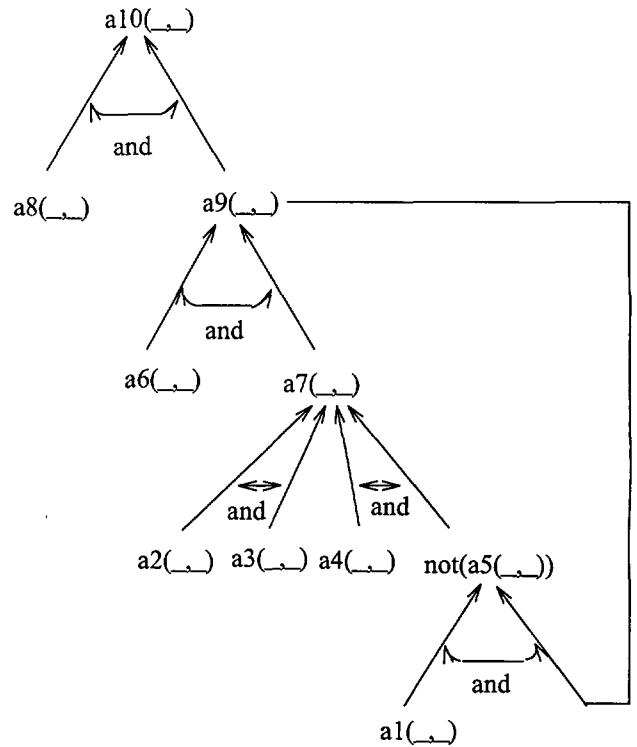


Figure 8: A live cycle

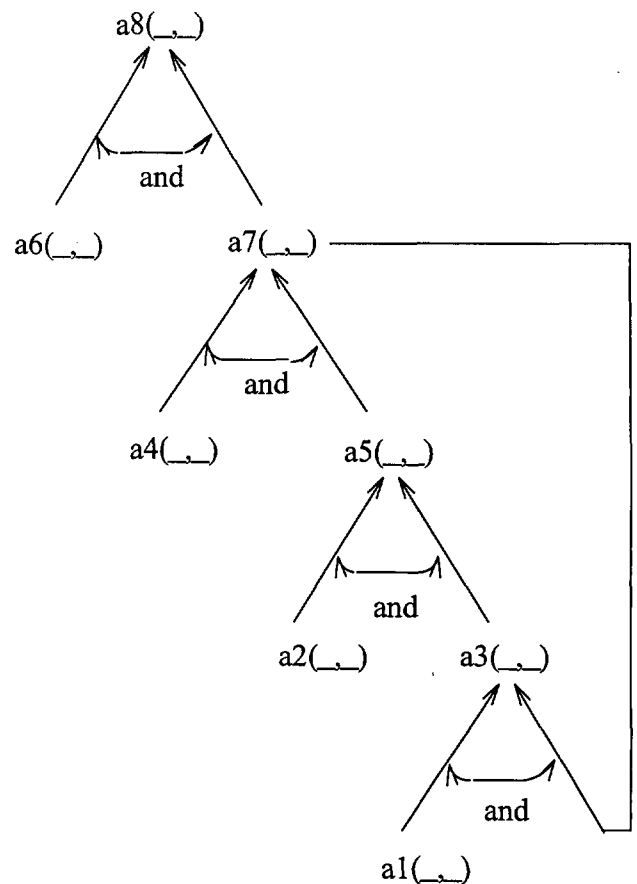


Figure 9: A dead cycle

- Processing of parallel schema set

If some schemata have the same conclusion factor (A negation schema set is treated as one schema) and are not involved in any dead cycles or any live cycles, then all these schemata together are referred to as a parallel schema set. A parallel schema set as a whole takes part in the numbering process, but inside the parallel schema set, the order of the schemata is arbitrary.

- Rule schema renumbering

Renumbering the schemata in a knowledge base until they are all in order. If they are already in order then stop.

• Linear-forward-chaining

After sorting knowledge in the knowledge base, the LFA+ algorithm performs the following process:

Loop: from the first to the last schema do

If data exist in the working memory for each of the premise factors (But for not(p), p may or may not exist) in the schema, then fire the corresponding rule body of the schema

Endloop

5.3 Advantages and disadvantages

The time complexity of the LFA+ algorithm is also $O(n)$, where n is the number of rules in the rule base, as the original LFA algorithm. Yet, with the original LFA algorithm, the robust LFA+ algorithm has the following advantages:

- It can deal with first-order logic rules.
- Recursive inference rules can be efficiently processed.
- It facilitates coping with rules that include negative condition elements.

However, the LFA+ algorithm also has some disadvantages in common with other algorithms:

- It cannot meet the dynamic requirements to the knowledge base during inference.
- Its chaining is in a fixed order and thus all problem evidence needs to be provided at the beginning of chaining in order to ensure that the inference can be accomplished.

5.4 An example

Suppose the following Prolog rules are given:

```

sibling(X,Y) :- brother(X,Y). (1)
sibling(X,Y) :- sister(X,Y). (2)
sibling(X,Y) :- brother(Y,X). (3)
sibling(X,Y) :- sister(Y,X). (4)
parent(X,Y) :- father(X,Y). (5)
parent(X,Y) :- mother(X,Y). (6)
ancestor(X,Y) :- parent(X,Y). (7)
parent(X,Y) :-
    bling(Z,Y), parent(X,Z). (8)
ancestor(X,Y) :-
    parent(Z,Y), ancestor(X,Z). (9)
sibling(X,Y) :-
    brother(Z,Y), sibling(X,Z),
    X \== Y. (10)
sibling(X,Y) :-
    sister(Z,Y), sibling(X,Z),
    X \== Y. (11)
sibling(X,Y) :-
    brother(Y,Z), sibling(X,Z),
    X \== Y. (12)
sibling(X,Y) :-
    sister(Y,Z), sibling(X,Z)
    X \== Y. (13)

```

Based on the knowledge representation method of LFA+, their corresponding rule schemata can be represented as follows:

```

IF brother(,_ ,_) THEN sibling(,_ ,_) (1')
IF sister(,_ ,_) THEN sibling(,_ ,_) (2')
IF brother(,_ ,_) THEN sibling(,_ ,_) (3')
IF sister(,_ ,_) THEN sibling(,_ ,_) (4')
IF father(,_ ,_) THEN parent(,_ ,_) (5')
IF mother(,_ ,_) THEN parent(,_ ,_) (6')
IF parent(,_ ,_) THEN ancestor(,_ ,_) (7')
IF sibling(,_ ,_),parent(,_ ,_)
    THEN parent(,_ ,_) (8')
IF parent(,_ ,_),ancestor(,_ ,_)
    THEN ancestor(,_ ,_) (9')
IF brother(,_ ,_),sibling(,_ ,_)
    THEN sibling(,_ ,_) (10')
IF sister(,_ ,_),sibling(,_ ,_)
    THEN sibling(,_ ,_) (11')
IF brother(,_ ,_),sibling(,_ ,_)
    THEN sibling(,_ ,_) (12')
IF sister(,_ ,_),sibling(,_ ,_)
    THEN sibling(,_ ,_) (13')

```

As shown in (10'), (11'), (12'), and (13'), variables in relational expressions of variables ($X \neq Y$) are not represented in schemata, because they are determined by the same variables in corresponding predicates. Since (1') and (3'), (2') and (4'), (10') and (12'), and (11') and (13') are the same respectively, they can be combined. So they become the following:

```

IF brother(,_ ,_) THEN sibling(,_ ,_) (1'')
IF sister(,_ ,_) THEN sibling(,_ ,_) (2'')

```

```

IF father(,_ ) THEN parent(,_ ) (3")
IF mother(,_ ) THEN parent(,_ ) (4")
IF parent(,_ ) THEN ancestor(,_ ) (5")
IF sibling(,_ ),parent(,_ )
  THEN parent(,_ ) (6")
IF parent(,_ ),ancestor(,_ )
  THEN ancestor(,_ ) (7")
IF brother(,_ ),sibling(,_ )
  THEN sibling(,_ ) (8")
IF sister(,_ ),sibling(,_ )
  THEN sibling(,_ ) (9")

```

Apparently, (1") and (2") are a parallel schema set, so are (3") and (4"). (1"), (2"), (8") and (9") form a kind of recursive schema set, so do (3"), (4") and (6"); (5") and (7") form another kind of recursive schema set, they can be combined. Based on the schema numbering measures of LFA+, these schemata can be put into the following order:

```

1 IF brother(,_ ) THEN sibling(,_ )
2 IF sister(,_ ) THEN sibling(,_ )
3 IF brother(,_ ),sibling(,_ )
  THEN sibling(,_ )
4 IF sister(,_ ),sibling(,_ )
  THEN sibling(,_ )
5 IF IF father(,_ ) THEN parent(,_ )
6 IF mother(,_ ) THEN parent(,_ )
7 IF sibling(,_ ),parent(,_ )
  THEN parent(,_ )
8 IF parent(,_ ) THEN ancestor(,_ )

```

where No. 8 is the combination of (5") and (7").

Finally, the ordered rule sets are as follows:

```

1 IF brother(,_ ) THEN sibling(,_ )
  IF brother(X,Y) THEN sibling(X,Y)
  IF brother(X,Y) THEN sibling(Y,X)
2 IF sister(,_ ) THEN sibling(,_ )
  IF sister(X,Y) THEN sibling(X,Y)
  IF sister(X,Y) THEN sibling(Y,X)
3 IF brother(,_ ),sibling(,_ )
  THEN sibling(,_ )
  IF brother(Z,Y) and sibling(X,Z) and
  X \== Y THEN sibling(X,Y)
  IF brother(Y,Z) and sibling(X,Z)
  and X \== Y THEN sibling(X,Y)
4 IF sister(,_ ),sibling(,_ )
  THEN sibling(,_ )
  IF sister(Z,Y) and sibling(X,Z)
  THEN sibling(X,Y)
  IF sister(Y,Z) and sibling(X,Z)
  THEN sibling(X,Y)
5 IF father(,_ ) THEN parent(,_ )
  IF father(X,Y) THEN parent(X,Y)
6 IF mother(,_ ) THEN parent(,_ )
  IF mother(X,Y) THEN parent(X,Y)
7 IF sibling(,_ ),parent(,_ )

```

```

  THEN parent(,_ )
  IF sibling(Z,Y) and parent(X,Z)
  THEN parent(X,Y)
8 IF parent(,_ ) THEN ancestor(,_ )
  IF parent(X,Y) THEN ancestor(X,Y)
  IF parent(Z,Y) and ancestor(X,Z)
  THEN ancestor(X,Z)

```

5.5 A chaining implementation of LFA+

The implementation is carried out with Sicstus Prolog on DEC stations. It consists of two parts namely receiving_data and chaining. Main program 'LFA+' has three parameters: Interp.f, Leaf_file and Kb_file.

Interp.f is an interpretation file, which has the following contents:

1. One or more domain value lists. For example, in domain([a,b,...]), the list is a value range of object variables; whereas in domain('~Y',[1,2,...]), the list is the value range of the variable '~Y'.
2. A truth value interpretation list. E.g. r_map([map(ancestor(adam,john,true)),...]).

Leaf_file includes the predicates which need to be provided by the users, and Kb_file is a rule file which contains all the rules for a problem solving.

The following is an implementation example. For convenience, we keep the Prolog rules as rule bodies since the order of rule sets is the order of rule schemata. So the contents of the Kb_file are as follows:

```

1.
'IF'. brother(,_ ). 'THEN'. sibling(,_ ).
  sibling(X,Y) :- brother(X,Y).
  sibling(X,Y) :- brother(Y,X).
2.
'IF'. sister(,_ ). 'THEN'. sibling(,_ ).
  sibling(X,Y) :- sister(X,Y).
  sibling(X,Y) :- sister(Y,X).
3.
'IF'. brother(,_ ). ','. sibling(,_ ).
  'THEN'. sibling(,_ ).
  sibling(X,Y) :- brother(Z,Y),
  sibling(X,Z),X \== Y.
  sibling(X,Y) :- brother(Y,Z),
  sibling(X,Z), X \== Y.
4.
'IF'. sister(,_ ). ','. sibling(,_ ).
  'THEN'. sibling(,_ ).
  sibling(X,Y) :- sister(Z,Y),
  sibling(X,Z),X \== Y.
  sibling(X,Y) :- sister(Y,Z),
  sibling(X,Z),X \== Y.
5.
'IF'. father(,_ ). 'THEN'. parent(,_ ).
  parent(X,Y) :- father(X,Y).

```

```

6.
'IF'. mother(_,_). 'THEN'. parent(_,_).
parent(X,Y) :- mother(X,Y).
7.
'IF'. sibling(_,_). ','. parent(_,_).
'THEN'. parent(_,_).
parent(X,Y) :- sibling(Z,Y),parent(X,Z).
8.
'IF'. parent(_,_). 'THEN'. ancestor(_,_).
ancestor(X,Y) :- parent(X,Y).
ancestor(X,Y) :- parent(Z,Y),
ancestor(X,Z).

```

Suppose the contents of the Leaf_file are the following:

```

brother('~X1', '~Y1').
sister('~X2', '~Y2').
father('~X3', '~Y3').
mother('~X4', '~Y4').

```

and the contents of the interpretation file are as follows:

```

domain([adam,eve,david,doris,john,mary,
edgar,fred,lucy,margaret,violet,
patrick]).
r_map([map(sister(doris,john),true),
map(sister(margaret,fred),true),
map(sister(lucy,edgar),true),
map(sister(margaret,violet),true),
map(sister(margaret,patrick),true),
map(sister(margaret,fred),true),
map(sister(violet,fred),true),
map(sister(violet,margaret),true),
map(sister(violet,patrick),true),
map(brother(fred,violet),true),
map(brother(fred,patrick),true),
map(brother(patrick,fred),true),
map(brother(patrick,margaret),true),
map(brother(patrick,violet),true),
map(brother(edgar,lucy),true),
map(brother(fred,margaret),true),
map(brother(john,doris),true),
map(father(adam,doris),true),
map(father(adam,john),true),
map(father(david,edgar),true),
map(father(david,lucy),true),
map(father(john,fred),true),
map(father(john,margaret),true),
map(mother(eve,john),true),
map(mother(eve,doris),true),
map(mother(doris,edgar),true),
map(mother(doris,lucy),true),
map(mother(mary,fred),true),
map(mother(mary,margaret),true)]).

```

The following records the example run.

```

>sicstus
SICStus 2.1 #9: Thu Apr 21 09:39:25 +1000
| ?-
| ?-consult('datalog.tex').
{consulting /fang/project/datalog.tex...}
{Undefined predicates will just fail}
yes
| ?- 'LFA+'(interp_file,leaf_file,kb_file).
{consulting /fang/project/interp_file...}
{/fang/project/interp_file consulted,
67 msec 1632 bytes}
Interpretation interp_file loaded.
Can you provide a value for ~X1 (y/n/q)?
y.
input:john.
Can you provide a value for ~Y1 (y/n/q)?
y.
input:diris.
Wrong value!
Can you provide a value for ~Y1 (y/n/q)?
y.
input:doris.
More values for prev. variables (y/n/q)?
y.
New value for ~X1 (y/n)?
y.
input:fred.
New value for ~Y1 (y/n)?n.
More values for prev. variables (y/n/q)?
n.
Can you provide a value for ~X2 (y/n/q)?
y.
input:doris.
Can you provide a value for ~Y2 (y/n/q)?
y.
input:john.
More values for prev. variables (y/n/q)?
n.
Can you provide a value for ~X3 (y/n/q)?
y.
input:adam.
Can you provide a value for ~Y3 (y/n/q)?
y.
input:john.
More values for prev. variables (y/n/q)?
q.
chaining ... 1: ->sibling(john,doris)->
sibling(doris,john) 2:
3: ->F->F 4: 5: ->parent(adam,john)
6: 7: ->parent(adam,doris)
8: ->ancestor(adam,john)->
ancestor(adam,doris) !.

```

As shown above, there are not data for rule sets 2, 4 and 6; rules in rule set 3 are false, in which there are two rules in the rule body.

6 Conclusions and future research

With the development of rule-based expert systems, efficiency has been a major consideration for chaining algorithms [Fang & Wu 94]. The naive approaches are to combine indexing with direct interpretation of the LHSs in the rule base. They are inefficient in dealing with large knowledge bases.

The RETE match algorithm has made some significant improvements. It compares a set of LHSs of rules with a set of data elements in the working memory to compute the conflict set, and does not need the interpretive step. The indexing function is represented as a network of simple feature recognisers. This algorithm can efficiently process the conflict set, since it does not iterate over the working memory and the rule base. However, the RETE match algorithm only incorporates memory support and condition relationship knowledge sources. It still has significant disadvantages. For example, deletion of working memory elements is expensive, and time complexity is NP-hard (non-polynomial) to the number of rules in a knowledge base.

The TREAT algorithm is a RETE-like algorithm. It makes use of condition membership, memory support, and conflict set support knowledge sources. The obvious improvement is that it can be easily adopted in parallel systems. However, in some cases its performance is worse than that by using RETE.

LFA is the best chaining algorithm up to date in terms of theoretical time complexity. By adopting rule ordering method, its time complexity of chaining can be $O(n)$ where n is the number of rules in a knowledge base. This advantage results from its knowledge representation method namely the 2-level “rule schema + rule body” knowledge representation (See Section 3.2.2), and using premise-conclusion knowledge to compile the rule set in the knowledge base. However, it is difficult to deal with first-order logic rules by using LFA.

LFA+ is a robust forward-chaining algorithm. It can process first-order logic rules efficiently. This mainly benefits from its knowledge representation. LFA+ inherits LFA’s knowledge representation method, namely represents knowledge in a 2-level “rule schema + rule body” structure. But it has been extended to cover first-order logic rules (See Subsection 5.1). Based on this representation, LFA+’s sorting and chaining procedures for first-order logic rules can be the same ones of LFA for processing propositional logic rules. Therefore, LFA+ is a powerful linear-chaining algorithm for rule-based expert systems.

However, due to its static rule ordering method, its chaining is in a fixed order. So all problem evidence must be provided at the beginning of chaining in order to ensure that the inference can be accomplished for

problem solving. This is a restriction to its application in data sensitive rule-based systems which give preferences to those rules that match the most recent data elements added to the working memory.

LFA+ is well suitable to be implemented by using logic programming language tools. When it is implemented by using imperative language tools, it has not provided memory support for avoiding iterating computations for matching working memory elements with condition elements when data exist for the corresponding rule schemata, so processing efficiency will decrease.

For future research, it is important to maintain LFA+’s linear performance. In order to extend its application, the following two directions may be considered:

- Combining other sorting measures with the 2-level “rule schema + rule body” knowledge representation method or modifying the “rule schema + rule body” structure if necessary in order that it can meet the requirements of data sensitive systems.
- Introducing the memory support knowledge source to LFA+. This involves building a mechanism related to how to organise those memories and how to efficiently locate the memories.

Also, meeting the dynamic demands of rule-based systems (i.e. the knowledge can be changed at run time) is a challenge. This is a common problem in all known chaining algorithms for rule-based systems. For example, RETE-like algorithms do not allow knowledge modification at run time either.

References

- [Avron and Edward 81] Avron Barr and Edward A. Feigenbaum, *The Handbook of Artificial Intelligence, Vol. I*, Heuris Tech Press, 1981
- [Bratko 90] Ivan Bratko, *PROLOG — Programming for Artificial Intelligence*, Second Edition, Addison-Wesley Publishing Company, 1990
- [Brownston et al. 86] Lee Brownston, Robert Farrell, Elaine Kant and Nancy Martin, *Programming Expert Systems in OPS5 — An Introduction to Rule-based Programming*, Addison-Wesley Publishing Company, Inc, 1986
- [Colomb 89] R. M. Colomb *Representation of Propositional Expert Systems as Decision Tables*, Proceedings of the 3rd Australian Joint Conference on Artificial Intelligence, 1989
- [Colomb & Chung 90] R. M. Colomb and C. Y. C. Chung *Ambiguity and Redundancy Analysis of a Propositional Expert System*, Proceedings of the

- 4th Australian Joint Conference on Artificial Intelligence, 1990
- [Fang & Wu 94] Guang Fang and Xindong Wu, *Chaining in Rule-based Systems*, Proceedings of the 7th Australian Joint Conference on Artificial Intelligence, 1994, 575-582
- [Forgy 81] C.L. Forgy, *OPS5 User's Manual*, Department of Computer Science, Carnegie-Mellon University, 1981
- [Forgy 82] C.L. Forgy, *A fast algorithm for the many pattern/many object pattern match problem*, Artificial Intelligence, 19(1982), 17-27
- [Gams et al. 91] Matjaz Gams, Matija Drobnic and Marko Petkovsek, *International Journal of Man-Machine Studies*, 34(1991),49-68
- [Ho & Marshall 92] Ho Soo Lee and Marshall I. Schor, *Match Algorithms for Generalised RETE Networks*, Artificial Intelligence, 54(1992), 249-274
- [Kacsuk 90] Peter Kacsuk, *Execution Models of PROLOG for Parallel Computers*, Pitman, London, 1990
- [Lavrac & Dzeroski] Nada Lavrac and Saso Dzeroski, *Inductive Logic Programming: Techniques and Applications*, Ellis Howood, (1994).
- [Malik 81] Malik Ghallab, *Decision Trees for Optimising Pattern-Matching Algorithms in Production Systems*, Proceedings of the 7th Inter. Joint Conf. on AI, 1981, 310-312
- [McDermott & Forgy 78] J. McDermott and C. Forgy, *Production System Conflict Resolution Strategies*, Academic Press, Inc., 1978
- [McDermott et al. 78] J. McDermott, A. Newell and J. Moore, *The Efficiency of Certain Production System Implementations*, Academic Press, Inc., 1978
- [Miranker 87] D.P. Miranker, *TREAT: A New and Efficient Match Algorithm for AI Production Systems*, PhD Thesis, Columbia University, 1987
- [Pedersen 89] Ken Pedersen, *Expert Systems Programming — Practical Techniques for Rule-based Systems*, John Wiley & Sons, Inc, 1989
- [Quinlan 86] J. R. Quinlan, *Induction of Decision Trees*, Machine Learning Vol 1, No. 1, 1986
- [Raeth 90] Peter G. Raeth, *Expert Systems — A Software Methodology for Modern Applications*, IEEE Computer Society Press, 1990
- [Rich & Knight 91] Elaine Rich and Kevin Knight, *Artificial Intelligence*(Inter. Edition(2)), McGraw-Hill Inc., 1991
- [Schalkoff 90] Robert J. Schalkoff, *Artificial Intelligence — An Engineering Approach*, McGraw-Hill Inc. 1990
- [Sterling et al. 86] Leon Sterling and Ehud Shapiro, *The Art of PROLOG — Advanced Programming Techniques*, The MIT Press, 1990
- [Steven 90] Steven L. Tanimoto, *The Elements of Artificial Intelligence — Using Common Lisp*, Computer Science Press, 1990
- [Waterman & Hayes-Roth 78] D. A. Waterman and Frederick Hayes-Roth, *An Overview of Pattern-directed Inferences Systems*, Academic Press Inc., 1978
- [Wu 93a] Xindong Wu, *LFA: A Linear Forward-chaining Algorithm for AI Production Systems*, Expert System: The Int. J. of Knowledge Engineering, 10(1993), 4: 237-242
- [Wu 93b] Xindong Wu, *Inductive Learning: Algorithms and Frontiers*, *Artificial Intelligence Review*, 7(1993), 2: 93-108

Appendix – Definitions

Backus-Naur form

This expression refers to a formal language for context-free grammars. A grammar consists of a set of rewrite rules, each of which has a *left-hand side* and a *right-hand side*, separated by the metalanguage symbol ::= . The left-hand side of each rule is a nonterminal symbol of the grammar, while the right-hand side is a sequence of nonterminal and terminal symbols. Nonterminal symbols are usually surrounded by angle brackets < and > .

Binding

This term refers to the association between a variable and a value for the variable that holds within some scope, such as the scope of a rule, function call, or procedure invocation.

Bound

A variable that has been assigned a value by the process of binding is said to be bound to that value.

Certainty factors (CFs)

Certainty factors are properties associated with attribute/value pairs and rules, commonly used to represent uncertainty, or likelihood. Certainty factors are usually automatically maintained by expert system shells.

Condition elements

The *left-hand side* of a rule in a rule-based system is sometimes expressed as a set of patterns (or templates) which are to be matched against the contents of the working memory; each such pattern is called a condition element. When a rule is instantiated, each condition element is found to match one element of the working memory.

Conflict set

A conflict set is a set of all instantiations generated by the match process during a recognise-act cycle. The process of conflict resolution selects one instantiation from the conflict set and fires it.

Cycle

A cycle is a single iteration of a loop. In production systems, an execution consists of iterated recognise-act cycles.

Domain reasoning network

A domain reasoning network (e.g. Figure 8 in Section 5.2) is an AND/OR tree associated with a knowledge base in *rule schema + rule body* by the following analogies:

1. Nodes in the tree correspond to factors in the knowledge base.
2. A rule schema such as *IF E1, ..., En THEN A* in the knowledge base corresponds to the arcs, which indicate the hierarchy among factors in the tree (shown in Figure 10).

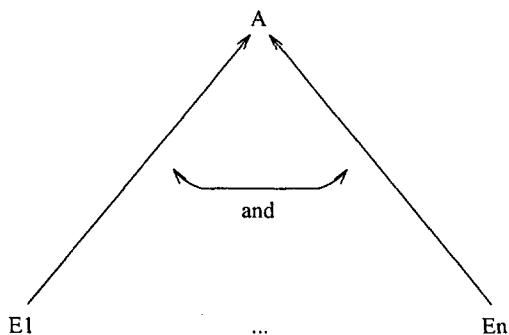


Figure 10: A rule schema

Filtering

The exclusion of either data (data filtering) or rules (rule filtering) from the match process for the sake of efficiency is termed filtering.

Fire

This term means to execute the set of actions specified in the right-hand side of an instantiation of a rule.

Forward chaining

Forward-chaining is a problem-solving method that starts with initial evidence of a problem and applies inference rules to generate new evidence until either one

of the inferences satisfies a goal or no further inferences can be made. In forward-chaining production systems, the applicability of a rule is determined by matching the conditions specified on the left-hand side against the evidence currently stored in working memory.

Instantiation

Instantiation refers to a pattern or formula in which variables have been replaced by constants. In a production system, an instantiation is the result of successfully matching a rule against the working memory contents. It can be represented as an ordered pair of which the first member identifies the rule that has been satisfied, while the second member is a list of working memory elements that match the condition elements of the rule.

Interpreter

The interpreter is a part of a production system that executes the rules.

Match

In a production system the match process compares a set of patterns from the left-hand sides of rules against the data in data memory to find all possible ways in which the rules can be satisfied with consistent bindings (i.e. instantiations).

Mixed chaining

A search strategy which uses both backward and forward chaining during a single processing of a knowledge base is known as mixed chaining.

Object

An object is an entity in a programming system that is used to represent declarative knowledge and possible procedural knowledge about a physical object, concept, or problem-solving strategy.

One-input node

This term refers to a node in the RETE match algorithm network associated with a test of a single attribute of a condition element. It passes a token if and only if the attribute test is satisfied.

Pattern

A pattern is an abstract description of a datum that places some constraints on the value(s) it may assume, but needs not specify it in complete detail.

Temporal redundancy

This term refers to the tendency of rule systems to make relatively few changes to the working memory, and hence to the conflict set, from one recognise-act cycle to the next. The RETE algorithm exploits temporal redundancy so as to avoid unnecessary recomputations of all matches.

Two-input node

Two-input nodes are nodes in the RETE algorithm network that merge matches for a condition element with matches for all preceding condition elements.

Forecasting from Low Quality Data with Applications in Weather Forecasting

Bavy Li and James Liu

The Hong Kong Polytechnic University, Hung Hom, Kowloon, HK

Phone: +852 2766 7273, Fax: +852 2774 0842

E-mail: {csnlli,csnkliu}@comp.polyu.edu.hk

AND

Honghua Dai

The University of New England, Armidale, NSW 2351, AUST

Phone: +612 6773 3182, Fax: +612 6773 3312

E-mail: hdai@cs.une.edu.au

Keywords: machine learning, naive Bayes, neural network

Edited by: Xindong Wu

Received: March 9, 1997

Revised: April 15, 1998

Accepted: July 29, 1998

Accurate prediction is the most important issue in the study of machine learning and knowledge discovery. Various learning approaches fail to achieve a higher accuracy in new unseen cases due to various causes. Low quality data is one of them. This paper reports the results of applying Naive Bayesian classifiers to real meteorological data in Hong Kong to learn weather forecasting rules for the prediction of binary classification problems, such as rain/no-rain, and for the prediction of unlimited classification problems such as the prediction of precipitation. The comparison results show that among the methods we compared, Backpropagation Network (BPN) achieved a better accuracy rate on binary classification problems of the rain/no-rain average, while the Naive Bayesian Network with initial probability density approximation (NBN-IPD) achieved the highest rate of the average of the three classes and achieved a very competitive rate on unlimited class classification problems.

1 Introduction

In the last decade, neural computation has become increasingly popular, both in university research and the commercial marketplace. Neural network models are already used for solving problems in various areas such as pattern-recognition, signal processing, optimisation, control theory, time series analysis and the simulation of complex dynamical systems.

Neural networks are a type of computational structure inspired by observed processes in the natural networks of biological neurons in the brain. Each simple computational unit, a neuron, is highly interconnected with a weight. Neural network models are effective even in conjunction with complex parallel processing, learning patterns from examples, insensitivity to disturbing noise, real-time capability, self-organization and generalization (Taylor 1993, Liu 1997, Liu and Sin 1997).

The classical application of neural networks to weather forecasting was introduced by Widrow and Smith in 1963. They applied the Adaline to predict the occurrence of the next day's rainfall on the basis of fluctuations in the barometric pressure over the two days preceding their calculation. The percentage of

successful predictions was comparable to that achieved by the official weather prediction agency, which used a huge set of related parameters for weather forecasting (Karayiannis and Venetsanopoulos 1993). In this decade, some researchers (Chung and Kumar 1993) have shown with their experimental results that neural networks are effective in acquiring knowledge automatically from meteorological data.

The aim of this paper is to look at an alternative way of learning the behaviour of the weather system. We focus on the Naive Bayesian classification for predicting the occurrence of rain/no-rain and the different levels of rainfall over a 24-hour period during the rainy season in Hong Kong. As a supplement, we will compare the performance of the above method with the BPN for forecasting purposes.

2 Naive Bayesian Classifiers

Naive Bayesian classifiers, or Naive Bayes, are a simplified form of Bayes' rule that assumes the independence of observations. In spite of this assumption, some research results (Buntine 1989, Cestnik 1990, Langley 1992) have demonstrated that Naive Bayes have com-

petitive performance in comparison with other learning algorithms such as Instance-Based Learning and the Decision Tree Algorithm, with the normal distribution assumption or some form of discretisation for continuous-valued attributes.

2.1 Naive Bayes Criterion

Given a set of instances, each instance is specified with a set of attribute values and a class. These instances are used for supervised learning. In such a learning environment, each instance belongs to only one class. Now we try to look for a classification rule that fits the training set, and which can be used for classifying new instances into classes.

Let $A_i, i = 1, \dots, n$, be a set of attributes. Each attribute has a certain number of possible values $V_{A_i,j}, \{j = 1, \dots, m_j\}$, which denote the observed value of the attribute A_i in instance j . Let m_1 be the total number of training instances and m_2 the total number of testing instances, implying that $m_1 + m_2 = m$. Let $C_k, k = 1, \dots, p$ be a set of target classes. Our data-set contains m_1 training instances, m_2 testing instances, p classes and n attributes.

The Bayesian approach to classification estimates the posterior probability given the observed attribute values for the instance. The class with the highest estimated posterior probability is selected. The posterior probability is calculated using the following formula,

$$\begin{aligned}
 p(C_k|V_{\{A_i\}}) &= p(C_k|V_{\{A_1\}}V_{\{A_2\}}V_{\{A_3\}} \dots V_{\{A_n\}}) \\
 &= \frac{p(C_kV_{\{A_1\}}V_{\{A_2\}}V_{\{A_3\}} \dots)}{p(V_{\{A_1\}}V_{\{A_2\}}V_{\{A_3\}} \dots)} \\
 &= p(C_k) \frac{p(C_k|V_{\{A_1\}}) p(C_k|V_{\{A_2\}})}{p(C_k) p(C_k|V_{\{A_1\}})} \\
 &\quad \frac{p(C_k|V_{\{A_1\}}V_{\{A_2\}}V_{\{A_3\}}) \dots}{p(C_k|V_{\{A_1\}}V_{\{A_2\}})} \dots \tag{1}
 \end{aligned}$$

where $V_{\{A_i\}} = V_{A_i,1}V_{A_i,2}V_{A_i,3} \dots V_{A_i,m_1}$.

Recall that the independence of the observations is assumed; Equation 1 can be further simplified as

$$\begin{aligned}
 p(C_k|V_{\{A_i\}}) &= p(C_k) \frac{p(C_k|V_{\{A_1\}}) p(C_k|V_{\{A_2\}})}{p(C_k) p(C_k)} \\
 &\quad \frac{p(C_k|V_{\{A_3\}}) \dots}{p(C_k)} \\
 &= p(C_k) \prod_{i=1, j=1}^{n, m_1} \frac{p(C_k|V_{A_i,j})}{p(C_k)} \\
 &= p(C_k) \prod_{i=1, j=1}^{n, m_1} H_{A_i,j} \tag{2}
 \end{aligned}$$

Equation 2 shows that the posterior probability can be computed by multiplying the a priori probability of C_k by a set of factors $H_{A_i,j}, 1 \leq j \leq m_1, 1 \leq i \leq n$ to obtain the final posterior probability, with a form $H_{A_i,j} = p(C_k|V_{A_i,j})/p(C_k)$.

2.2 Estimation of Probabilities

Let $N(V_{A_i,j})$ be the count of the same value $V_{A_i,j}$ of attribute A_i which was being observed in the training

set, and similarly, let $N(C_kV_{A_i,j})$ be the count of $V_{A_i,j}$ of attribute A_i and class C_k being observed.

Consider the approximation of probabilities for Equation 2 with relative frequencies. We have

$$p(C_k|V_{A_i,j}) = \frac{N(C_kV_{A_i,j})}{N(V_{A_i,j})}$$

with properties:

$N(V_{A_i,j})$	$N(C_kV_{A_i,j})$	$H_{A_i,j}$
0	0	1
> 0	0	0
> 0	> 0	$N(C_kV_{A_i,j})/[N(V_{A_i,j}) \times p(C_k)]$

According to Equation 2, some problems arise with the above estimation. $H_{A_i,j}$ will be undefined, with ($N(V_{A_i,j}) = 0$) if it is not predefined to 1. The approximation seems unreliable when $N(V_{A_i,j})$ is very small. In 1965, Good was the first to introduce a more flexible and convenient class of initial probability density approximation, which involved two parameters a and b for estimating conditional probabilities. Let $a + b = \xi$, where ξ is related to the amount of noise in the domain. Once ξ is determined, we can assume that $a = p(C_k) \times \xi$ and $b = \xi - a$ (Cestnik 1990).

To estimate the probabilities of Equation 2 when applying the initial probability density function, we let

$$p(C_k|V_{A_i,j}) = \frac{N(C_kV_{A_i,j}) + a}{N(V_{A_i,j}) + \xi}$$

with properties:

$N(V_{A_i,j})$	$N(C_kV_{A_i,j})$	$H_{A_i,j}$
0	0	1
> 0	0	$\xi/[N(V_{A_i,j}) + \xi]$
> 0	> 0	$[N(C_kV_{A_i,j}) + a]/[(N(V_{A_i,j}) + \xi) \times p(C_k)]$

3 Data Collection

Meteorological data for our training and testing purposes was obtained from the Hong Kong Royal Observatory (now the HK Observatory) between 1984 and 1992. Daily observation were taken at Observatory headquarters and King's Park. We were interested in rainy season data in Hong Kong, so we only extracted data from May to October each year.

3.1 Data Preprocessing

The raw data set must be converted into a suitable format before training. This is not only a necessary step for processing a large low-quality data set, but also guarantees that reasonable analysis will be obtained from the experiments.

When a weather phenomena data item was found missing in the instances, we approximated it using a linear interpolation function which was constructed with the nearby values of the same element. The corresponding input attributes must adopt the same set

Input Col.	Description	Remarks
1-12	Month value of each records	12 codes are used; ie, [000010000000] = May, [000000100000] = July, etc.
13	Today mean of air temp.	measurement unit: °C
14	5-day mean: mean of air temp.	
15	Diff. between the temp. at 04:00 and 24:00	
16	Today mean of atmo. pressure	measurement unit: hPa
17	5-day mean: mean of pressure	
18	Diff. between the pressure at 04:00 and 24:00	
19-26	Prevailing wind direction (8 nodes for 8 directions)	The order of them are: N, NE, E, SE, S, SW, W, NW; ie. [10000000] = N and [00100000] = E, etc.
27	Prevailing wind speed	measurement unit: meters/sec
28	Today mean of cloud amount	measurement unit: okats
29	5-day mean: mean of cloud	
30	Diff. between the cloud amount at 04:00 and 24:00	
31	Total rainfall	measurement unit: mm
32	5-day mean: total rainfall	
33	Diff. between the rainfall at 04:00 and 24:00	
34	Today mean of humidity	measurement unit: %
35	5-day mean: mean of humidity	
36	Today total of sunshine	measurement unit: hour
37	5-day mean: total of sunshine	
38	Daily potential evaporation	measurement unit: mm
39	Daily mean of dew point temp.	measurement unit: °C

Table 1: Basic Input Parameters for Weather Prediction

of measurement units; as full metric units have been used by the Observatory since 1986, data from some elements in 1984 and 1985 had to be converted to the same measurement units. The input data were examined and transformed to normally distributed input in order to achieve a better performance.

3.2 Data Preparation

Different input factors would have different effects on the final predictions, so we have to select those significant input parameters for different prediction purposes. Table 1 illustrates 39 attributes which are basic input factors for our experiments. For the prediction of the occurrence of rainfall in one day (00:00 hr to 24:00 hr), days with less than 0.05mm rainfall were defined as 'no rain', and the rest were defined as 'rain' after the defuzzification process. Although the bulk of the information we assimilate every day fuzzy, most of the actions or decisions taken by people or machines are crisp or binary (Tanaka 1993, Türkşen 1997). In our experiments, Rain was the fuzzy data set for predicting the occurrence of either rain or no-rain, and Depth3 was the fuzzy data sets for the precipitation prediction with membership functions as follows:

$$\underline{\text{Rain}} = \{\text{no-rain, rain} | \forall \mu_{\underline{\text{Rain}}}(\text{no-rain}), \mu_{\underline{\text{Rain}}}(\text{rain}) \in [0, 1]\}$$

where

$$\mu_{\underline{\text{Rain}}}(\text{no-rain}) = \begin{cases} 1 & \text{if } 0 \leq x < 0.05 \\ \frac{(0.055-x)}{0.005} & \text{if } 0.005 \leq x \leq 0.055 \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_{\underline{\text{Rain}}}(\text{rain}) = \begin{cases} 1 & \text{if } x \geq 0.005 \\ 0 & \text{otherwise} \end{cases}$$

$$\underline{\text{Depth3}} = \{\text{Light, Moderate, Heavy} | \forall \mu_{\underline{\text{Depth3}}}(\text{Light}), \mu_{\underline{\text{Depth3}}}(\text{Moderate}), \mu_{\underline{\text{Depth3}}}(\text{Heavy}) \in [0, 1]\}$$

where

$$\mu_{\underline{\text{Depth3}}}(\text{Light}) = \begin{cases} 1 & \text{if } 0.05 < x < 2.6 \\ \frac{(2.855-x)}{0.255} & \text{if } 2.6 \leq x \leq 2.855 \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_{\underline{\text{Depth3}}}(\text{Moderate}) = \begin{cases} 1 & \text{if } 2.6 \leq x \leq 14 \\ \frac{(15.14-x)}{1.14} & \text{if } 14 < x \leq 15.14 \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_{\underline{\text{Depth3}}}(\text{Heavy}) = \begin{cases} 1 & \text{if } x > 14 \\ 0 & \text{otherwise} \end{cases}$$

It is clear that the range of each left boundary is 10% of its membership function for the above defined fuzzy sets. λ-cuts for fuzzy sets is one defuzzification method, and was considered in our study. In general, the set A_λ is a crisp set called λ-cut (or α-cut) set of the fuzzy set A, where A_λ = {x | μ_A(x) ≥ λ} and λ ∈ [0, 1]. The two rainfall categories (Rain, Depth3)

fuzzy sets apply the $\lambda = 1$ cut sets for further computation and discussion. Tables 2 and 3 show the two and three classes of rainfall categories for our next-day rainfall level predictions.

4 Class-Dependent Discretization

Class-dependent discretization refers to the discretization of our continuous data for the use of the Naive Bayes, which requires an input of discrete values. It is said that Naive Bayes which uses discretization during preprocessing performs better than Naive Bayes using the normal distribution assumption in some domains with continuous-valued attributes. We can apply Naive Bayes in domains with continuous-valued attributes when integrated with a discretization algorithm to transform continuous data into ordered discrete ones. Unlike most traditional discretization procedures, a class-dependent discretizer can automatically determine the preferred number and width of intervals of continuous data, and significantly improve the classification performance of many existing learning algorithms (John et al. 1995).

In our general assumption, we have a set of m_1 training instances. Each of these instances has been preclassified into one of p classes, where each class represents a label $C_k, k = 1, \dots, p$. The whole set of data is described by n distinct attributes, A_1, \dots, A_n . For any attribute A_i , there is a domain of attribute values defined as $V_{A_i,j}, j = 1, \dots, m_1$ (training set) and $V_{A_i,j'}, j' = 1, \dots, m_2$ (testing set). They can be numeric, symbolic, or both.

4.1 The Discretization Criterion

The discretization criterion, based on the concept of Class-Attribute dependence, seeks to maximize the dependency relationship between the target class and a continuous-valued attribute, and minimize the amount of information lost due to discretization. Let a boundary set of any attribute, A_i , be $B_i = \{e_0, e_1, \dots, e_{L_i}\}$ with a set of ordered endpoints, where e_0 denotes the lower boundary value and e_{L_i} denotes the upper boundary value of the observed attribute, A_i . The ordered boundary points give $e_{\beta-1} < e_\beta$ with $\beta = 1, 2, \dots, L_i$, where L_i represents the total number of intervals of the attribute. Let Q_i be a 2-dimensional Quanta Matrix of the observed attribute A_i , which is shown in Table 4. The matrix's element $q_{\alpha\beta}$ denotes the total number of observed attribute values which fall within the partition $[e_{\beta-1}, e_\beta]$ and belong to class C_α from the training set.

Based on the theory of maximum mutual information discretization, we partitioned the original m_1 possible continuous-valued data to L_i initial possible or-

Class	Boundary				Row Total	
	$[e_0, e_1]$	\dots	$[e_{\beta-1}, e_\beta]$	\dots		$[e_{L_i-1}, e_{L_i}]$
C_1	q_{11}	\dots	$q_{1\beta}$	\dots	q_{1L_i}	q_{1*}
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
C_α	$q_{\alpha 1}$	\dots	$q_{\alpha\beta}$	\dots	$q_{\alpha L_i}$	$q_{\alpha*}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
C_p	q_{p1}	\dots	$q_{p\beta}$	\dots	q_{pL_i}	q_{p*}
Col Total	q_{*1}	\dots	$q_{*\beta}$	\dots	q_{*L_i}	m_1

Table 4: The 2-Dimensional Discretization Quanta Matrix

dered discrete intervals. The rule of maximum allowable number of intervals, $L_i \leq m_1/(\lambda \times p)$, would be applied here. In this case, Wong and Chiu suggest the parameter λ should be 3 for liberal estimation, with m_1 training instances and p classes.

4.2 Boundary Improvement

Considering the 2D quanta matrix in Table 4, the estimated joint probability is,

$$p(C_k = C_\alpha, V_{A_i,j} \in [e_{\beta-1}, e_\beta]) = P_{\alpha\beta} = \frac{q_{\alpha\beta}}{m_1} \tag{3}$$

In a similar way, the estimated marginal probabilities can be set as,

$$p(C_k = C_\alpha) = P_{\alpha*} = \frac{q_{\alpha*}}{m_1} \tag{4}$$

$$p(V_{A_i,j} \in [e_{\beta-1}, e_\beta]) = P_{*\beta} = \frac{q_{*\beta}}{m_1} \tag{5}$$

The Class-Attribute Mutual Information (CAMI) between the classes C_k and the observed attribute interval boundaries with the Quanta Matrix Q_i is considered as,

$$MI = \sum_{C_k} \sum_{V_{A_i,<j>}} P_{\alpha\beta} \ln\left(\frac{P_{\alpha\beta}}{P_{\alpha*} \times P_{*\beta}}\right) \tag{6}$$

where $MI \geq 0$.

The boundary improvement process attempts to adjust the initial boundary set between the lower and the upper boundary pairs, which starts from the first ordered interval $[e_0, e_1]$. Within the boundary set, each interval can be perturbed either boundary up or boundary down in order to maximize the value of the CAMI. Both the boundary set and the relevant quanta matrix must be modified after each interval adjustment. In order to obtain an optimal estimation of global interdependence, the boundary improvement process will be repeated until no improvement is found.

4.3 Interval Reduction

Because of redundancy, elements of some adjacent intervals of the modified quanta matrix may have a very similar frequency distribution within the observed class. If two neighboring boundary pairs fail the statistical test in Equation 7, the frequency distribution between the two intervals and the observed class must be

Input Column	Description	Remarks
1-39	Basic Inputs from Table 1	
40-46	Class of rain/no-rain of 1-day before, 2-day before and so on, up to 1-week before	Class 1(no-rain): $x < 0.05$; Class 2(rain): $x \geq 0.05$; where x denotes daily total rainfall in mm.

Table 2: Input Parameters for Rain/No-rain Prediction

Input Column	Description	Remarks
1-39	Basic Inputs from Table 1	
40-46	Class of rainfall level of 1-day before, 2-day before and so on, up to 1-week before	(Light rain): $0.05 < x < 2.6$; (Moderate rain): $2.6 \leq x \leq 14$; (Quite heavy rain): $x > 14$; where x denotes daily total rainfall level.

Table 3: Input Parameters for Rainfall Depth Prediction

insignificantly interdependent (Wong and Chiu 1987), and the two intervals can be merged.

The Class-Attribute Interdependence Redundancy (CAIR) is $IR = MI/JE$, which is bounded by zero, where MI is calculated from Equation 6 and JE is called Joint Entropy, with the formula as follows:

$$JE = - \sum_{C_k} \sum_{V_{A_i,j}} P_{\alpha\beta} \ln(P_{\alpha\beta})$$

Once the "redundant" intervals are eliminated, the following statistical test is introduced:

$$IR \geq \frac{\chi_{(p-1)(L_i-1)}^2}{2m_1 \cdot JE} \tag{7}$$

where $\chi_{(p-1)(L_i-1)}^2$ represents a Chi-square distribution with degrees of freedom $(p - 1) \times (L_i - 1)$. In this case, p denotes the total number of output classes and L_i denotes the total number of current intervals of the observed attribute. To check the adjacent interval pair, we can only supply the partial CAMI and the partial joint entropy in Equation 7 according to the frequency elements and subtotal training instances of the two neighbouring columns. This interval reduction algorithm is performed until all pairs of adjacent intervals pass the statistical interdependence test, and achieve an optimal number and an appropriate width of intervals in the observed attribute.

5 Results Comparison and Analysis

BPN is one machine learning technique applied in the automation of knowledge acquisition. The detailed construction of such networks can be obtained from Widrow et al., 1994. For our experiment, we used a neural network toolkit called NeuralWorks Professional II/Plus¹, version 5, running it on a SUN SPARC

¹Neural network simulator by NeuralWare, Inc, Pittsburgh, PA 15276-9910, USA.

workstation. The algorithms for classifying the NBN were written in C, a general-purpose programming language, using the UNIX system.

5.1 Rain/No-rain Prediction

Referring to Table 2, a total of 46 input columns are used to predict the occurrence of rain/no-rain during the Hong Kong rainy season. Table 5 illustrates the classification rates (C-Rate) of daily rain which is less than 0.05mm (Class 1) and daily rain ≥ 0.05 mm (Class 2) for different types of learning algorithms. In the table, the format "1120_{560,560}" denotes the fact that out of 1120 instances, 560 belong to Class 1 (the first parameter) and 560 to Class 2 (the second parameter), and so on. Both NBN-RF (relative frequency approximation) and NBN-IPD (initial probability density approximation) rows of information were calculated by randomly selecting 2/3 instances from the whole set of data for training. The other two rows (NBN-RF*, NBN-IPD*) of information were computed by randomly selecting 2 out of 3 training instances in each output representative. Table 5 reveals that if a reasonable portion of the training and testing instances were collected from each output class, a better classification rate could be achieved. However, this rate is also affected by the quality of the original data set. Since the best architecture of each learning algorithm was recorded, the optimal performance were about 73% for Class 1 and 71% for Class 2 with the learning algorithm BPN.

5.2 Precipitation Prediction

Similarly, Table 3 shows the 46 input parameters used for rainfall depth prediction. The three output nodes in Table 6 represents light rain (Class 1), moderate rain (Class 2) and heavy rain (Class 3). Let d be the volume of rainfall in mm, and the above 3 classes re-

Learning Algorithms	Training Sets	Testing Sets	C-Rate of Class 1 (%)	C-Rate of Class 2 (%)	Remarks
BPN	1120 _{560,560}	540 _{270,270}	73.1	71.1	optimum choices: 30 hidden nodes and Momentums = 0.3
NBN-RF	1104 _{193,911}	552 _{367,185}	57.8	77.4	
NBN-IPD	1104 _{193,911}	552 _{367,185}	58.9	75.5	amount of noise = 13
NBN-RF*	1104 _{374,730}	552 _{186,366}	65.6	74.3	
NBN-IPD*	1104 _{374,730}	552 _{186,366}	66.7	75.1	amount of noise = 2

Table 5: Prediction of Rain/No-rain

Learning Algorithms	Training Sets	Testing Sets	C-Rate of Class 1 (%)	C-Rate of Class 2 (%)	C-Rate of Class 3 (%)
BPN	540 _{180,180,180}	270 _{90,90,90}	64.3	66.1	65
NBN-RF	528 _{173,178,177}	263 _{86,100,77}	100	84	84.4
NBN-IPD	528 _{173,178,177}	263 _{86,100,77}	96.5	98	96.1
NBN-RF*	528 _{173,185,170}	263 _{86,93,84}	100	90.3	95.2
NBN-IPD*	528 _{173,185,170}	263 _{86,93,84}	95.4	97.9	96.4

Table 6: Prediction of Rainfall Depth

fer to $0.05 < d < 2.6$, $2.6 \leq d \leq 14$ and $d > 14$ respectively. The results of the **BPN** algorithm were collected from 25 hidden nodes with momentums of 0.32. The best outcomes of **IPD** and **IPD*** were obtained by setting the amount of noise as 1. The optimal performance (**NBN-IPD***) for the prediction was approximated to 95% for Class 1, 98% for Class 2 and 96% for Class 3. For rainfall depth prediction, the results from both the **RF** and the **IPD** are much better than that of the **BPN**, achieving around 90% accuracy for the unseen testing data.

The Forecast Group at the Observatory suggested the ranges of the rainfall level (after the similar defuzzification process in subsection 3.2) for the prediction shown in Table 7. The training set, 1104_{373,204,240,174,113} could be constructed from our meteorological data (out of 1104 instances, 373 is for no rain, 204 for trace rain, 240 for light rain, 174 for moderate rain and 113 for heavy rain) and the testing set is 552_{186,102,120,87,57}. The optimal results of the **NBN-IPD*** were approximated to 73%, 14%, 60%, 93%, and 87%. Figure ?? shows the average classification rates with the amount of noise ranging from 1 to 15. In general, the average performance varies between 64.4% and 65.7%.

Average Performance of Precipitation Prediction by **NBN - IPD*** fig:depth

6 Conclusions and Discussions

The experimental results indicate that the performances of both the **BPN** and **NBN** are comparable, and that they gave a reasonable performance on the prediction of the occurrence of next-day rain/no-rain. The peak performance of rain and no-rain was

about 73% and 71%. The optimal performance was produced by **BPN** with enough and relevant information for training.

Although **NBN** performed extremely well on the prediction of rainfall depth within 3 categories (see Table 6 for details), the average performance on the prediction of rainfall range levels within 5 categories dropped to 65% with the same networks. Some possible reasons for such great differences can be concluded (Liu and Wong 1996) as follows:

- Rainfall is a stochastic event and the cause of its occurrence is very complex. Even under identical weather conditions, it is possible that it will rain at one moment, but not at another.
- The number of explanatory variables used as input parameters may not be sufficient to capture all of the necessary features for the 24-hour-period prediction, since significant changes in weather conditions may take place during that time.
- Before forecasting experts make a prediction of future weather element distribution over a particular area, they require extra information about the surrounding area. However, the data used in the experiment was extracted from a point station (the Observatory at King's Park) only. Data from other weather stations, such as Waglan Island and Hong Kong International Airport, were excluded from the study for reasons of simplicity and data unavailability.

Currently, most research papers in weather forecasting limit themselves to short-range effects. Apart from the study of other elements in weather forecasting (temperature changes, severe thunderstorms, etc),

Category	Nil	Trace	Light	Moderate	Heavy
Depth d (mm)	$0 \leq d < 0.05$	$0.05 \leq d < 0.1$	$0.1 \leq d < 4.9$	$4.9 \leq d < 25.0$	$d \geq 25.0$

Table 7: Rainfall Categories

we have to be concerned about long-term predictions. All in all, weather forecasting will remain a permanent scientific challenge for future research.

Acknowledgement

The authors appreciate the support from UGC Grant (B-Q092).

References

- [1] Buntine W. (1989) Learning Classification Rules using Bayes. *Proceedings of the 6th International Workshop on Machine Learning*, p. 94-96.
- [2] Cestnik B. (1990) Estimating Probabilities: A crucial task in machine learning. *Proceedings of the European Conference on AI*, p. 147-149.
- [3] Ching J.Y., Wong A.K.C. and Chan K.C.C. (1995) Class-Dependent Discretization for Inductive Learning from Continuous and Mixed-Mode Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 6, p. 1-11.
- [4] Chung C.Y.C. and Kumar V.R. (1993) Knowledge Acquisition using a Neural Network for a Weather Forecasting Knowledge-based System. *Neural Computing & Applications*, 1, p. 215-223.
- [5] Dai H.H. (1994) *PhD. Dissertation, Learning of Forecasting Rules from Large Noisy Real Meteorological Data*. Department of Computing, RMIT, Australia.
- [6] Dai H.H. and Ciesielski V. (1994a Nov) Learning of Inexact Rules by the FISH-NET Algorithm from Low Quality Data. *Proceedings of 7th AJCAI*, World Scientific, p. 108-115.
- [7] Dai H.H. and Ciesielski V. (1994b Nov) The Low Prediction Accuracy Problem in Learning. *Proceedings of 2nd Australian and New Zealand Conf. on IS, IEEE*, p. 367-371.
- [8] Good I.J. (1965) *The Estimation of Probabilities*. M.I.T. Press, Cambridge, Massachusetts.
- [9] Karayiannis N.B. and Venetsanopoulos A.N. (1993) *Artificial Neural Networks Learning Algorithms, Performance Evaluation and Applications*. Kluwer Academic Publishers, USA.
- [10] Langley P., Iba W. and Thompson K. (1992) An Analysis of Bayesian Classifiers. *Proceedings of the 11th National Conference on AI*, p. 223-228.
- [11] Liu J. and Wong L. (1996) A Case Study for Hong Kong Weather Forecasting. *Proceedings of the International Conference on Neural Information Processing*, Hong Kong, 2, p. 787-792.
- [12] Liu J.N.K. (1997) Quality Prediction for Concrete Manufacturing. *Automation in Construction*, Elsevier Science, 5, p. 491-499.
- [13] Liu J.N.K. and Sin K.Y. (1997) Fuzzy Neural Networks for Machine Maintenance in Mass Transit Railway System. *IEEE Transactions on Neural Networks*, 8, 4, p. 932-941.
- [14] Royal Observatory of Hong Kong (1984-1986) *Surface Observations in Hong Kong*. The Government Printer, Hong Kong.
- [15] Royal Observatory of Hong Kong (1987-1992) *Surface Observations in Hong Kong*. The Government Printer, Hong Kong.
- [16] Tanaka Y. (1993) An Overview of Fuzzy Logic. *Proceedings of WESCON, IEEE*, San Francisco, USA, p. 446-450.
- [17] Taylor J.G. (1993) *The Promise of Neural Networks*. Springer-Verlag, London, Great Britain.
- [18] Türkşen I.B. (1997) Fuzzy Logic: Review of Recent Concerns. *Proceedings of IEEE Int. Conf. on SMC*, Orlando, USA, 3, p. 2975-2978.
- [19] Widrow B. and Simth F.W. (1963) Pattern-Recognizing Control Systems. *Proceedings of Computer and Informations Sciences Symposium*, Spartan Books, Washington, USA, p. 462-467.
- [20] Widrow B., Rumelhart D. and Lehr M. (1994) Neural Networks: Applications in Industry, Business and Science. *Communications of the ACM*, 37, 3, p. 93-105.
- [21] Wong A.K.C. and Chiu D.K.Y. (1987) Synthesizing Statistical Knowledge from Incomplete Mixed-Mode Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9, 6, p. 796-805.
- [22] Zhang M. and Scofield R.A. (1992) Satellite-derived Estimation of Rainfall in Forward-backward Thunderstorm Propagation Model Using Neural Network Expert System Techniques. *Proceedings of IEEE/INNS Int. Joint Conf. on NN*, Baltimore, 2, p. 425-430.

Reverse Engineering and Abstraction of Legacy Systems

Margot Postema, and Heinz W. Schmidt
 School of Computer Science & Software Engineering
 Monash University
 900 Dandenong Road
 Caulfield East, VIC 3145, Australia
 Email: {margot, hws}@csse.monash.edu.au

Keywords: reverse engineering, legacy systems, transformation, abstraction

Edited by: Rudi Murn

Received: June 23, 1998

Revised: September 11, 1998

Accepted: October 11, 1998

Extremely large software systems which have been developed and maintained by many different people are termed legacy systems. These legacy systems were traditionally developed using methods such as structured analysis and design, or even individual programming techniques and styles. Over time, maintenance has changed the original program structure and specifications. However, usually the specifications have not been maintained, and the current design and program understanding is lost. Maintenance of these systems becomes so costly, that they become candidate for reengineering. Reverse engineering of legacy systems aims at discovering design and specification of existing software programs. The recovered designs are then forward engineered to improved systems. This difficult task can be assisted by CASE tools, but still requires human expertise and domain knowledge. With a technology shift from structured to object-oriented software construction, an additional problem arises i.e. transforming the structured legacy system to an object-oriented system. We outline the current state of CASE tools, followed by research directions. Further, we indicate that a two staged approach may be advantageous, where self-contained subsystems can be abstracted for design recovery and object discovery.

1 Introduction

Extremely large software systems which have been developed and maintained by many different people are termed legacy systems. Since the early 1970's, these legacy systems were developed using methods such as structured analysis and design (Yourdon, 1989). Prior to this, individual programming techniques and styles were used to develop software systems. Maintenance and understanding of legacy software can pose problems, particularly when the original programmers have left, and replacement staff do the modifications. Due to individual programming styles, programs developed prior to using structured programming techniques, could be even more difficult to understand and maintain. Over time, source code maintenance has changed the original software specification and design. However, as is often the case, the specifications and design have not been updated. Thus, program design and understanding is lost, and the only documentation of the system is the source code itself.

These software systems become difficult to maintain, and a decision needs to be made as to their future, i.e. are they to be discarded, maintained or reengineered? For a system to be reengineered, it must be firstly reverse engineered to discover system design

and specifications. From here, program understanding, and abstraction of higher level concepts occurs prior to forward engineering or reconstruction. Tools assist the reverse engineering process. Research is now also developing tools to assist program abstraction for reconstruction.

In the following Section, we discuss aspects of the decision process. System renovation (or reengineering) are in Section 3. Sections 4 and 5 define reverse engineering (a subset of reengineering), and give an overview of CASE tools and methods used to recover as well as abstract specification and design. Section 6 describes the current state of research and is followed by future research direction and a conclusion.

2 The Decision Process

Jacobson's decision matrix (Jacobson & Londström, 1991 p. 341) identifies systems which contain high business value and need to be reengineered. A similar model to this, is also presented in Sneed (1995a). Sneed further discusses quantifying and proving benefits of reengineering.

As the reengineering process can become expensive, analysis is necessary to determine the business value

embedded in the software itself. Hence, methods and techniques are essential for success. When analysing and assessing software systems, it may be beneficial to isolate only portions (or sub-systems) of the software which are heavily maintained, and contribute to the majority of the maintenance cost. These sub-systems could be interfaced to the rest of the legacy software and then reimplemented. Thus, we have extended the decision matrix for the reengineering decision (mentioned previously), to a 3-dimensional (3 – d) view (see Figure 1). This view additionally identifies reuse value. In Figure 1, the 3 – d decision matrix has an x -axis representing business value, a y -axis representing changeability and a z -axis representing encapsulation. We assume a scale of 0 to 1 for each axis, where 0 indicates no value and 1 indicates a maximum value. Our previous discussion on isolating sub-systems for reimplementing is highlighted in Figure 1 as systems candidate for *partial reengineering* in a society of objects.

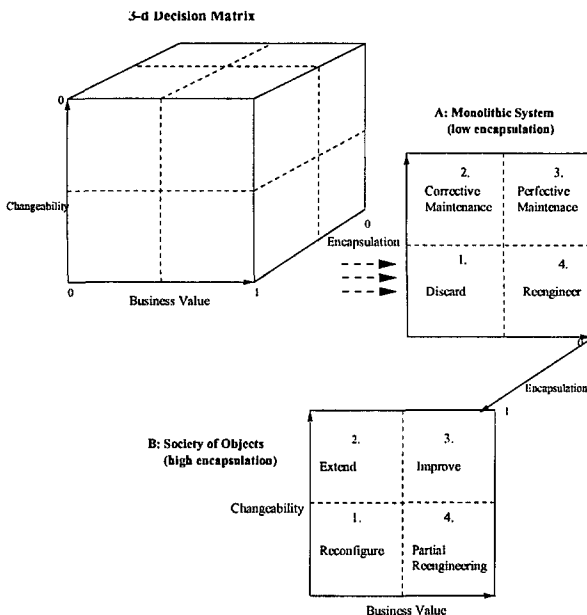


Figure 1: 3-dimensional Decision Matrix

The 3 – d decision matrix can then be viewed as a **Monolithic System** (low encapsulation) and a **Society of Objects** (high encapsulation). It is represented as follows:

– **A: Monolithic System** (low encapsulation).

1. Low business value and low changeability indicates **discard** the system.
2. Low business value and high changeability indicates **corrective maintenance** of the system.
3. High business value and high changeability indicates **perfective maintenance** of the system.

4. High business value and low changeability indicates **reengineer** the system.

– **B: Society of Objects** (high encapsulation).

1. Low business value and low changeability indicates **reconfigure** the system.
2. Low business value and high changeability indicates **extend** the system.
3. High business value and high changeability indicates **improve** the system.
4. High business value and low changeability indicates **partial reengineering** of the system i.e. isolate portions of the system which are suitable for reengineering, and interface these to the existing system.

The outcome of analysing legacy software's value will often result in a decision to reengineer or renovate the software. We now describe the process involved.

3 System Renovation

For system renovation or reengineering to occur, methods are required to recover the current design and reengineer the software system. These methods assisted by tools include:

- reverse engineering
- redocumentation
- design recovery
- restructuring
- transformation
- forward engineering

Reengineering initially reverse engineers the current program to recover the high level abstraction or design of the software. The recovered design is then forward engineered, with a low-level implementation of the high level abstraction. The result is a new program with either the same functionality, or enhanced functionality to meet new requirements.

The reengineering process can be summarized by the diagram in Figure 2, where tools are used to decompose software for representation in various forms. Software documents are input and parsed, prior to storage in repositories. From here, extracted information is passed to browsers, where viewing of abstract syntax trees, graphs, call graphs, interfaces, documentation, reports, etc. occurs. The maintenance programmer now obtains understanding of the software system. Specifications can be abstracted, before proceeding with the forward engineering step.

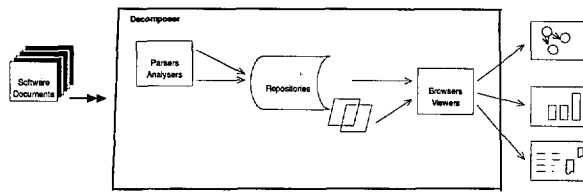


Figure 2: Reengineering Process

Experience and knowledge play an important role in the comprehension of a program (van Zuylen, 1993a). van Zuylen further describes how software engineers study program code. Two processes have been identified: top-down and bottom-up. The bottom-up approach looks at building higher level concepts by grouping together statements which perform a common function. The top-down approach, as is demonstrated in Figure 3, uses heuristics, refinement and adjustment until the software engineer has an understanding of the program. A combination of the two approaches is required for understanding and abstraction. Whilst the abstraction process is currently mostly manual, tools and research developments for semi-automated assistants are emerging. In Sections 5 and 6, we describe how partial automation of this process has been achieved with the use of tools.

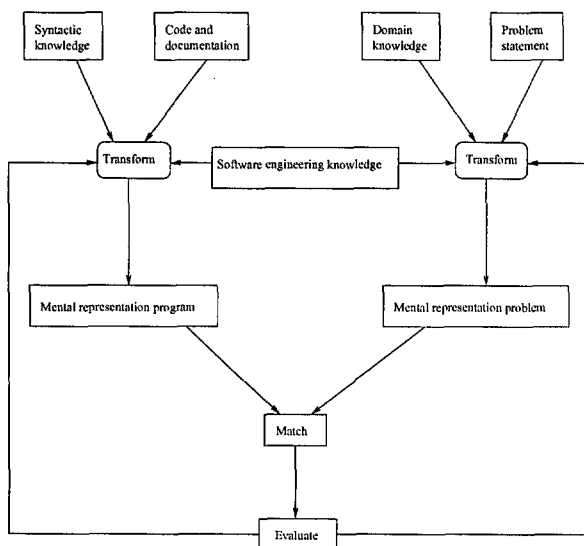


Figure 3: Inference structure for program comprehension, from van Zuylen 1993a, p88

Technology has now moved from structured to object-oriented (Meyer, 1988) techniques with many added advantages, particularly in the area of reduced maintenance costs. Thus, the current focus is to re-

verse engineer legacy systems, and convert these to object-oriented applications. At this stage, we must point out, that translating programs from one language and/or technology to another doesn't necessarily improve the design. Thus the current trend doesn't address the issue of redesigning programs, for improved understandability and maintainability.

Legacy codes are increasingly being made accessible as services in a network. This requires the isolation of self-contained legacy components, and their wrapping for interoperability. Such wrappers do not require an understanding or modeling of the entire set of functions originally required to be implemented when the system development was commissioned. Only those object and functions currently used and required from outside need to be modeled, i.e. another level of abstraction simplifies the reverse modeling or design recovery. A two staged approach seems more promising than total redesign from scratch, where:

1. In *Stage 1*, existing tested functionality is encapsulated in a first stage, and made accessible in a network with minimal changes to the original code
2. In *Stage 2*, portions of encapsulated code are successively replaced in one or more subsequent stages.

In *Stage 2* above, the discovery of semantic objects or reconstruction of an OO design is necessary to enable modern technique application. Identification of subsystems candidate for *partial reengineering* (as mentioned in Section 2) would then be possible.

The approach mentioned above is similar to that of Brodie & Stonebraker (1995), where legacy systems are incrementally migrated, and would be greatly aided by partial automation. Brodie & Stonebraker (1995, page 78) mention that incremental rewrites of legacy systems (i.e. migration) are much more successful than restructuring. The difficulty lies in identifying the candidate modules and being assured of an explicit interface or wrapper, i.e. any changes made in the wrapped module will not affect the implementation. Hybrid tools are being introduced to assist this transformation.

Before abstraction and reconstruction can commence, reverse engineering with the aid of tools can be performed.

4 Reverse Engineering

Reverse engineering may be viewed as the process of discovering software design and specification from source code, and representing it in forms such as data flow diagrams, flowcharts, specifications, hierarchy charts, call graphs etc. For an annotated bibliography in reverse engineering, the reader is referred to

Acronyms	
CASE	Computer Aided Software Engineering
OO	Object Oriented
CIA	C Information Abstraction
AST	Abstract Syntax Tree
FOL	First Order Logic
DESIRE	Design Recovery
ILP	Inductive Logic Programming
SQL	Structured Query Language
CBMS	Code-base management system
TPI	Transform Plug-In
SDK	Software Developer Kit
CQML	Code Query & Manipulation Language
CORBA	Common Object Request Broker Arch.
RCS	Revision Control System

Table 1: Acronyms

van den Brand (1997). *Ideally*, the reverse engineering process should be able to discover design abstractions and objects from the source code. Research is focusing its attention on this problem, which we elaborate in Section 6.

Currently, case tools for reverse engineering generate call graphs from legacy code (e.g. GDPro (Advanced Software Technologies, 1997)). Murphy et al. (1995) present an empirical study of five call graph extractors (cflow, CIA, Field, mkfunctmap and rigi-parse), which shows that the call graphs produced, vary according to treatment of inputs such as macros, function pointers, input formats etc. This can inherently affect the view that the software engineer has of the source code. The call graphs are produced by parsing the source code to produce abstract syntax trees (ASTs).

Example tools which use parsers to store information about the program include the C Information Abstraction System (CIA) (Chen, 1995a) and REDO (van Zuhlen, 1993b). The tools analyse and perform transformations on the parse tree and then generate information such as call graphs or data dependency analysis from the parse trees. CIA (Chen, 1995a) for example, constructs an *entity relationship* model and stores this in a database. Query and analysis tools can be used to visualise the information. REDO uses an intermediate language or representation, called *Uniform* (Breuer & Lano, 1991) to represent the source code and produce low level specifications. The REDO tools aim at reverse engineering from the source code in a bottom-up way. Additionally to REDO, standard CASE environments can be used to build views top-down (van Zuhlen, 1993b, p. 106)

However, large systems are often difficult to comprehend and need to be broken down. Other approaches to reverse engineering for program understanding and

hence abstraction assistants are presented below.

5 Program Abstraction

5.1 Program Slicing

Program slicing (introduced by Weiser (1984)) is a method for automatically decomposing programs by analysing their data flow and control flow. Slicing, or code isolation is a method for extracting components of a program that are concerned with a specific behaviour or interest such as a business rule or transaction. It is very useful in reverse engineering, but requires knowledge about data flows, dependencies and “dead code” (Rochester & Douglas, 1993).

“Many traditional techniques employed in reverse engineering (e.g. structure charts and call graphs) are inadequate for the recognition of algorithms, and certainly for the recognition of design” (Beck & Eichmann, 1993). Beck and Eichmann (1993) introduce a new form of program slicing (*interface slicing*) which aids in algorithm and design recognition in the reverse engineering process.

Traditionally, program slicing is defined as building a dependence graph using data- and control-flow analysis, and then generating actual slices of graph operations (Beck & Eichmann, 1993). An approach to identifying slices can be *specification driven* (Cimitile, Lucia & Munro, 1995). Interface slicing (Beck & Eichmann, 1993) reduces the procedures appearing in the interface (i.e. procedures used inside other procedures are excluded). The method reduces the view of procedures, thereby reducing graph size. Hence, abstraction is introduced to aid understanding. Other approaches to aid program understanding include approximate reasoning systems (see Section 5.3.)

5.2 Domain Specific Approaches

Research has also focused on using knowledge-based or expert systems approaches. The main principle behind this approach is to gather a library of previously discovered design patterns and apply these to the program in question. These systems are used in software and reverse engineering and are also features of analogical reasoning and relational matching.

The ASPIS (Aslett, 1991) project, uses knowledge about development methods and in particular, knowledge about application domains (i.e. domain knowledge) in the form of expert systems. This type of approach shows a great potential for improving software development of new systems. If we take a *banking domain* as an example, we could develop a new banking system by simply using an existing one in the knowledge base.

The ASPIS system can be used to construct a specification from reusable components. It’s retrieval mech-

anism makes use of a component called the Refinement History (Freeman, 1987), consisting of a log of previous results. It searches a Reuse Database; has a comprehensive thesaurus to match synonyms, and uses the notion of conceptual closeness based on the number of common microfeatures (analogical reasoning). Hence, components from different domains can also be identified. It is important to note, that a human administrator decides whether a component is suitable for inclusion in the reuse library.

The knowledge acquisition sessions for ASPIS to obtain the domain knowledge has been developed through the "interview method". However, this approach brings up the old *knowledge acquisition bottleneck* problem (Feigenbaum, 1977).

Examples using knowledge bases in reverse engineering use languages which support first order logic (FOL) (Bennett, 1993). These include the powerful pattern matching abilities of Prolog (Aslett, 1991) and Refine (Burson, Kotik & Markosoan, 1993). Generally, the idea behind these approaches can be summarised in the view presented in Figure 4 where:

1. Programs are compared to information available in the knowledge base.
2. Various views can be obtained by a generalisation process.
3. The user interacts with the knowledge base.
4. The user finds a solution.

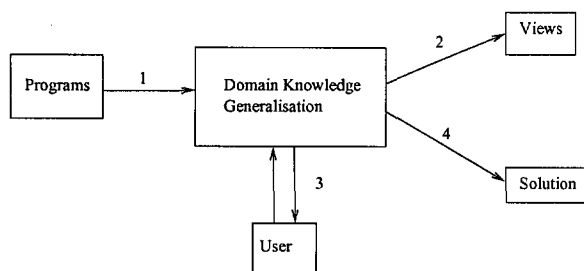


Figure 4: Using a knowledge-based approach to understand programs

Bennett (1993) describes the ReForm project which transforms code to FOL - it theoretically should be a general tool to read in any code (not language specific), and then output in any code. The system currently reads in assembler code and produces FOL, with the aim of producing specifications that are semantically correct with the original code. The knowledge base system should then transform the original code into a logically equivalent piece. It is an interactive system which requires input from the maintenance programmer. These inputs include domain specific knowledge about the representation of the specification. The

maintenance programmer can make changes to the semantics via an editor. Other systems such as REDO and MACS (Bennett, 1993) are very similar.

More example approaches using knowledge bases include analogical reasoning (Tausend & Bell, 1992) and relational matching (Vosselman, 1992). Exact relational matching was first used in chemistry and mathematics (Vosselman, 1992), growing out of a need for classification of molecules (consisting of atoms and connections). New molecules were compared to examples in the database. The search can start with a comparison of structural similarities, or expand to inexact matching using weighted primitive and relation tuples, and attribute valued thresholds. A primitive p is considered to inexactly match primitive q if all corresponding attribute value differences were less than the threshold. Analogical reasoning (or inexact matching) finds a solution for a new problem by using known solutions of a similar problem. The process makes use of a knowledge base and finds examples with similar structures and arguments. The relational match method uses the information theory to compare two relational descriptions. It finds the best mapping from features in one description to another, and requires an evaluation function to measure similarities. The relations are represented as a tuples or lists which may have attributes.

Whilst the above approaches do aid the reverse engineer, they assume comprehensive knowledge/databases which are domain specific and timely to build. Surprisingly, this approach is being followed in the current trend to convert procedural systems to object-oriented systems (Section 6.1).

5.3 Concept Understanding

Program analysis is usually the most time consuming and difficult part of software maintenance (Bennett, 1993). Understanding of a program and its concepts is often very different from just looking at the program specification or its formal definition (Biggerstaff, 1993a). Parsing technology, which is so commonly used in reverse engineering, assumes complete formal structures of patterns (Biggerstaff, 1993a). Furthermore, Biggerstaff (1993a) points out that as patterns are also open to ambiguity, the concept of understanding a program cannot be determined by algorithms and abstractions, but by "chaotic piecing together of evidence for a concept until some threshold of confidence is reached about its identity" (Biggerstaff, 1993a).

Currently there is much interest in the notion of reuse and design patterns. Numerous references are made to the design patterns identified by Gamma et.al. (1995) also known as the *gof*¹. However, it is still another new idea and one wanders whether the design

¹"gang of four"; i.e. the four authors of the OO design patterns text (Gamma et.al., 1995)

patterns identified by a particular group such as the *gof* are really reusable (Menzies, 1996). As Menzies (1996) points out, one needs to consider facts such as:

- Different developers use different designs to solve the same problem.
- At least one library of reusable design patterns contains significant deviations to the *gof* library.

Biggerstaff (1993a) summarises the various approaches as:

- Highly domain specific, knowledge bases - not suitable for large programs.
- Algorithmic program understanders or recognizers (such as a graph-parser), which are also computationally infeasible for complex systems.
- Model driven, plausible reasoning systems (e.g. DESIRE) - where results are approximate and imprecise.

The use of general, as well as domain specific knowledge, is featured in the Design Recovery system DESIRE (Biggerstaff, 1993a), *to assist reverse engineers understand programs*. It includes query facilities, slicing, domain knowledge, and call graphs, and relies heavily on user interaction. This seems to be the current approach to reverse engineering (particularly, with an emphasis of program understanding and abstraction), and is further expanded in Section 6.

Recovered program design is usually very poor, and requires restructuring for improved understandability and maintainability. Total automation of this process is very difficult, however some techniques and tools to assist this are described in the following Section. As mentioned in Section 3, a two staged approach to reverse engineering would be beneficial, where portions of the system are interfaced to the legacy code, and then restructured or redesigned. Further discussions are included in Section 7.1.

5.4 Restructuring

Restructuring involves transforming to improve code and/or design, eg. changing *If* statements to *Case* statements. A decision needs to be made when and where to restructure, requiring the use of metrics, and an evaluation function. Designers need to select good abstract roles for the modules, using abstraction and information hiding principles. They must be able to predict future likely changes - extendibility. The software engineer needs to make design decisions to incorporate information hiding principles, and which information needs to be shared. Then the system must adapt to new business requirements and/or enhancements. This is when the maintenance engineer discovers that the original design could be better "Therefore, modularisation as a reverse-engineering process

must be treated heuristically, rather than by a formal set of rules." (Schwanke, 1993, p.267). Maughan and Avotins (1995) investigate the production of an object-oriented reengineering tool-set for performing architectural restructuring of object-oriented systems. The system produces formal specifications from the source code, and records the reverse engineer's restructuring plans, indicating which axioms fail with the decisions made.

New design implementation can be further enhanced by including a metrics function. Metrics can be used to measure the quality of the object-oriented design. Martin (1995) describes a simple set of metrics that can be used in terms of interdependence between sub-systems of the design. The metric measures stability (based on class inter and intra-dependence), abstractness (number of abstract versus total classes) and distance from main sequence (using the abstract and stability measures) to determine where restructuring should occur.

Chu (1993) summarises previous approaches to program translation as:

- Traditional techniques translate programs on an element by element basis and then refine the programs. This results in semantically equivalent programs, but no improvement for maintainability or readability.
- Translation via abstraction and reimplementing of programs was proposed to address these issues. This is achieved by recognition of standard computational components called *cliques*, which are stored in a library. This method unrealistically assumes programs are built from standard components, and further requires exhaustive searching to recover designs.

Chu (1993) proposes that the re-engineer should restructure the parsed software and then apply rules to semantic patterns. He also follows a knowledge based approach, which seems quite timely to build, and is domain specific.

Methods and tools currently used in reengineering are *assistants* to the engineers. Specific reverse engineering tools have been useful, however abstraction techniques and concept understanding is still difficult. Hence we investigated if a machine learning technique such as inductive logic programming could be applied.

5.5 Inductive Logic Programming

Inductive logic programming (ILP) is a research area formed at the intersection of Machine Learning and Logic Programming. ILP can be used in the area of generalisation and classification of relations. ILP systems use background knowledge (*B*) and training examples (*E*) to induce hypotheses (*H*) expressed in a

logic program form (Lavrac & Dzeroski, 1994). These have the relationship:

$$B \wedge H \models E$$

An example from Lavrac (Lavrac & Dzeroski, 1994) is given as follows:

The task is to define the target relation *daughter*(X,Y) which states that X is a daughter of person Y, in terms of the background relations *female* and *parent*. Positive and negative examples are presented.

The following definition is formulated in the form of a Horn clause:

$$daughter(X, Y) \leftarrow female(X), parent(Y, X)$$

ILP systems can learn a single concept or multiple concepts. They can work as batch or incremental learners. ILP could be used to generalise software programs and classify a new example based on the induced hypothesis. Most ILP systems have been used for prediction. However Grendel2 (Cohen, 1994b) has been used for discovery.

GRENDDEL (Cohen, 1994a) is an extension of the algorithm used in Quinlan's FOIL (Quinlan, 1990). GRENDDEL is guided by an *antecedent description grammar* i.e. it adds an explicit grammatical bias to FOIL. The concept description language is input to the learner in the form of background knowledge. An example where GRENDDEL learns illegal chess positions from background knowledge, positive and negative examples is presented in Cohen (1994a).

Grendel2 (Cohen, 1994b), a successor to GRENDDEL has been extended to discover specifications from C code. A case study (Cohen, 1994b) of a large C program (over one million lines of source code) shows the ILP software can discover two thirds of the specification with about 60 percent accuracy. The system recovered declarative view specifications from relational database examples. Positive examples were obtained by executing the code to give materialised views of the database and the table relations were encoded as background knowledge. No negative examples are required for the learning system. The method focuses on using *inductive* reasoning about the behaviour of the code, rather than deductive reasoning of the static code.

ILP systems use examples and background knowledge to form a hypothesis. In the area of reverse engineering, this is only useful when many runs of a program (such as relational database views, representing dynamic behaviour) are used as input examples to the learner, and the business rules (eg. table relations) are used as background knowledge. ILP could be useful in a situation where there were many similar programs which could be generalised. However, this doesn't seem to be the case in legacy systems. Data representation would also need to be carefully considered.

In the following Section, we describe a reverse engineering environment, which is widely used in the development of reverse engineering and transformation applications.

5.6 Reasoning - Transformation Software

Reasoning, Inc. (Reasoning, 1997) is renowned for its products, formerly known as Software Refinery and the Refine Language Tools. These have been used in many reverse engineering research projects (Antoniol et.al. 1995, Newcomb & Kotik 1995, Wells et.al. 1995, Yeh et.al. 1995). In late 1996, Reasoning rearchitected and upgraded its products which are now called Reasoning5 code-base management system (CBMS), Reasoning Language Gateways, Reasoning Transform Plug-Ins (TPIs) and the Software Developer Kit (SDK). The technology represents a powerful solution for the Year 2000 problem (Millennium). The Millennium concerns the practice of storing calendar year dates in two-digit format instead of four-digits. Thus in 1999, the year is stored as the digit 99 and in the year 2000, this would be represented as 00. Major problems to this practice are foreseen if not corrected now. Reasoning's Transform Plug-Ins have been developed to automatically transform non Year 2000 software to Year 2000 compliant software.

The architecture of Reasoning Transformation Software is shown in Figure 5.

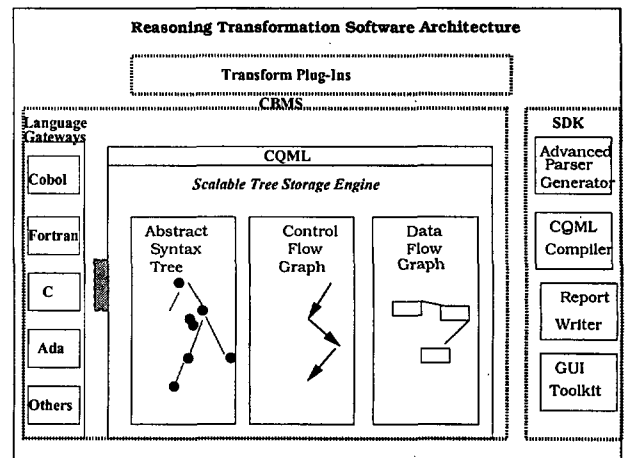


Figure 5: Reasoning Transformation Software Architecture

Reasoning5 uses a technology similar to that of database management systems i.e. data can be stored and queried. It is a code-based management system (CBMS) which stores information about source code. In Reasoning5, Code Query and Manipulation Language (CQML) can be used for transforming the source code. A language gateway is used to import source code to the CBMS, where it is represented as

abstract syntax trees, control flow graphs and data flow graphs. Once source code is in the CBMS, it can be transformed by Transformation Plug-Ins (TPIs). Reasoning provides language gateways for Ada, C, Cobol and Fortran. By using the Software Developer Kit, language gateways and TPIs can be customised to handle various dialects, or other languages. Reasoning5 runs on powerful, scalable Unix hardware and has no limit to program size.

6 State of The Art

We have presented tools and techniques useful for the reverse engineering process, but full automation of the whole reengineering process is not usually successful. Hence the focus is on using semi-automated and hybrid approaches.

Legacy programs can be parsed and stored as databases which can be then queried, similar to SQL queries on a relational database (Chen 1990, Jarzabek & Keam 1995). Rules (Gall & Klösch 1995, Jarzabek & Keam 1995) are used to discover an initial E-R model. Gall & Klösch (1995) for example, integrate domain knowledge into the object identification process. Jarzabek & Keam (1995) have implemented a knowledge based assistant which allows queries on the program database to assist the reverse engineer.

Graphical views of software systems are also an important issue. For instance, call graphs can become too large and complex for comprehension and understanding. Work is emerging on the visualisation aspects of large software systems. Ciao (Chen, 1995b), for example is a graphical navigator which allows queries on a program database, and also incorporates change management systems. Ciao highlights differences in entities between different versions of a program, which is essential for locating any files modified during a particular instance of modification. "The program difference technology is a critical first step in building a *software history* database that helps us study *aging* of legacy code." (Chen, 1995b). Another software visualisation tool is SHriMP (Simple Hierarchical Multi-Perspective) (Storey & Müller, 1995). The SHriMP visualization technique has been incorporated into the Rigi reverse engineering system and creates fisheye views. Program slicing (Section 5.1) could also be used to view subsections of programs.

6.1 Design Recovery and Abstraction

The notion of design recovery (Biggerstaff 1993b, Chikofsky 1993, Lano & Breuer 1993) includes using domain knowledge, external information, and deduction or fuzzy reasoning in the software system, to identify high levels of abstraction (Chikofsky, 1993).

The automation of the design recovery process doesn't seem to be currently feasible, as there is too much expert and domain knowledge required. Examples of this include various programming styles, different designs and implementation of the same problem, depending on the software engineer, and the various syntactical versions in the many programming languages and environments. Specifications which have been successfully extracted were mostly a manual effort (Brodie & Stonebraker, 1995, p. 168). Ideally, a learning system guided by the reverse engineer is required. Further investigations in areas of artificial intelligence and machine learning could be promising. Michie and Sammut (1995) for example are researching a machine learning model of human cognition.

Higher levels of abstraction are required for program understanding and comprehension. This can be achieved by using methods such as interface slicing (Beck & Eichmann, 1993) (Section 5.1). Once understanding is achieved, new technology (object-oriented) can be implemented.

6.2 Object Discovery

The current aims are to provide semi-automatic tools or cooperative environments to discover objects. These environments usually rely on some form of knowledge base or encoded heuristics.

Lano and Breuer (1993) describe reverse engineering structured COBOL to a formal object specification language (Z++). From there forward engineering can occur in any language. The COBOL code is firstly translated to UNIFORM intermediate code. Some transformation occurs (eg. PERFORM UNTIL → DO UNTIL); files and subprograms are classified as objects. Higher level object abstractions are found from dataflow analysis, where variables associated with main data structures are discovered.

DECODE (Quilici & Chin, 1995), another reverse engineering environment, comprises of:

1. An automated plan recogniser (APU) which examines source code for instances of patterns i.e. compares the source code to program plans stored in a plan library.
2. A knowledge base for recording extracted design information.
3. A structured notebook for editing and querying the extracted design.

The system is limited in that it only works well where the patterns encoded in the plan library match the legacy code. It provides a graphical view of the system, allowing easier programmer understanding and editing facilities.

7 Future Direction

As experience, knowledge, external information, deduction and refinement are required to recover design, we need a model to represent the cognition process i.e. a *recovery system*. The reverse engineer needs to guide the *recovery system* (Sneed, 1995b), which should also record the process, and allow for comments to be added. This is similar to the refinement history (Freeman, 1987) and reengineering plan (Maughan & Avotins, 1995). The process is analogous to the software development and maintenance processes, where the software engineer, armed with knowledge and aided by tools, decides how things are done, i.e. we need to *reverse* this process.

7.1 ToRevise

In our project "Towards Reverse Engineering by Visual Interactive Semantics" (ToRevise), we focus on modeling interfaces (wrappers for near self-contained legacy components).

Legacy codes are increasingly being made accessible as services in client/server networks. Hence a model is required to *assist* in isolating and abstracting self-contained legacy components, and their wrapping for interoperability. We assume for simplicity, that the underlying interoperability substrate is based on the CORBA (Mowbray, 1995) object model, and that only a fraction of the functionality of legacy components is actually used in the enterprise and therefore required interface specification recovery. The recovery and re-design can be seen as a two stage process where:

1. In *Stage 1*, modules are selected and encapsulated; and subsequently tested for original functionality.
2. *Stage 2* involves the reimplementing of the encapsulated modules.

For ToRevise we are mainly focusing on *Stage 1* support. The initial problem which needs addressing, is to identify candidate components of the system to reimplement, using object-oriented techniques. A reverse engineering *decision support tool*, which analyses source code information from a different perspective (eg. configuration management data), should indicate which modules are candidate. With a focus on the decision process, the approach should help to identify and rank the most promising areas of reverse engineering in legacy code. The second problem is then being able to guarantee that a module can be isolated and self-contained.

In industry, most large software development projects use configuration management tools to control access to source files and provide version control. The configuration management library, such as RCS (a tool that comes with most Unix systems) stores

Module	Size	Age	Reason
	KLOC	(Weeks)	1:Failure 2:Change
A	1000	2	1
B	1000	20	2
C	500	50	2
D	500	10	1
E	300	10	2
A	1000	4	1
A	1000	8	1
C	500	60	1
B	1000	50	1

Table 2: Example Configuration Management Data

the current version of the program and the difference (delta) from the previous version. RCS requires the software engineer check out each file for editing and check it in again when changes are complete. A version number is assigned to each checked in file. As such, the configuration management library is a starting point for providing information to identify the most promising modules for reverse engineering. Modules which should be *interfaced only* to the new implementations should also be highlighted. Once the candidates are selected, the design abstractions need to be discovered. Table 2 represents some example configuration management data, and can be represented by the view in Figure 6. Module *A* is highlighted as candidate for reverse engineering.

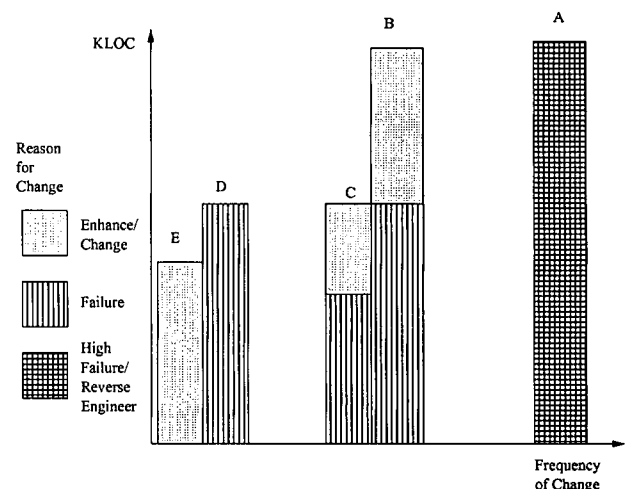


Figure 6: View of KLOC vs Frequency of change showing reason

Further analysis of configuration management data can also indicate which modules are related, i.e. modified at the same time, and for the same reason as another module. For Example, if we examine Figure 6, we might find that module *D*, and portions of modules

B and *C* were modified at the same time as module *A* in Figure 6. This may isolate self-contained sub-systems, and can be represented by the view in Figure 7, where:

- The current legacy system requires reengineering.
- Candidate sub-systems are identified.
 1. The sub-systems are isolated and tested for interoperability.
 2. Portions of the sub-systems are replaced.

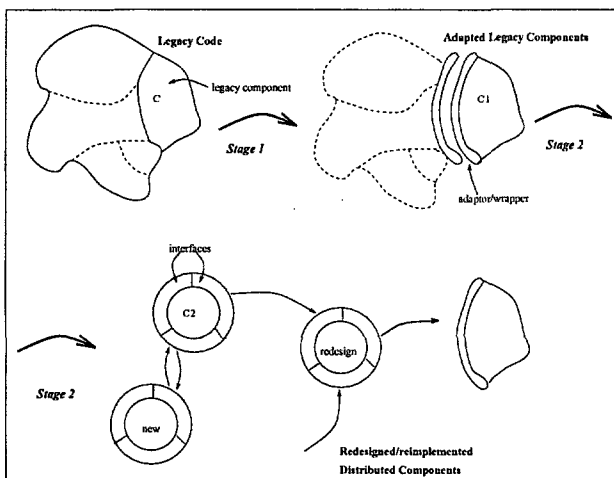


Figure 7: A Staged Approach using Abstraction for Design Recovery

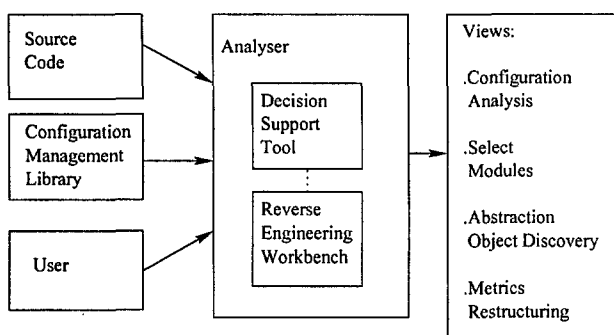


Figure 8: Design Abstraction and Object Discovery Tool

Figure 8 describes how the tool or *recovery assistant* should function. Inputs are source code, configuration management data and user knowledge. Various views are presented and the user interacts with the assistant to select modules for reimplementation.

At this stage, it must be emphasised that this may be a rather idealistic view of a solution to real world problems. Interleaving (Rugaber, 1995), or code fragmentation can make it difficult to achieve self-contained components. Interleaving can occur due to

efficiency reasons or inadequate software maintenance. The emphasis is still on an intelligent assistant, which needs to be aware of ever so many problems and programming idioms.

7.2 Related Work

As mentioned above, efficiency reasons or maintenance (patches and quick fixes) could lead to code fragmentation or routines which achieve many independent goals. With a focus on systems where programmers haven't aggregated related data structures and methods, techniques are required to abstract objects from source code.

Rugaber et.al. (1996) have developed a tool to detect and extract independent plans from a routine, leaving a smaller, more coherent routine. Plans are defined as descriptions of computational structures, proposed by designers to solve a problem. Domain specific knowledge seems necessary to develop these types of tools to extract novel and idiosyncratic plans. For example, Rugaber et.al. have used knowledge about usage patterns in order to discover exception handlers in program source code. Software understanding process involves two parallel knowledge acquisition techniques (Rugaber et.al., 1996):

- using domain knowledge to understand the code;
- using knowledge of the code to understand the domain.

Through empirical evaluation of source code, Rugaber et.al. have also detected co-occurrences of routines (routines always called by the same routines, executed under the same conditions, and computed data flows from one to another). They suggest that routines which co-occur are candidate for *reformulation wrappers*, and are building tools to perform this automatically.

Other work includes used a graph-oriented, direct-manipulation approach, (called the *star diagram*) to assist restructuring source code (Bowdidge & Griswold, 1994). The tool allows the user to manipulate source code and discover abstract data types. An approach similar to that mentioned in Section 5.2, has been adopted by Harris & Yeh (1996), who apply source code recognition techniques to supporting libraries.

As mentioned in Section 5.6, Software Refinery is the prevalent development environment for the latest reverse engineering tools. Adaptable automated tools such as these are required to assist the reverse engineering problem. Case studies and available data sets would significantly contribute to reverse engineering research (Wills & Cross, 1996). Analysis techniques now include references other than source code (eg. reference manuals, business rules) (Wills & Cross, 1996).

8 Conclusion

Legacy software systems are difficult and costly to maintain. As many of these systems contain the business rules, the reengineering (or partial reengineering) process is necessary. Design and specifications need to be rediscovered, and then forward engineered to improved systems. Today's technology has shifted from structured (procedural programs) to object-oriented systems, which offer many advantages in the area of reduced maintenance costs and program reusability. The difficulty of recapturing program's design and specification, with understanding, is assisted by current CASE tools.

We have presented an overview of existing technology and tools including CASE tools that represent programs as abstract syntax trees, call graphs etc.; program slicing techniques, knowledge based approaches, approximate reasoning systems and inductive logic programming. Visualisation techniques can also aid in program understanding. The knowledge based approach has also been used frequently, and whilst it has been pointed out as being inadequate and very domain specific, this path is still being followed in current trends to convert procedural to object-oriented systems.

We have sketched an ongoing project ToRevise, whose focus is on using semi-automated and hybrid approaches. Intelligent assistants are required which can aid the reverse engineer to understand and recover program design. Candidate modules need to be selected, and assured of an explicit interface. Module migration can then proceed, where the engineer can reimplement the system with an improved design.

References

- [1] Advanced Software Technologies (1997) <http://www.advancedsw.com>, August.
- [2] Antoniol G., Fiutem R., Merlo E. & Tonello P. (1995) Application and user interface migration from basic to Visual C++. *International Conference on Software Maintenance*, IEEE Computer Society Press, Los Alamitos, Calif, p. 76-85.
- [3] Aslett M.J. (1991) *A Knowledge Based Approach to Software Development: ESPRIT Project ASPIS*. Elsevier Science Publishers, The Netherlands.
- [4] Bowdidge R.W. & Griswold W.G. (1994) Automated Support for Encapsulating Abstract Data Types. In *Proc. 2nd ACM SIGSOFT Symposium on Foundations of Software Engineering*, December, 19, 5, p. 97-110.
- [5] Beck J. & Eichmann D. (1993) Program and interface slicing for reverse engineering. In R.C. Waters & E.J. Chikofsky, editors, *Working Conference on Reverse Engineering*, IEEE Computer Society Press, p. 54-63.
- [6] Bennett K.H. (1993) Automated Support of Software Maintenance. In R.S. Arnold, editor, *Software Reengineering*, IEEE Computer Society Press, California, p. 59-70.
- [7] Biggerstaff T.J. (1993a) The concept assignment problem in program understanding. In R.C. Waters & E.J. Chikofsky, editors, *Proceedings Working Conference on Reverse Engineering*, IEEE Computer Society Press, p. 27-43.
- [8] Biggerstaff T.J. (1993b) Design recovery for maintenance and reuse. In R.S. Arnold, editor, *Software Reengineering*, IEEE Computer Society Press, California, p. 520-533.
- [9] Breuer P.T. & Lano K. (1991) Creating specifications from code: Reverse-engineering techniques. In *Journal of Software Maintenance: Research and Practice*, 3, p. 145-162.
- [10] Brodie M.L. & Stonebraker M. (1995) *Migrating Legacy Systems: Gateways, Interfaces & The Incremental Approach*. Morgan Kaufmann Publishers, Inc.
- [11] Burson S., Kotik G.B. & Markosoan L.Z. (1993) A program transformation approach to automating software re-engineering. In R.S. Arnold, editor, *Software Reengineering*, IEEE Computer Society Press, California, p. 275-283.
- [12] Chen Y. Nishimoto M.Y. & Ramamoorthy C.V. (1990) The C information abstraction system. In *IEEE Transactions on Software Engineering*, IEEE Computer Society, March, 16, p. 325-334.
- [13] Chen Yih-Farn (1995a) Reverse engineering. In B. Krishnamurthy, editor, *Practical Reusable Unix Software*, John Wiley & Sons, chapter 6.
- [14] Chen Y.R., Fowler G.S., Koutsofios & Wallach R.S. (1995b) Ciao: A graphical navigator for software and document repositories. In G. Caldiera and K. Bennett, editors, *International Conference on Software Maintenance*, IEEE Computer Society Press, p. 66-75.
- [15] Chikofsky E.J. & Cross J.H. (1993) Reverse engineering and design recovery: A taxonomy. In R.S. Arnold, editor, *Software Reengineering*, IEEE Computer Society Press, California, p. 54-58.
- [16] Chu W.C. (1993) A re-engineering approach to program translation. In D. Card, editor, *Conference on Software Maintenance 1993*, IEEE Computer Society Press, p. 42-50.

- [17] Cimitile A., De Lucia A. & Munro M. (1995) Identifying functions using specification driven program slicing: A case study. In G. Caldiera and K. Bennett, editors, *International Conference on Software Maintenance*, IEEE Computer Society Press, p. 124-133.
- [18] Cohen W.W. (1994a) Grammatically biased learning: learning logic programs using an explicit antecedent description language. In *Artificial Intelligence*, Elsevier Science, The Netherlands, August, 68, p. 303-366.
- [19] Cohen W.W. (1994b) Recovering software specifications with inductive logic programming. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, AAAI Press, 1, p. 142-148.
- [20] Feigenbaum E. (1977) The art of AI. In *Proceedings of the Fifth International Conference on Artificial Intelligence*.
- [21] Freeman P. (1987) A conceptual analysis of the Draco approach to constructing software systems. In *IEEE Transactions on Software Engineering*, 13, p. 830-844.
- [22] Gall H. & Klösch R. (1995) Finding objects in procedural programs: An alternative approach. In L. Wills, P. Newcomb & E. Chikofsky, editors, *Second Working Conference on Reverse Engineering*, IEEE Computer Society Press, p. 208-216.
- [23] Gamma E., Helm R., Johnson R. & Vlissides J. (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- [24] Harris D.R., & Yeh A.S. (1996) Extracting Architectural Features from Source Code. In *Automated Software Engineering - Special Issue: Reverse Engineering*, Kluwer Academic Publishers, June, 3, 1/2, p. 109-138.
- [25] Jacobson I. & Londström F. (1991) Re-engineering of old systems to an object-oriented architecture. In A. Paepcke, editor, *OOPSLA '91 Conference Proceedings*, Addison-Wesley Publishing Company, p. 340-350.
- [26] Jarzabek S. & Keam T.P. (1995) Design of a generic reverse engineering assistant tool. In L. Wills, P. Newcomb & E. Chikofsky, editors, *Second Working Conference on Reverse Engineering*, IEEE Computer Society Press.
- [27] Lano K.C. & Breuer P.T. (1993) Reverse engineering Cobol via formal methods. In H.J. van Zuhlen, editor, *The REDO Compendium*, John Wiley & Sons, chapter 16.
- [28] Lavrac N. & Dzeroski S. (1994) *Inductive Logic Programming Techniques and Applications*, Ellis Horwood.
- [29] Martin R. (1995) OO design quality metrics: An analysis of dependencies. In *ROAD*, Sep/Oct, <http://www.oma.com/Publications/publications.html>.
- [30] Maughan G. & Avotins J. (1995) A meta-model for object-oriented reengineering and metrics collection. Technical Report TR95-34, Department of Software Development, Monash University.
- [31] Menzies T. (1996) Patterns in inference strategies: Lessons and potential pitfalls for reuse design patterns. Technical Report TR96-9, Updated August 1996, Department of Software Development, Monash University.
- [32] Meyer B. (1988) *Object-Oriented Software Construction*. Prentice Hall, UK.
- [33] Michie D. & Sammut C. (1995) Behavioural clones and cognitive skill models. In K. Furukawa, D. Michie & S. Muggleton, editors, *Machine Intelligence 14*, Clarendon Press, chapter 15.
- [34] Mowbray T.J. & Zahavi R. (1995) *The essential Corba: Systems Integration Using Distributed Objects*. John Wiley & Sons, USA.
- [35] Murphy G.C., Notkin D. & Lan E.S. (1995) An empirical study of static call graph extractors. Technical Report 95-08-01, Department of Computer Science & Engineering, University of Washington.
- [36] Newcomb P. & Kotik G. (1995) Reengineering procedural into object-oriented systems. In L. Wills, P. Newcomb & E. Chikofsky, editors, *Second Working Conference on Reverse Engineering*, IEEE Computer Society Press, Canada, p. 237-249.
- [37] Quilici A. & Chin D.N. (1995) A cooperative environment for reverse-engineering legacy software. In L. Wills, P. Newcomb & E. Chikofsky, editors, *Second Working Conference on Reverse Engineering*, IEEE Computer Society Press, Canada, p.156-165.
- [38] Quinlan J.R. (1990) Learning logical definitions from relations. In *Machine Learning*, 5.
- [39] Reasoning Inc. (1997) <http://www.reasoning.com>, August.
- [40] Rochester J.B. & Douglas D.P. (1993) Re-engineering existing systems. In R.S. Arnold, editor, *Software Reengineering*, IEEE Computer Society Press, California, p. 41-53.
- [41] Rugaber S., Stirewalt K. & Wills L.M. (1995) The interleaving problem in program understanding. In

- L. Wills, P. Newcomb & E. Chikofsky, editors, *Second Working Conference on Reverse Engineering*, IEEE Computer Society Press, Canada, p. 166-175.
- [42] Rugaber S., Stirewalt K. & Wills L.M (1996) Understanding Interleaved Code. In *Automated Software Engineering - Special Issue: Reverse Engineering*, Kluwer Academic Publishers, June, 3, 1/2, p. 47-76.
- [43] Schwanke R.W. (1993) An intelligent tool for re-engineering software modularity. In R.S. Arnold, editor, *Software Reengineering*, IEEE Computer Society Press, California, p. 265-275.
- [44] Sneed H.M. (1995a) Planning the Reengineering of Legacy Systems. *IEEE Software*, 12, 1, January 1995, p. 24-34.
- [45] Sneed H.M. (1995b) Reverse engineering as a bridge to case. In L. Wills, P. Newcomb & E. Chikofsky, editors, *Second Working Conference on Reverse Engineering*, IEEE Computer Society Press, Canada, p. 300-313.
- [46] Storey M.D. & Müller H.A. (1995) Manipulating and documenting software structures using shrimp views. In G. Caldeira and K. Bennett, editors, *International Conference on Software Maintenance*, IEEE Computer Society Press, p. 275-284.
- [47] Tausend B. & Bell S. (1992) Analogical reasoning for logic programming. In S. Muggleton, editor, *Inductive Logic Programming*, Academic Press, chapter 19, p. 397-408.
- [48] van Zuhlen H.J. (1993a) Understanding in reverse engineering. In H.J. van Zuhlen, editor, *The REDO Compendium*, John Wiley & Sons, chapter 6.
- [49] van den Brand M.G.J., Klint P. & Verhoef C. (1997) Reverse Engineering and System Renovation - An Annotated Bibliography. In *ACM SIGSOFT Software Engineering Notes*, ACM Press, 22, 1, January, p. 57-68.
- [50] van Zuhlen H.J. (ed) (1993b) *The REDO Compendium*, John Wiley & Sons.
- [51] Vosselman G. (1992) *Relational Matching*, Springer-Verlag, Germany.
- [52] Weiser M. (1984) Program Slicing. In *IEEE Transactions on Software Engineering*, IEEE Computer Society, July, SE-10, p. 352-357.
- [53] Wells C.H., Brand R. & Markosian L. (1995) Customized tools for software quality assurance and reengineering. In L. Wills, P. Newcomb & E. Chikofsky, editors, *Second Working Conference on Reverse Engineering*, IEEE Computer Society Press, Canada, p. 71-77.
- [54] Wills L.M. & Cross J.H. (1996) Recent Trends and Open Issues in Reverse Engineering. In *Automated Software Engineering - Special Issue: Reverse Engineering*, Kluwer Academic Publishers, June, 3, 1/2, p. 165-172.
- [55] Yeh A.S., Harris D.R. & Reubenstein H.B. (1995) Recovering abstract data types and object instances from a conventional procedural language. In L. Wills, P. Newcomb & E. Chikofsky, editors, *Second Working Conference on Reverse Engineering*, IEEE Computer Society Press, Canada, p. 227-236.
- [56] Yourdon E. (1989) *Modern Structured Analysis*. Prentice-Hall, Englewood Cliffs, N.J.

Consciousness in Science and Philosophy 1998—“Charleston I”— Abstracts

Edited by Anton P. Železnikar and Mitja Peruš

November 6–7, 1998
Eastern Illinois University
Department of Mathematics, Charleston, IL

1 Keynote Lectures

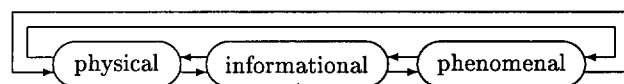
1 ANTON P. ŽELEZNIKAR. **Informational consciousness in philosophy, science, and formalism.** Volaričeva u. 8, SI-1111 Ljubljana, Slovenia.
Email locator: <s51em@lea.hamradio.si> at home, and <anton.p.zeleznikar@ijs.si>.
<<http://lea.hamradio.si/~s51em/>>.

How does consciousness as informational conceptualism reflect in philosophy, science, and the most pretentious symbolic, abstract, and powerful formalism? This paper brings an overview concerning informational supervenience and the most relevant philosophical, logical, and formalistic concepts concerning consciousness as a not-yet scientifically well-determined and philosophically sufficiently recognized object. The paper follows some principles of the established philosophism, scientism, and formalism as a developing form, approach and methodology of consciousness informationalism, that is, informational phenomenalism [4,5,6,7]. The following structured keyword list represents best the perplexed investigation possibilities, joining that what was thought and researched up to now in a new field of investigation, called informational consciousness:

- attention, consciousness, experience, intention, memory, mind, motivation
- Being (Sein), eidetics, entity (Seiende), essence (Wesen), the informational (i.)
- i. conceptualism, pure phenomenology, science, phenomenalism, i. formalism
- mentalism, consciousnessism, beingism, essentialism, existentialism, informationalism
- i. spontaneism (a metaphor for i. emergentism including change and decay), i. circularism, i. decompositionism (initial reduction, i. $\epsilon\pi\omicron\chi\eta$) and i. emergingism—being axiomatic, deductive, inductive, abductive, and inferential in a serial, parallel, and circular way—constituting the new emerging formal discipline called **informational mathematics**
- causalism, serial and parallel i. decomposition, circular causal informing (loops), perplexed serial, parallel, and circular spontaneous i. web causality, and
- i. formula, i. formula system, i. graph theory, i. topology and i. geometry

A logical objective scheme (in contrast to the serial one proposed by Chalmers [1]) of the general supervenience

concerning the physical (hard conscious), the informational and the phenomenal (soft conscious), seems to function bicircularly (circularly and bidirectionally), that is,



What has to be carefully analyzed is the single transition [3] of the form $\alpha \models \beta$ (an informational communication from α to β). The operator \models has to be understood as a consequence of two intersecting informational frames¹ [2], that is,

$$(\alpha \models \beta) \Rightarrow \left(\boxed{\alpha} \models \boxed{\beta} \right)$$

A reflection to the previously discussed supervenience would deliver

$$(\alpha \models \beta) \Rightarrow \left(\boxed{\boxed{\alpha}} \models \boxed{\boxed{\beta}} \right)$$

In this scheme, $\boxed{\alpha}$ represents a physical something and $\boxed{\beta}$ its phenomenal equivalent (percept). That what physically gives α as something's informing is $\boxed{\alpha \models}$. The informational coupling to the percept $\boxed{\beta}$ is $\boxed{\models \beta}$. Both, the physical α , in the form $\boxed{\alpha \models}$, and the phenomenal β , in the form $\boxed{\models \beta}$ enclose the informational operator \models .

A conceptual comparison of informational and philosophical terms delivers the following informational—philosophical pairs: informational (i.)—being; informing—Being; i. entity—Being-there; metaphysicalistic (m.)—self-subsistent; m. entity—Self; i. inclusivism—Being-in; circular—Being-in-itself; phenomenalistic informing—Being-in-the-world; spontaneous informing—Being-as-it-is; i. transition—Being-with-one-another (communication); i. formula—proposition; operand—predicate; operator—predicate junction; i. functionalism—Being-of; decomposition—deconstruction; etc.

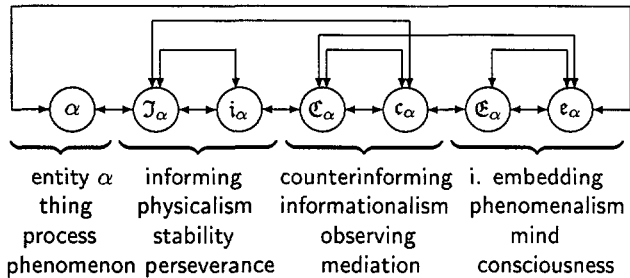
The paper brings informational conceptualism, technicalities, and formalism concerning operands, operators, transition, axioms, formulas, formula systems, serial and circular formulas, inclusivism, functionalism; serial, circular, and parallel decomposition; i. graphs, operand rotation principle; basic, standardized, general, and two-way metaphysicalistic decomposition; i. consciousness [4, 5, 7], metaphysicalistic decomposition of consciousness, i. emergentism, i. topology, i. investigation [6], one-way and multiple communication, and exclusive references to the theory

¹Metaphorically, the informational frame means an arbitrary something within an informational formula belonging together. How frames can be built up is shown transparently in [2,3].

and formalism.

In the paper the most important items concerning the last century philosophy and the informational approach, especially in a new formalistic form and meaning, are critically and cross-referentially discussed to some reasonable details.

For instance, informational metaphysicalism [2, 4, 5] is founded logically and superveniently [1]: logically in the sense of consciousness functioning and superveniently from the interrelatedness of the physical, informational, and phenomenal point of view. The following informatial graph shows evidently the metaphysicalistic corelates:



This case shows a broad applicability of the metaphysicalistic approach in different diciplines (philosophy, psychology, modeling of consciousness [4, 5, 7], etc.).

References

[1] CHALMERS, D.J. 1996. *The Conscious Mind*. In *Search of the Fundamental Theory*. Oxford University Press. Oxford, U.K.
 [2] ŽELEZNIKAR, A.P. 1996. Informational frames and gestalts. *Informatica* 20:65–94.
 [3] ŽELEZNIKAR, A.P. 1996. Informational transition of the form $\alpha \models \beta$ and its decomposition. *Informatica* 20:331–358.
 [4] ŽELEZNIKAR, A.P. 1997. Conceptualism of Consciousness. *Informatica* 21:333–344.
 [5] ŽELEZNIKAR, A.P. 1997. Informational Theory of Consciousness. *Informatica* 21:345–368.
 [6] ŽELEZNIKAR, A.P. 1997. Informationelle Untersuchungen. *Grundlagenstudien aus Kybernetik und Geisteswissenschaft/Humankybernetik* 38:147–158.
 [7] ŽELEZNIKAR, A.P. 1997. Informational Consciousness. *Cybernetica* 40:261–296.

2 JAMES H. FETZER. **Consciousness and cognition: Semiotic conceptions.** Department of Philosophy, University of Minnesota, Duluth, MN 55812. <jfetzer@d.umn.edu>

Inference to the best explanation involves selecting an hypothesis or theory from among a set of alternatives as the hypothesis or theory that provides the best explanation for the available evidence. Alternatives that explain more of the evidence are preferable to those that explain less, while those that are incompatible with the evidence are rejected. With respect to consciousness and cognition, considerations adduced within this study suggest that the computational conception confronts insuperable problems, which

stem from inadequate understanding of the nature of psychophysical laws and from static and dynamic differences between the modes of operation of digital machines and of thinking things. The theory of minds as semiotic systems explains more of the available evidence and thus provides a preferable theory.

The most sophisticated version of the computational conception bearing directly upon these issues appears to be David Chalmers, *The Conscious Mind*. Systems stand in relations of simulation when they produce the same outputs from the same inputs. Systems stand in relations of replication when they both produce the same outputs from the same inputs and do so by means of the same processes or procedures. According to Chalmers, systems that satisfy the condition of organizational invariance must stand in relations of replication, necessarily, and also possess the same associated psychophysical properties. Chalmers, however, overlooks the consideration that at least some of those psychophysical properties may depend upon the stuff of which such systems happen to be made. His conclusion, therefore, does not follow.

The author attempts to clarify the nature of psychophysical laws and to overcome the obstacles that confront computational conceptions on the basis of the theory of minds as semiotic (or sign-using) systems, which implies the existence of at least three kinds of minds. These appear to correspond to levels of mentality that can be found distributed throughout nature, where species that are lower on the evolutionary scale tend to have lower grades of mentality, while those higher on the scale tend to have higher. Semiotic accounts of consciousness and of cognition provide a framework for solving the mind/body problem and for coping with the problem of other minds, building on the author’s earlier work, including “Signs and Minds”, *Philosophy and Cognitive Science*, “Mental Algorithms”, and “Thinking and Computing”.

References

[1] CHALMERS, D. 1996. *The Conscious Mind*. Oxford University Press. NY.
 [2] FETZER, J.H. 1988. Signs and minds. In J. Fetzer, Ed. *Aspects of Artificial Intelligence*. 133–161. Kluwer Academic Publishers. Dordrecht. The Netherlands.
 [3] FETZER, J.H. 1994. Mental algorithms: Are minds computational systems? *Pragmatics & Cognition* 2:1–29.
 [4] FETZER, J.H. 1996. *Philosophy and Cognitive Science*. 2nd Ed. Paragon House. Minneapolis, MN.
 [5] FETZER, J.H. 1997. Thinking and computing: Computers as special kinds of signs. *Minds and Machines* 7:345–364.

2 Cybernetic Concepts of Consciousness

3 JERRY L. CHANDLER. **On the evolutionary roots of consciousness.** Krasnow Institute for Advanced Study, George Mason University, 837 Canal Drive, McLean, VA 22102, U.S.A. <jlrchand@erols.com>

Awareness will be taken as a root of consciousness. A definition (Shorter Oxford, 1993) is "responsive to conditions". Here, we presuppose that continuous awareness follows from the dynamics of a historical organization of structures and initial conditions. In earlier writings, the C^* hypothesis was described in terms of third order cybernetic systems with a minimum of three degrees of organization. Here, I presuppose the logical structure and organization of C^* as described elsewhere (Chandler, 1991, 1992, 1996, 1997, 1998, Chandler, Ehresmann and Vanbremeersch, 1996). Any system is considered as a mathematical object (category) within a degree of organization, $O_1^o, O_2^o, O_3^o, \dots$.

Pre-pre-primitive responsiveness (PPPR) in simple systems (O_1^o, O_2^o, O_3^o) includes responds to conditions. Awareness is locally defined at the atomic scale, in terms of closure, conformation, concatenation and cyclicity. Such systems follow the second and third principles of thermodynamics and functions can be described in terms of energy and work. Locally, one form of awareness is spectra, systematic responses to electromagnetic energy and fields (in terms of quantum rules.) Equilibrium behaviors are intrinsic to quantum descriptions. System composition may vary by gain and loss of parts. Parts (components) may be either symmetric with respect to the whole or not.

Pre-primitive responsiveness (PPR) in biopolymers is slightly more sophisticated than of PPPR. Responses to changes in initial condition can greatly accelerate or decelerate flow of matter and energy. Collaboration (co-operative work) among molecules systems (i.e., O_1^o, O_2^o, O_3^o and O_4^o) changes the response times but not the thermodynamic equilibrium values.

Primitive responsiveness (PR) in a simple (prokaryotic) cell responds to external conditions in an intentional pattern with growth, movement and creative changes in material composition. Awareness to the chemical and physical environment is evaluated by cyclic communication channels and decisions are locally implemented by changes in behavior. Awareness of the environment (sources of nutrients) binds primitive awareness to the conservation laws and the "web of life". Exponential growth creates a congenitive thermodynamic system, which accumulates organized matter and energy via biopolymeric operations of material addition and multiplicative dynamic influences. Local behaviors of PPPR, PPR, and PR follow conservation rules.

Sensory awareness (SA) in higher organisms responds to external stimuli concomitantly. Categories of PPPR, PPR, and PR may be integrated to generate an evaluation of the intensity, coherence and value of external events. Decisions within the pre-responsiveness systems select from a range of generated options. Responses (with respect to specific stimuli) may be (chaotically) amplified or suppressed in intensity, duration, and scale. Pre-responsiveness systems may co-ordinate implementation of responses over substantial durations and substantial spatial domains. Pre-existing organized structures and stores of energy control the internal dynamics of response.

Non-human mammalian awareness is continuously composed from electromagnetic, physical, chemical and biological sensory systems. Categories of PPPR, PPR, and PR are integrated via sequences of PRS after evaluation of the intensity and value of external events. Internal memory al-

lows historical record keeping and context-specific access. Following central decisions, responds are staged in terms of the short, intermediate and long range needs of the organism. External communications are planned and executed.

(I am grateful to Professor Andree Ehresmann whose category-theory based "memory evolutive systems (MES)" have strongly influenced the development of these ideas.)

4 BRUCE MACLENNAN. **The protophenomenal structure of consciousness, with especial application to the experience of color.** Computer Science Department, University of Tennessee, Knoxville, TN 37996-1301, U.S.A.

<macleNNAN@cs.utk.edu>.

<http://www.cs.utk.edu/~mclennan>.

Introduction. This paper addresses the *principal problem of consciousness*, which is to reconcile our experience of subjective awareness with the scientific world view; it is essentially the same as Chalmers' "Hard Problem." This problem arises because subjective experience has a special epistemological status, since it is the personal (and private) substratum of all observation, whereas empirical science is typically based on common (nonpersonal, public) particular observations. Nevertheless, although subjective experience cannot be reduced to physical observables, we may have parallel phenomenological and physical reductions, which inform each other. However, naive introspection is treacherous since it may be unduly influenced by theoretical preconceptions, but phenomenological training aids unbiased (or less biased) analysis of the structure of consciousness. Through phenomenologically trained observers we may acquire unbiased (public) data about the structure of consciousness. (We use "phenomenology" and related terms in the sense of Husserl and Heidegger, that is, to refer to the analysis of the *phenomena*, the given (*data*) of conscious experience.)

Protophenomena. Such an analysis has led us to postulate *protophenomena* as the elementary constituents of phenomena [1, 2]. Each has the property of elementary (irreducible) subjectivity. Very simple examples of protophenomena include the experience of a spot of color at a particular location in the visual field and the feeling of pressure at a particular location on the skin. However, there are much more complex and subtle protophenomena, including elementary components of recognitions, judgments, expectations, intentions, moods and so forth. Further, protophenomena are very "small," in the sense that changes in the activity of individual protophenomena will not typically affect the macroscopic phenomenal state; nevertheless the state of consciousness is no more than the sum total of the states of all the protophenomena.

Protophenomena are postulated to be associated with *activity sites* in the brain, the "activity" (degree of presence in consciousness) of a protophenomenon corresponding to some physical variable at that site. (Protophenomena and their activity sites need not be discrete, but that seems the most likely possibility at this time.) There are a number of candidates for the activity sites, but their identity remains an open question. Some of the possibilities include synapses, neural somata and dendritic microtubules, but their exact identity is not crucial for the theory of protophenomena.

What is the ontological status of protophenomena; do they exist? It is best for now to treat protophenomena as "theoretical entities," analogous to atoms when they were first hypothesized. Theoretical entities are validated by the role they hold in the theory and by their fruitfulness for scientific progress. Ultimately, we may find that protophenomena exist individually (in the same way that atoms were found to exist), e.g. as properties of individual activity sites, or we may find that protophenomena

exist only in the context of large numbers of activity sites, and thus that they are emergent properties, analogous to emergent physical properties. For now this is an open question.

Causal dependencies among activity sites suggest how protophenomena are integrated into a phenomenal world. Just as physical processes in an activity site depend on physical processes in other activity sites, as well as on extrinsic processes (e.g. in sensory neurons), so the activity of a protophenomenon depends on the activities of other protophenomena, as well as on variables that are not directly dependent on protophenomenal activity (i.e., variables associated with the external world). The dynamics of protophenomenal activity can be described by differential equations. In many cases the dependencies (the equations) are approximately linear, and protophenomenal activity can be described in terms of a *characteristic function* (often known as an "impulse response").

Protophenomenal dependencies establish connections among protophenomena and thereby assemble them into a phenomenal world. One way they do this is by establishing continuity through expectations. Another way is by means of conjunctive dependencies and by more complex temporal dependencies. As a result a phenomenal world may be described by a set of possible trajectories in protophenomenal state space.

In summary, the fact of phenomenal experience corresponds to a protophenomenon's activity, since that activity represents its degree of presence in conscious experience; the quality of conscious experience corresponds to the protophenomenon's dependencies, which relate it to other protophenomena.

Color and Spectral Inversions. As an example of the protophenomenal approach, we can consider the well-known problem of a spectral inversion. In brief, the problem is as follows: Although we agree on the names for various wavelengths, is it possible that you experience red wavelengths the same way I experience blue wavelengths, and vice versa? Before we can solve this problem we need a more accurate phenomenology of color. The plausibility of a spectral inversion derives in part from an oversimplified phenomenology of color, since we have imagined that color can be reduced to a single dimension (wavelength) but a phenomenological analysis shows it to be much more complex (see [3] and the references cited therein).

Setting aside many of the higher-level complexities of color (e.g. its emotional and cultural connotations), yet avoiding the trap of a one-dimensional view, we can observe that it has long been known that we can identify four pure hues, which are termed the "unique hues," an observation that has led to the double-opponent theory of color vision. In this theory the three color receptors (short, medium and long wavelength, henceforth S , M and L) are combined in various ways to yield three orthogonal axes. The light-dark axis is formed by $S + M + L$ and its opposite; the yellow-blue axis is formed by $M + L - S$ and its opposite; the red-green axis is formed by $S + L - M$ and its opposite (here we use a common form of the theory). The two zeroes on each of the two chromic axes (yellow-blue and red-green) define the four unique hues.

The problem of a spectral inversion can be recast in terms of inversions between the poles on one or more of these axes or in terms of exchanges between two or more of the axes. However, we will show by phenomenological analysis that such spectral inversions are impossible, that is, that abnormal neurological connections would lead to abnormalities in conscious experiences that could be detected by the subject. Here the arguments will be summarized briefly.

First, it is fairly obvious that dark and light have phenomenologically distinct characters, and hence are noninterchangeable: in the dark, forms and hues are indistinguishable, but not in the light.

Second, phenomenological analyses of color from ancient times to our own have observed that yellow is intrinsically brighter than blue (the neurophysiological reason being the large overlap between $S + M + L$ and $M + L - S$). Hence, blue and yellow are phenomenologically similar to dark and light, and hence noninterchangeable. Therefore, in a case of abnormal vi-

sion, whatever receptor combination has the largest overlap with $S + M + L$ will be experienced as phenomenal-yellow, and if this does not correspond to spectral-yellow then the anomaly will be detectable.

The case of a red-green inversion is more subtle, but phenomenological analysis again exposes a difference. For example, Goethe observed that green is a phenomenological mixture of yellow and blue, whereas red results from an "augmentation" of yellow and blue. Further, the experience of "unique red" is nonspectral; that is, it cannot be created by monochromatic light, whereas experience of the other three unique hues (including green) can. (The well-known studies of Berlin and Kay also support the phenomenal differences between red and green.)

Finally, the red-green axis cannot be exchanged with the yellow-blue, because the former is less similar to light-dark than the latter. This phenomenological fact, which has been recognized since ancient times, is consequence of $S + L - M$ ("red") having a smaller overlap with $S + M + L$ ("light") than does $M + L - S$ ("yellow").

As a result of this neurophenomenological analysis, we can begin to understand the topology of color. First we have the three axes, which define three polar oppositions: light-dark, yellow-blue, red-green. Superimposed on this structure are relations of similarity: yellow is most similar to light, and blue is most similar to dark. Green is most similar to yellow and blue and is intermediate in its similarity to light and dark. Red is similar to yellow, but not to blue. These conclusions are objective in that they result from observations made independently by many phenomenologists over the centuries.

Finally, we will consider several more examples of abnormal or nonhuman color perception. For example, if we have $S + M - L$ instead of $M + L - S$ in the yellow-blue channel, then spectral blue-greens will be experienced as yellows, and spectrally orange light will be experienced as green. On the other hand, if we have $S + M - L$ and $M + L - S$ (two asymmetric channels) for the chromic channels, then color phenomenology will have several detectable anomalies: there will be two spectral unique hues (as opposed to three) and one whole phenomenal color quadrant (purple) will be nonspectral (unexperiencable with monochromatic light).

Many other neural anomalies can be hypothesized. However, if a sensory system is too different from our own, we may be neurologically unable to imagine the experience, although we can describe its topology. Since imaginal areas have parallel structures to perceptual areas, we have limited ability to imagine qualia that are distinctly different from what we can perceive.

Conclusions. The protophenomenal perspective has several benefits. First, it allows the fact of conscious experience to be integrated into scientific theory without denying or distorting the nature of that experience. Second, it permits a form of reduction of the more complex to the simpler while acknowledging the complexity of phenomena and avoiding naive introspectionism. Third, it permits a detailed account of the structure of conscious experience.

Of course, many open questions remain. For example: What are the activity sites and what sorts of physical systems can be activity sites? (This has implications for nonbiological consciousness.) What distinguishes conscious from nonconscious neural activity? Are protophenomena emergent? (This has implications for degrees of consciousness.) Are protophenomena qualitatively exhausted by their mutual dependencies (structuralism)? What can we say about the boundaries and unity of consciousness? Finally, much detailed neurophenomenological work remains to be done before we will understand the detailed structure of consciousness.

References

- [1] MACLENNAN, B.J. 1995. The investigation of consciousness through phenomenology and neuroscience. *In* J. King & K. H. Pribram, Eds. *Scale in Conscious Experience: Is the Brain Too Important to be Left to Specialists to Study?* Lawrence Erlbaum, Hillsdale.

[2] MACLENNAN, B.J. 1996. The elements of consciousness and their neurodynamical correlates. *Journal of Consciousness Studies* 3:409-424.

[3] MACLENNAN, B.J. 1998. Finding order in our world: The primacy of the concrete in neural representations and the role of invariance in substance reidentification. *Behavioral and Brain Sciences* 21:78-79.

5 BEN WEEMS. **The genesis of analogical and logical operations in sensory-motor organisation.** McGill University.

The paper deals with the genesis of two basic modes of organisation: call them metaphor and proto-logic; not yet logical operations, only the basis on which they will be formed: call them Both/And and Either/Or, for the moment. These two basic modes of order, when reflected at another level of organisation, after the development of logical operations, will appear as logical disjunction and logical conjunction, but I wish to examine their roots in earlier pre-logical processes. Metaphor can be described as the process of gradually differentiating the common ground AB, from its initially wholly implicit existence, in the Venn diagram of logical disjunction, to its eventual explicit distinct existence in the Venn diagram of logical conjunction. But those two logical operations themselves do not allow for such a process. How do the two primitive modes of order operate in practice, and how do they bring about the possibility of logical operations? It substantially follows Piaget's description of this process in the child, but refers it to cultural history for additional data, as Piaget repeatedly called for.

The intent of the paper is to adduce a body of data that is not normally used in such discussions at all; that of intellectual history. The reasons why those data are not used are plain to see: on the one hand, the truth is what is true in all times and places, else what could we mean by truth? On the other hand, the history of ideas means entering into a world in which the truth is a function of the terms which are current in any given period. I will suggest that room must be found for both of these requirements; for the implications of logic, and for the history of ideas, which evolves ultimately, by the "logic" of metaphor. Examples of that metaphoric process: at most trivial level, the etymological history of the word "astonish", from a live metaphor to a dead abstraction, and then reduplicated in "thunderstruck". At a serious level, the evolution of the fate of Newton's Absolute Time and Space, and their replacement in Special Relativity.

Sources of the common process in Pre-linguistic organisation, à la Piaget, and his description of the reiteration of that sensorimotor process on itself twice, at the level of symbolic organisation, creating the two symbolic modes, metaphor and something else, that will become logic. The result of this first process is to differentiate the subject from the outer world against some further body of neural organisation that is invariant in the two sets of contexts, outer world and the body image of the subject.

Symbolic order is the following: a matter of giving progressively more distinct form to elements of that common ground against which subject and world are differentiated; two forms or stages of this process; examples of these two processes, at different levels of generality, from different societies in the past will be given; the way specific symbolic forms mediate the shift from one overall paradigm of organisation to another at the last level of generality in our own culture, and the degree to which the relation of metaphor underlies and must always underlie, those of logic, as for example, in the relation between the equation, and the phenomena which it has been used to illuminate; and the relations between the emergent further common ground and its two contexts, subject and world, are always changing, and only an analysis of their historical course can produce a picture of the process.

But that body of further invariant neural organisation can never be anything but an attribute of the body image of the subject, and an aspect of the terms in which that subject orders its experience. The underlying paradox of all symbolic order, then, is that to become invariant in all contexts, and so an objective fact of the world, invariant in all coordinate systems, for all observers, is to become an attribute of the body image of the subject, as in the case of Newton's Absolute Time and Space.

The first considers the emergent common ground as the potentially material and factual common ground of subject and object, the process of its gradual ever more explicit embodiment, that of metaphor. The limit of that process is to discover that the final invariant in all man's experience is himself, man, the entire sentient creature, and so there is no more common ground of man and world, and the basic paradigm of Both/And, of the first mode of order, is now transformed into that of Either/Or, mere subjective impression, or the world in itself. This paradigm of Either/Or, of the excluded middle, and so of the eternal choice between appearance and reality, recurs at every instant. It is also the birth of two mutually exclusive modes of order, that of the live unit of change of state, the unit of input, or change of state, and the unit that has become the unit of code, which will in various incarnations give rise to the distinctions of matter and motion, objects and events, nouns and verbs, among others. It is the order of logical operations.

The limit of the second process is to arrive at a new final invariant in all our experience, from all points of view, for all observers, and so absolutely objective, at just the point where we discover that it is not a last fact of the world, but of the way men organise their experience of the world, and so has become a new symbolic form, in that it is at once all we know of the world, yet is at the same time an attribute of the subject, and so can be identical with neither. The two part process starts over again at a new level of generality.

3 Physical Background of Consciousness

6 MENAS KAFATOS. **Foundational principles and consciousness.** Center for Earth Observing and Space Research & Institute for Computational Sciences and Informatics, MSN 5C3, Science and Tech I, Room 301, George Mason University, Fairfax, VA 22030-4444, U.S.A. <mkafatos@compton.gmu.edu>. <<http://www.ceosr.gmu.edu>>.

An axiomatic approach is proposed for addressing the problem of consciousness. If indeed the underlying reality of the universe is a background of consciousness out of which all objective processes arise [1, 2], attempting to define this background in terms of its derivable theoretical physical constructs is futile. As such, the singular subjective experience, the so-called "hard problem", might not be describable in terms of mathematical algorithms. Yes, we know that the physical universe is quite successfully describable in terms of theories which are mathematical in nature, i.e. mathematics is the best language for our views of the universe.

Perhaps one should approach the problem of consciousness by searching for a "pre-mathematical" background out of which all mathematics arise. In other words, another approach might be to look at foundational principles which might be informative of properties of consciousness as we know them. These principles should still be linked

to mathematics but in a *much more general, qualitative sense*. They can be considered similar to the axioms in *specific geometries*, except in this case the principles are much more general and in analogy the “theorems” are actually complete theoretical world constructs (such as the quantum theory, etc.).

We have identified several such principles: Simplicity; Self-organization; Complexity; Complementarity; and Non-locality; Union/Wholeness. This list is certainly not complete and one could even come up with different constructs and certainly different names. For example, Simplicity could also be termed Reductionism; Union/Wholeness is a more scientific term for Love, the core of all subjective experiences, etc. One requirement of the proposed approach is to come up with foundational principles that apply across many different domains (such as physical theories, e.g. principles that apply equally well to quantum theory, classical theory, etc.) and at the same time are connected to both the subjective and objective experiences. This approach has to be more qualitative and certainly more subjective than the usual deductive methodology of science and yet should be tied to science in a most fundamental way.

We suspect that the proposed approach might constitute a new way for moving forward with the scientific framework and to put this framework within the conscious experience, rather than the other way around (the latter is to date the usual approach and might just be a totally wrong approach).

References

[1] M. KAFATOS, M. & R. NADEAU. 1990. The conscious universe: Part and Whole in modern physical theory. Springer-Verlag. New York.

[2] DRAGANESCU, M. 1997. On the structural-phenomenological theories of consciousness. Noetic Journal 1:28–33.

7 EVAN HARRIS WALKER. **Consciousness as a factor in the modified Schrödinger equation.** Walker Cancer Research Institute Inc., Aberdeen, Maryland 21001, U.S.A. <wcri@erols.com>.

The problem of consciousness involves three major issues: (1) philosophical problems, (2) mind-brain interactions, and (3) consciousness relationship to the basic equations of physics. Each of these entails difficulties unlike those encountered elsewhere in efforts to understand nature. Each of these areas has been dealt with in the quantum mechanical theory of consciousness developed by the present writer. The philosophical problems include the monism/duality and the binding/identity problems. Monism/duality is resolved by limiting the definition of “physical reality” to things physically measurable as defined by physics; binding physical events so as to produce consciousness identity is resolved by appeal to quantum theory. The mind-brain interaction is treated as arising from QM tunneling—at and between synapses. Tunneling lets us treat the mechanisms of consciousness quantitatively and in detail so that

comparison between experimental data and the characteristics of consciousness can be made. Finally, determining the relationship of consciousness to the basic equations of physics is achieved by introducing a modification to the Schrödinger equation. The modified Schrödinger equation (MSE) results from the addition of an information measure term, $\Psi \ln \Psi$ that doubles as a dispersion measure. The MSE allows for automatic state vector collapse on completion of measurement loops. “Measurement loops” refers to interactions of the Einstein-Podolsky-Rosen type, wherein resolution of the EPR paradox entails measurement completion in both parts of the interacting system, and a comparison of the measurement results. The $\Psi \ln \Psi$ term then takes the form $\Psi \Psi^\dagger \ln \Psi \Psi^\dagger$, a real quantity, forcing the system into automatic collapse—and the term vanishes. Introducing a coefficient matrix G , having 0’s off diagonal, -1 on diagonal for the selected term, and 1’s otherwise, represents state selection. The minimum G matrix is 4×4 , and of the form of the Dirac gamma matrices in the uncollapsed MSE. Thus, consciousness may underlie the space-time continuum.

8 JOHN S. HAGELIN. **Is consciousness the unified field? A field theorist’s perspective.** Department of Physics & Institute of Science Technology and Public Policy. Maharishi University of Management, 1000 North 4th Street, Fairfield, IA 52557, U.S.A. <jhagelin@mum.edu>. <<http://www.mum.edu>>.

Advances in theoretical physics during the past decade have led to a progressively more unified understanding of the laws of nature, culminating in the recent discovery of completely unified field theories [1–6]. The parallel discovery of a unified field of consciousness raises fundamental questions concerning the relationship between the two. Following a general introduction to unified quantum field theories, we consider the proposal due to Maharishi Mahesh Yogi that the unified field of modern theoretical physics and the field of “pure consciousness” are identical. We show that the proposed identity between consciousness and the unified field is consistent with all known physical principles, but requires an expanded physical framework for the understanding of consciousness [7,8]. Such a framework may indeed be necessary to account for experimentally observed field effects of consciousness and phenomenological aspects of higher states of consciousness. Finally, we will consider the important question of a “technology” of the unified field—i.e., a practical means of applying this most advanced and fundamental knowledge of natural law and consciousness for the benefit of mankind [9].

References

[1] HAGELIN, J.S. 1989. Superstring Grand Unification. MIU Preprint 89/41. Proceedings of 1989 Superstring Workshop. Texas A&M.

[2] ANTONIADIS, I., J. ELLIS, J.S. HAGELIN & D.V. NANOPOULOS. 1989. The Flipped $SU(5) \times U(1)$ String

Model Revamped. *Phys Lett* 231B:65–74.

[3] HAGELIN, J.S. 1991. Recent Progress in Realistic 4-D String Model Building. Proceedings of the 1991 PASCOS Conference. Boston, MA, U.S.A.

[4] HAGELIN, J.S., S. KELLY & T. TANAKA. 1993. Exact Supersymmetric Amplitude for K^0-K^0 and B^0-B^0 Mixing. *Mod Phys Lett A* 8:2737–2745.

[5] HAGELIN, J.S., S. KELLY & T. TANAKA. 1994. Supersymmetric Flavor Changing Neutral Currents: Exact Amplitudes and Phenomenological Analysis. *Nucl Phys B* 415:293–331.

[6] CONNORS, L., A.J. DEANS & J.S. HAGELIN. 1993. Supersymmetry Mechanism for Naturally Small Density Perturbations and Baryogenesis. *Phys Rev Lett* 71:4291–4294.

[7] HAGELIN, J.S. 1987. Is Consciousness the Unified Field? A Field Theorist's Perspective. *Modern Science and Vedic Science* 1:28–87.

[8] HAGELIN, J.S. 1989. Restructuring Physics from Its Foundation in Light of Maharishi's Vedic Science. *Modern Science and Vedic Science* 3:3–72.

[9] HAGELIN, J.S., M.V. RAINFORTH, D.W. ORME-JOHNSON, K.L. CAVANAUGH & C.N. ALEXANDER. Effects of Group Practice of the Transcendental Meditation Program on Preventing Violent Crime in Washington, DC. Results of the National Demonstration Project. June–July, 1993. *Journal of Offender Rehabilitation*. In press.

9 STEPHEN THALER. **The fragmentation of the universe and the devolution of consciousness.** Imagination Engines, Inc., St. Louis, MO 63146-4331. <sthaller@ix.netcom.com>. <<http://www.imagination-engines.com>>.

In a previously published paper (Thaler, 1995), I have described how any trained neural network, whether artificial or biological, preferentially interprets spontaneous fluctuations within its synaptic connections and processing units as either learned or confabulated memories. Thus, the sustained random and internal perturbation of such a network provides a compelling, all-neural model for how cortical networks spontaneously manufacture a succession of thoughts and ideas, loosely coupled to perceived external stimuli, to produce stream of consciousness. Realizing that the inorganic universe may likewise be viewed as a connectionist system, bound together by various physical forces and evolving into distinct dynamical states in response to internal chaos, stream of consciousness may be viewed as a simplistic simulation of the perceivable cosmos, implemented through electrochemical connectivity. Biological organisms utilize this neurological emulation of the external world to advantageously anticipate potential environmental scenarios impacting their existence.

Apart from any cultural mythologies that attempt to explain and glorify the emergence of human consciousness, we must inevitably ask why the physical world has compartmentalized into pieces that must anticipate the activity within other pieces, using such inner simulations? To answer this question, we conduct a thought experiment

in which some portion of the inorganic universe topologically pinches-off from the whole, to produce a parallel-processing island surrounded by connection sparse boundaries through which only sporadic and impoverished information exchange may occur. Supplied only scanty clues about the state of the external world, abundant internal chaos drives this region into any of a multiplicity of physical states for any given condition of the external universe. Therefore, in lieu of an accurate knowledge about the universe at large, this insular region engages in 'bad guesses' as to the state of the external environment, in a process we commonly acknowledge and mistakenly celebrate as perception. With further spontaneous subdivision of this region by new semi-insulating partitions, the resulting submodules activate to lend interpretation to each other's activity. Now a conglomerate entity, this topologically distinct region may spontaneously invent significance to its own overall collective behavior in a process that we have grown to recognize as biological consciousness.

Considering such a fragmentation process ubiquitous and ongoing, similar regions detaching from the whole that do not invent merit to their own collective behavior take no action in preserving their topological boundaries, inevitably recombining with the external universe. Those that do accidentally interpret their inner activity as significant, proliferate and develop additional survival apparatus. Ultimately, the world naturally segregates into the uniform inorganic background we call unconscious, and the highly delusional protoplasmic islands we glorify as conscious. At the core of this self-preserving illusion is the inherent capacity of neural networks to map any input space to any other. Here, generic, global activation turnover within one set of neural networks is mapped to activation patterns within other neural networks, to produce the qualia of being, self-awareness, and the general subjective feel of consciousness (Chalmers, 1996).

Certainly, such a perspective suggests not only that biological consciousness is a degraded form of universal connectionism, but that the resulting intellect is highly suspect and largely based upon neurologically habituated metaphor (i.e., long-standing inner simulations that are inherently faulty or ambiguous). Therefore, apart from deducing the micro-features of cognition, the human quest for a science of consciousness is doomed, since the analogy machine, the brain, can at best produce an analogy for itself. Such a perspective would be consistent with the late Paul K. Feyerabend's philosophical view of science as a collection of useful myths (Horgan, 1993). Furthermore, the door is now opened to an acknowledgement that consciousness may not be a uniquely human or even organic phenomenon. For decades, this point of view has been advanced by such scientist-philosophers as Olaf Stapledon, Arthur C. Clarke, and Carl Sagan, through the conjecture of a "cosmic consciousness" (Ash, 1977). In this presentation, I offer a plausible model for how human consciousness has arisen by fragmentation and degradation of just such a prototypical master consciousness.

References

[1] CHALMERS, D. J. 1996. *The conscious mind: in search of a fundamental theory*. Oxford University Press. New York.

- [2] HORGAN, J. 1993. The worst enemy of science. *Scientific American* 268:5:36–37.
- [3] ASH, B. 1977. *The visual encyclopedia of science fiction*. Harmony Books. New York.
- [4] THALER, S. L. 1995. “Virtual Input Phenomena” within the death of a simple pattern associator. *Neural Networks* 8:1:55–65.

10 RICHARD L. AMOROSO. **The role of gravitation in the dynamics of consciousness.** The International Noetic University, Physics Lab 610, 16th Street #506, Oakland, CA 94612–1229, U.S.A.

<ramoroso@hooked.net> preferred, and
<ramoroso@wenet.net>.

Both general relativity and quantum theory are known to be incomplete. Twistor and superstring theories are currently considered the most promising candidates for their nonlocal integration; but both lack a ‘Rosetta stone’ for delineating the unique topological package of higher dimensional hyperstructure required to complete the task. Both eastern and western theologies claim that gravitation is caused by the movement of spirit—spirit, ki, chi, or prana not immaterial but Bose or photon based. The spin exchange model of quantum gravity [1] incorporating the expanded Wheeler-Feynman absorber theory of radiation [2,3] putatively describes gravitons as superposed moments of confined nonlocal photons mediated by unitary field dynamics. The cosmological constant is the coupling constant between both domains; the zero averaged fluctuation of the gravitational potential localizes and delocalizes the flow of conscious energy. The lower limit for the quantization of mind is a Planck scale hypercavity where the gravitational potential may remain balanced when at rest. Gravitational mass dependency is not required by conscious entities for state evolution as in the Hameroff-Penrose Orch-Or model because spacetime curvature provides boundary conditions gating the energy flow of Bose psychons. No gravitational work is required, mental activity is frictionless at this level, but not at the higher organic species level. Radiation or light pressure is sufficient to modulate the boundary conditions. All levels of scale are proportional to the elemental Planck unit through the law of energy quantization. Degenerate energy from infinite density singularities not being applicable to consciousness. Thus one may whimsically query ‘How many Einstein’s (moles of photons) does it take to turn on a light bulb?’ The bulb being a ‘byte’ of Planck bits pertinent to the conscious scale of the entity. Thus entity Z with a 10^n Planck byte raster of consciousness, has a 10^n byte psychosphere and resolves factors of 10^n bytes of external and internal mental events. Thought being dynamic moments of local quantization and summation of conscious energy. This is the cosmological root of consciousness. Nothing is said about the branches in quantum brain dynamics and nonlinear neurodynamics essential to higher brain function and the phenomenological inputs of awareness. An experimental protocol has been developed to empirically isolate and extracellularly confine the Bose psychon [4,5].

References

- [1] AMOROSO, R.L., M. KAFATOS & P. EČIMOVIČ. 1998. *In* G. Hunter & S. Jeffers, Eds. *Causality and Locality in Modern Physics*. Kluwer Academic. In press.
- [2] WHEELER, J.A. & R.P. FEYNMAN. 1945. *Rev. of Modern Physics* 17:157.
- [3] CRAMER, J.G. 1986. Transactional interpretation of quantum mechanics. *Rev. of Modern Physics* 58:3:647–687.
- [4] AMOROSO, R.L. 1996. *Bioelectrochemistry & Bioenergetics* 40:39–42.
- [5] AMOROSO, R.L. 1995. *Informatica* 19:585–90.

11 RICHARD L. AMOROSO. **The feasibility of constructing a conscious quantum computer.** The International Noetic University, Physics Lab 610, 16th Street #506, Oakland, CA 94612–1229, U.S.A. & Cerebroscopic Systems, Inc., 48 Shattuck Square # 148, Berkeley, CA 94704–1119.

<ramoroso@hooked.net> preferred, and
<cerebroscopic@mindspring.com>.

Extensions of quantum theory (QT) and Einstein’s general theory of relativity (GR) to a new cosmological model of a conscious universe suggests a nonlocal conscious process (NLCP) is associated with the unified field. The NLCP is coupled to a nonlocal ‘noetic space’ that confines a fundamental singularity of individual noumenal consciousness. Singularities of elemental intelligence interact with the brain through a quantum of action autopoietically regulating the psychosphere (defined as the hyperstructure of an individual’s consciousness).

This model is called noetic field theory: the quantization of mind (NFT). Based on the putative concept of the brain as a naturally occurring quantum computer; according to the tenets of NFT, the extracellular confinement of natural intelligence provides a viable cosmology for constructing a conscious quantum computer (CQC).

Two types of optical quantum conscious computer are described as the most viable platform for a CQC:

- A. Optical DNA oligomer computers and
- B. Optical multimesh hypercube core with holographic interconnects utilizing heteroseric crystals.

This type of CQC would not be considered synonymous with artificial life, but more as a class of ‘zombie’ platform exhibiting evanescent natural intelligence during operation. Applications would include sensory bypass prosthesis and research platforms for studying the physical nature of consciousness and its interactions.

4 Transcendental States of Consciousness

12 KAPIL DEO PANDEY. **Transcendental states of consciousness.** C 27/239-1(ka), Jagatganj, Varanasi–221002, India.
<vimarsh@mailexcite.com>.

(1) Brahman, an identity revealed by the Upanisads, as the ultimate reality, is of the nature of Consciousness [1]. It is indeterminate without any kind of prediction [2]. Regarding its negative description [3] is the only description possible, since it is beyond the approach of the language-capacity [4]. Hence it can be ascribed as a completely transcendental reality.

(2) According to view of some other scholars, the Pure Consciousness that is common [5] to both the individual self (Jiva) and God (Isvara) is the thing reflected and the reflection of that very thing in the cosmic illusion (Maya) [6], in the Consciousness called God, while the reflection in different minds is the Consciousness called the individual self [7].

(3) To explain the process of world (Samsara), bondage and liberation, the ultimate reality has to be shown, not merely as sheer transcendence, but as immanence also. These two aspects are two facets of the same real existent identity. Upanisads, therefore, declare that Brahman, alone, exists and, as such, is transcendent [8].

(4) Considering this reality, empirically, it is essential to give an account of the levels of Consciousness according to the Vedanta system of Indian Philosophy, as follows.

Three states of Consciousness [9]:

(I) Vaisvanara/Virat, Consciousness associated with the aggregate of fourfold gross bodies and Visva, consciousness associated with the individual fourfold gross bodies. Their gross bodies are called alimentary sheath (Annamayakosa) and are said to be in the waking state (Jagrat).

(II) Sutratma/Hiranyagarbha/Prana: the consciousness associated with the aggregate of all the subtle bodies and Taijasa, the consciousness associated with individual subtle bodies. Their subtle bodies are called Pranamayakosa, Manomayakosa and Vijñanamayakosa and are also called the dream state, a state intermediate between the waking state and the state of dreamless sleep.

(III) Isvara, the Consciousness associated with the aggregate of ignorance on account of its being the illuminator of aggregate of ignorance, and Prajñā, the Consciousness associated with the individual ignorance. They are called associated with the casual body (Karanasarira) on account of its being cause of egoism and etc. and are, also, called the blissful sheath (Anandamayakosa) and further known as the state of dreamless sleep (Susupti).

(5) Pure Consciousness: There is an unlimited state of Consciousness which is the substratum of the Consciousness Isvara/Prajñā. This is called the fourth (Turiya caitanya), and Mandukyopanisad says: "Santam Sivamad-vaitam Manyate" (The wise conceive of as the fourth aspect). This is the transcendental state of consciousness [10].

(6) According to Vedanta [11], the perceptual knowledge is nothing but pure Consciousness as Brhadaranyakopnisad says: "Yat saksad aparoksad Brahma" [12]. Hence perception is two fold Jivasaksi (that due to the witness in the individual self) and Isvarasaksi (that due to the witness in God). The individual self (Jiva) is Consciousness limited by the mind (Antahkarana) and the witness in that (Jivasaksi) is the Consciousness that has antahkarana as its limiting adjunct. Similarly the witness in God (Isvara) is the Consciousness of which the cosmic illusion (Maya) is the limiting adjunct. The witness is the same as Brahman

(Transcendental Consciousness) [13].

(7) The Vedantic concept of absolute is the both immanent as well as transcendent simultaneously, Brahman is not the seer [14], since there is no other thing to be seen. Brahman is the utter 'Beyond', the 'Turiya'. It is not the Saksi (witness). Here, it will not be irrelevant to point out that neither the absolute of Bradley and Hegel [15] nor the Sunya [16] of Nagarjuna seem to be really transcendent.

References

- [1] Taittiriya Upanisad 2.1.1.
- [2] Svetasvatara Upanisad 6.11.
- [3] Chandogya Upanisad 7.25.2.
- [4] Taittiriya Upanisad 2.4.1.
- [5] Kathopanisad 1.3.12.
- [6] Svetasvatara Upanisad 4.10.
- [7] Brahmasutra with Sankarabhasya 3.2.18.
- [8] Ibid 1.1.24 and Chandogya Upanisad 3.13.7 & 7.25.2.
- [9] Vedantasara, Chapter II.
- [10] Ibid, Chapter II and Mandukyopanisad 7.
- [11] Vedantaparibhasa, Perception.
- [12] Brhadaranyakopnisad 3.4.1.
- [13] A.K. CHATTERJEE, A.K. & R.R. DRAVID. The concept of Saksi in advaita-vedanta.
- [14] Brhadaranyakopnisad 3.7.23.
- [15] TRIPATHI, R.K. Problems of Philosophy and Religion. BHATTACHARYA, K.C. Studies in Philosophy.
- [16] MURTHI, T.R.V. Central Philosophy of Buddhism.

[13] DAVID W. ORME-JOHNSON & KENNETH G. WALTON. **The myth of the relaxation response.** 191 Dalton Drive, Seagrove Beach, FL 32459, U.S.A. & Neurochemistry Laboratory, FM 1005, Maharishi University of Management, Fairfield, IA 52557, U.S.A. <davidoj@emeraldcoast.com> & <kwatson@mum.edu>. <<http://www.mum.edu>>.

Although relaxation and meditation techniques have been hypothesized to produce the so-called relaxation response, a review of the literature finds that the acute physiological changes that occur during most techniques are not significantly different from uninstructed rest, sitting eyes closed. Compared to rest, some techniques produce specific acute changes resulting from their specific methodologies, such as reduced muscle tension in muscle relaxation techniques, reduced respiration according to the well known orienting response in techniques that require focused attention, and reflexive entrainment of the heart rate with the breath for techniques that control respiration. The relaxation response was originally modeled on the changes produced by the Transcendental Meditation (TM) technique, but some changes that occur during the TM technique, such as increases in cardiac output, skin conductance, and plasma epinephrine, are in the opposite direction of the relaxation response. Many other changes associated with practice of this technique, such as increased cerebral blood

flow and EEG coherence, are unpredicted by the relaxation response. With regard to clinical outcomes, randomized clinical trials that controlled for expectation, placebo, and other design features, as well as meta-analyses and reviews of over 790 studies, provide strong evidence that different techniques are not equivalent and that they have specific effects [1]. For example, it appears that muscular disorders are best treated with muscularly oriented methods, while autonomic dysfunction such as hypertension and migraine headaches are more effectively treated with techniques that target the autonomic nervous system. The Transcendental Meditation technique appears to be the most effective treatment overall for a broad range of stress-related disorders, including hypertension, anxiety, substance abuse, and mental health. This is purportedly due to its effectiveness at promoting a state of "transcendental consciousness", a proposed fourth state of consciousness whose characteristics differ from those of ordinary sleeping, dreaming and waking [2, 3].

References

- [1] ORME-JOHNSON, D.W. & K.G. WALTON. 1998. All approaches to preventing or reversing effects of stress are not the same. *American Journal of Health Promotion* 12:297-299.
- [2] TRAVIS, F. & R.K. WALLACE. 1997. Autonomic patterns during respiratory suspensions: Possible markers of Transcendental Consciousness. *Psychophysiology* 34:39-46.
- [3] MASON, L.I., C.N. ALEXANDER, F.T. TRAVIS, G. MARSH, D.W. ORME-JOHNSON, J. GACKENBACH, D.C. MASON, M. RAINFORTH & K.G. WALTON. 1997. Electrophysiological correlates of higher states of consciousness during sleep in long-term practitioners of the Transcendental Meditation program. *Sleep* 20:102-110.

14 FREDERICK TRAVIS. **Physiological patterns during transcendental meditation practice as a basis for a neural model of transcending.** Psychology Department, Maharishi University of Management, Fairfield, IA 52557, U.S.A.
<ftravis@mum.edu>.

... our normal waking consciousness, rational consciousness as we call it, is but one special type of consciousness, whilst all about it, parted from it by the filmiest of screens, there lie potential forms of consciousness entirely different.

—James. 1961. *Varieties of Religious Experiences* P. 305

Over the last three decades, scientific research has probed into the physiological parameters of experiences during meditation practice. Because the functioning of the nervous system underlies and gives rise to specific experiences, physiological patterns (EEG, breath rate, skin conductance, heart rate) can index changes in internal states of conscious. Through these physiological "windows," experiences during meditation practice, which are distinctly different from ordinary waking, have been extensively investigated [1, 2]. Such meditation experiences may be examples of the potential forms of consciousness that James eluded to at the turn of this century.

The largest number of studies have been conducted on the Transcendental Meditation (TM) technique, as developed by

Maharishi Mahesh Yogi [3]. This presentation will present the latest findings on physiological patterns during TM practice, as the basis for discussing a neural model of potentially unique forms of consciousness during TM practice. I will present:

- 1) Six distinct EEG patterns of transcending, suggesting that EEG markers may be a secondary marker of these experiences;
 - 2) breath, skin conductance and heart rate markers that distinguish the deepest experiences during TM practice from other meditation experiences; and
 - 3) A comparison of EEG and autonomic markers during equal-length counterbalanced eyes-closed and TM sessions.
- These physiological patterns suggest that meditation involves frontal executive areas switching awareness to a state of restful alertness at the beginning of TM practice, and then maintaining that state through subcortical basal ganglia/thalamic/cortical loops.

References

- [1] SHAPIRO, D.H. & R.N. WALSH. 1984. *Meditation: Classic and contemporary perspectives*. Aldine Publishing. Hawthorne, NY.
- [2] TRAVIS, F.T. & R.K. WALLACE. 1997. Autonomic markers during respiratory suspensions: Possible markers of transcendental consciousness. *Psychophysiology* 34:39-46.
- [3] MAHARISHI MAHESH YOGI. 1969. *Maharishi Mahesh Yogi on the Bhagavad-Gita: A new translation and commentary*. Chapters 1-6. Penguin Books. Baltimore.

15 EDMOND CHOUINARD. **Floating consciousness—a coupling of mind and matter—with related experimental results on—"mental moving of flowers—a non-local mind fluctuation sensor"**. Metaphysical Measurements Inc, 485 River Avenue, Providence, RI 02908.
<edmeasure@aol.com>
<<http://members.aol.com/edmeasure>> for related book, "Holographic Mind".

The ultimate transcendental Consciousness of a Seer is considered to be one aspect of a foam labyrinth of striations formed by a spatially unique structure of quantum mechanical photon/boson particulants of vacuum state. The Seen is considered to evolve as a series of spatial Fourier transforms of the Seer structures, each uniquely defined by its own sets of low pass transformation filters. Such spatial transforms replicate from preceding states of evolution, resulting in an unfolding of creation. Explicated forms evolve from more implicated existences.

Both special and general relativity help explain the evolution of such transforms. The ever changing geometrical configuration of mass points (planets, black holes, stars, etc) in an expanding universe provides a rich assortment of naturally occurring light bending structures needed to generate the multitude of spatial Fourier transforms (giving rise to the time varying richness of awareness in the consciousness of Seers). The Seer consciousness is necessarily non-local since its constituents, the photon/boson particulants, consume no time in moving from point to point in configuration space, considered with respect to their own frame of reference. The mechanics of Seeing is thus defined by the ever on-going specialized and individualized (via select optical Fourier filters) photon/boson spatial transformations.

This triune Seer-Seeing-Seen structure is related to a number of principles (tattvas) developing from ancient Vedic systems of classification that are based upon sensory information mechanisms and upon various states of awareness that can arise, moving from abstract transcendental subtleties of mind to the most pragmatic dense forms of matter. This threefold aspect of consciousness may lie anywhere among the tattva principles, for instance, in analogy to a sounding of any three tones on a piano keyboard. The repeating spatial Fourier transforms thus so generate a glissando movement of differentiated states of consciousness as it moves downwards over the keyboard of tattvas. The Seer is always the highest tone and the Seen is always the lowest tone. The process of Seeing lies somewhere between. The Seer may thus float from the most rarefied to the most worldly pragmatic, but exactly what can be Seen at any time depends on the instantaneous location of the Seer.

This Model of Floating Consciousness is founded upon its ability to explain some most extraordinary personal experiences while remaining consistent with the world of physics. The Model finds collaboration with many popular forms of dualism which postulate the self existence of subtle mind, independent of a biological body, though normally tightly coupled to such a body. Such ideas can initially appear as fantasy but when they are experienced, even the "hard problem" of consciousness tends to disappear.

Experimental research on related mind-matter topics are presently being performed under the umbrella of this Floating Consciousness Model. Ongoing, developing experimental protocols simultaneously measure many dozens of physical, electro-optical, and biophysical variables on subjects, the interrelating environment, and targets (i.e., on the Seer, the process of Seeing, and the Seen), as a variety of mental and catalytic environmental conditions are invoked. High power plasma physics devices are also a part of this research as they have been involved, in a yet unknown way, in occasional and mysterious extraordinary visual sightings.

Tantalizing experimental results are now beginning to accrue and many photographic slides along with particulars on setups and protocols will be presented on some experiments entitled, "Mental Moving of Flowers—A Non-Local Mind Fluctuation Sensor". The results will be compared with the theoretical framework outlined here-in. Obviously, since this Floating Consciousness Model is so wide, is so inclusive, and has so many critical points that are yet mostly undeveloped, we will expect to find much room for updating the overall scheme. I can only ask for your help to further mature this risky venture.

16 RICHARD L. AMOROSO. **The utility of the transcendental state as a tool in theory formation.** The Noetic Institute, 120 Village Square # 49, Orinda, CA 94563-2502, U.S.A.
<ramoroso@hooked.net> preferred, and
<ramoroso@wenet.net>.

We have come full circle from the time of Galileo when the logic of philosophical deduction gave way to the empiri-

cal methods of science as the primary investigative tool of epistemology to a time when science and spirituality may finally have the impetus to merge and form a new methodology for comprehending the universe. In some arenas science has reached, at least to our current understanding, the limits of empiricism; and more pertinently the standard models of quantum theory and cosmology have begun to discover that we live in a conscious universe that contains an inherent teleological principle. For the first time in the evolution of human consciousness this sets the stage for a complete unified epistemology—One that integrates Science, Philosophy, and Theology; which Plato defined as Noetics, the only complete epistemology. Since there are 10,000 religions or spiritual paths in the world today most of which have conflicting teachings or dogmas; how could developing an 'empirical metaphysics' be possible? Since all theory formation has an inherent metaphysical component the solution seems rather straight forward. If the premise that we live in a conscious universe is true, and that this conscious universe contains an inherent teleology, in principle it should not matter which path is utilized to achieve the required 'transcendental state'. Whether through dance, fasting, meditation, chanting, prayer or peyote, it would be the resulting pure transcendental state that is of paramount importance. Investigators might also utilize the historical tracts and scriptures pertinent to their individual path to help guide questions posed to the universe through meditation and receive input through the mediation of the transcendental state. Examples from history might be the dream of the snake joining head to tail in the discovery of benzene, or perhaps more pertinent to our interest in the nature of consciousness, DesCartes claim of receiving a revelation from God pertaining to the distinction between mind and body. DesCartes 'vision' has remained controversial for 300 years and still not been tested. Science may now have progressed to the point where this is possible. The most important point to this discussion initially is that the information received through 'empirical metaphysics' for theory formation or otherwise need not interfere with science; because however a theory is formed it must or may still be experimentally tested. The great value of developing an integrative discipline of science and spirituality is that potentially 10's, 100's, or 1,000's of years could be saved, along with the resources expended on spurious paths that could alternatively be used to alleviate human suffering or maintain the environment etc. For example if the 'Big Bang' is wrong some of the best minds in astrophysics could have more efficiently expended millions of man hours over the last 75 years. Many of us would argue that an individual path would be best or the only one suited as a tool for empirical metaphysics; the most proficient test would be demonstrated in the fruits resulting from the labor. Historically Science and spirituality have remained somewhat mutually exclusive; where the former has been satisfied only by empirical evidence and the latter quietly submitting to faith. The possibility of integrating science, philosophy, and spirituality is discussed

along with potential methods and implications. The nature of the transcendental state and methods of achieving it and mutual verification withing research groups is also explored.

17 DEJAN RAKOVIĆ. **Transitional states of consciousness as a biophysical basis of transpersonal transcendental phenomena.** Faculty of Electrical Engineering, P.O.Box 35-54, 11120 Belgrade, Yugoslavia. <rakovic@buef31.etf.bg.ac.yu>.

It is pointed out (in the framework of the author's biophysical relativistic model of altered states of consciousness [1]) that transitional states of consciousness (as transpersonal biological macroscopic quantum gravitational tunneling phenomena) represent an excellent framework for understanding and mental control of transpersonal states of consciousness described in various religious and esoteric traditions of East and West during the history of civilization, implying their real biophysical nature. Transitional character of these states also demonstrates why these phenomena are short-lasting and poorly reproducible, and also why they are most easily mentally controlled shortly before a waking/sleeping transitional state. This might be also an explanation for extraordinary efficiency of prayer accomplished shortly before sleeping, which is recommended by all religious traditions, and its significance in mutual reprogramming of psychic conflicts (as a germ of the future interpersonal fights, as well as of potential psychosomatic and psychological disorders of the persons in conflict) during transpersonal interactions of the person in conflict in transitional states of the praying person (with direct mental addressing to the person conflicted with, or energetically more efficient indirect mental addressing via ionic abundant disembodied archetype structures from religious traditions).

At the same time, this offers a new insight in psychosomatic health and illness as essentially transpersonal phenomena, because a curing of psychosomatic disorder in only one person involved in some conflict without mutual reprogramming of the psychosomatic conflict is not final—as the non-reprogrammed conflict in the other person causes its (unconscious) transpersonal reinducement in the first one, in mentally-loaded transitional states of consciousness of the second one. All this suggests that deeper biophysical understanding of the nature of consciousness and transpersonal phenomena might soon give rise to scientific understanding and empirical verification of even fundamental philosophical/religious questions (like practical/spiritual significance of imperative moral behavior of every individual, extremely important for accelerated integration and spiritual evolution of personality)—and appearance of a new humanism, without meaningless and painful inter-personal, inter-ethnic, and inter-religious conflicts.

Keywords: theoretical mechanisms, biophysics, relativistic and quantum physics, brainwaves, neural networks, ionic structures, transcendental states of consciousness, transitional states of consciousness, altered states of consciousness, acupuncture system, psychosomatic healing, religion/esoteric transpersonal conflict reprogram-

ming, prayer, morals.

References

[1] RAKOVIĆ, D. 1995. Brainwaves, neural networks, and ionic structures: Biophysical model for altered states of consciousness. *In* D. Raković & Dj. Koruga, Eds. *Consciousness: Scientific Challenge of the 21st Century*. 291-316. ECPD. Belgrade.

RAKOVIĆ, D. 1997. Consciousness and quantum collapse: Biophysics versus relativity. *The Noetic J.* 1:1.

RAKOVIĆ, D. 1997. Prospects for conscious brain-like computers: Biophysical arguments. *Informatica* 21:507-516. Special Issue on Consciousness Studies.

18 SUHRIT K. DEY. **Scientific studies on consciousness in Indian Philosophy.** S.K. Dey, Professor of Mathematics, Eastern Illinois University, Charleston, IL 61920, U.S.A. <cfskd@eiu.edu>.

Indian philosophy is essentially a collection of many apparently distinct and sometimes contradictory thoughts and ideas. But except only one group of philosophers belonging to the materialistic views of Charvak, all essentially merge into one ultimate Reality—the Cosmic Consciousness. In this article a scientific analysis has been conducted on various thoughts in Indian Philosophy and it has been found that Vedanta Philosophy which categorically declares Cosmic Consciousness to be the very core element of this universe is the synopsis and a harmonious synthesis of all fragments of philosophical tenets of ancient India. Only Charvak's works have been completely rejected in Vedanta.

5 Practical Applications of Consciousness-Based Technologies

19 JOHN S. HAGELIN. **Natural Law and Practical Applications of Consciousness-Based Technologies in Government.** Institute of Science Technology and Public Policy. 1000 North 4th Street, Fairfield, IA 52557, U.S.A. <jhagelin@mum.edu>. <<http://www.mum.edu>>.

Scientifically proven technologies of consciousness can help governments overcome critical problems in the fields of education, health, crime, the economy, and the environment. Natural law—the laws of nature discovered by physical and biological sciences—upholds the lives of over eight million species in earth's complex ecosystems. Natural law governs the entire universe with absolute efficiency and economy—in accordance with the universal principle of least action. A growing body of scientific evidence suggests that most

national problems can be averted by government policies that effectively harness the limitless organizing power of natural law for the governance of society. These policies would require understanding and application of available technologies of consciousness.

Specifically, according to research, most social problems can be traced to the widespread "violation of natural law". "Violation of natural law" denotes actions which cause negative physiological, psychosocial or environmental repercussions—actions that fail to take advantage of the laws of nature governing health, the environment, or economic and social systems. It is therefore vitally important that governments prevent the widespread violation of natural law, and adopt policies that effectively apply the most up-to-date scientific knowledge of the laws of nature for the smooth and efficient administration of society.

In particular, governments need a technology to enable their citizens to live spontaneously in accord with natural laws. Scientific research over the past 30 years has verified that there are consciousness-based technologies which align individual thought and action with natural law, allowing individuals to live spontaneously in accord with natural law.

According to the Department of Health and Human Services, 70% of all disease is preventable—caused by an "epidemic of unhealthy habits", such as smoking, alcohol and drug abuse, and improper diet. Scientifically proven technologies of consciousness have been found to spontaneously reduce such health-afflicting behaviors, improve health, and reduce medical costs by 50% or more. Thus, a well-designed national health policy, incorporating these consciousness-based technologies, can better harness the laws of nature which uphold human health, and thereby improve health and dramatically decrease medical expenditures.

I will summarize material from a recent book (Hagelin J., *Manual for a Perfect Government*, 1998) detailing how specific technologies of consciousness align individual thought and action with natural law, and will discuss how such technologies enable governments to overcome critical national problems in the areas of education, health, crime, economy, and the environment.

20 JEREMY Z. FIELDS. **Consciousness-Based Medicine: Breaking the Vicious Cycle of Stress, Chronic Disease and Aging.** Center for Healthy Aging, Saint Joseph Hospital, Chicago, IL 60657, U.S.A. <jzfields@pop.interaccess.com>.

Maharishi Vedic Medicine (MVM) is a comprehensive system of natural medicine that holds "pure" or "transcendental" consciousness to be the most fundamental field of life [1]. This level of consciousness is different from the familiar states of waking, dreaming and sleeping. MVM recommends approaches such as the Transcendental Meditation technique and the TM-Sidhi program to increase access to this field of pure consciousness. As access to this field is increased through daily practice of the techniques, stabilized levels of consciousness higher than waking, dreaming and sleeping are attained. On the physiological level, the order and organizing power inherent in pure consciousness

mobilize the inner intelligence of the body, allowing defense and repair systems to remove the chronic effects of stress more effectively than other stress-reduction approaches [2]. With this reduction in "allostatic load" (the accumulated wear and tear resulting from effects of stress on adaptive mechanisms), and with the increase in adaptability to new stressful experiences, the mind/body moves in the direction of perfect health, and aging is increasingly disallowed.

This model, therefore, predicts that individuals who regularly practice the TM and TM-Sidhi programs will exhibit lower rates of physical and psychosocial stress, lower incidence and severity of disease, and less disability. Along with a healthier body, thinking and feeling should also become more refined, and positive health behaviors and happiness should increase. Finally, risk factors for disease-related mortality should decrease, and longevity and other indicators of successful aging should increase. Twenty-eight years of published data from controlled, clinical trials and from studies with other experimental designs strongly support the above model of consciousness-based medicine [1, 2]. These data will be reviewed and their far-reaching implications will be discussed.

References

- [1] ORME-JOHNSON D.W. & R.E. HERRON. 1997. An innovative approach to reducing medical care utilization and expenditures. *The American Journal of Managed Care* 3:135-144.
- [2] ORME-JOHNSON D.W. & K.G. WALTON. 1998. All approaches to preventing or reversing effects of stress are not the same. *American Journal of Health Promotion* 12:297-299.

21 KENNETH G. WALTON. **Consciousness-Based Technologies in Offender Rehabilitation: Psychological/Physiological Benefits and Reduced Recidivism.** Neurochemistry Laboratory, FM 1005, Maharishi University of Management, Fairfield, IA 52557, U.S.A. <kw Walton@mum.edu>. <<http://www.mum.edu>>.

Understanding consciousness has been enhanced in recent years by theories and practical approaches derived from ancient traditions. The most thoroughly researched of these approaches have been the technologies brought out by Maharishi Mahesh Yogi, modern exponent of the Vedic tradition in India. Although an important body of knowledge of consciousness has been kept alive in this tradition for over 8000 years, Maharishi holds this knowledge to be universal, that is, it is true and effective for all people, independent of their geographic and cultural origins.

Some key components of this knowledge are: (1) that there is an underlying, simplest state of consciousness which is the self-sufficient and self-referral basis of all experience; (2) through special techniques it is possible to experience this state in isolation, that is, awareness aware only of itself, completely without thoughts, emotions or sensory perceptions; and (3) through regular experience of this state, as is held to occur in the Transcendental Meditation technique, both mental and physical functioning are enhanced, and concrete benefits can be observed in all ar-

eas of life.

Offender rehabilitation is one of the areas in which this technology has been tested [1]. Controlled studies in maximum security prisoners have shown significant improvements in Loevinger's sentence completion test of ego or self development, as well as significant reductions in scores on a variety of tests of psychopathology, and a 50% reduction in recidivism rates. Because low scores on tests of self development and high scores on psychopathology tests have been associated with conduct disorders and delinquency, the positive changes in these factors due to consciousness based technologies, along with other stress reduction effects, may provide the basis for rehabilitation. These positive changes due to practice of the Transcendental Meditation technique were in sharp contrast to the minimal effects of comparison programs, including counseling, religious programs, and drug rehabilitation programs. Other studies outside of prison have indicated that the Transcendental Meditation program is an effective approach to reducing addictions, alcohol and drug use [2], which may further contribute to the reductions in criminal tendencies. The implications of these results for effective rehabilitation of criminals will be discussed.

References

- [1] ALEXANDER, C.N., K.G. WALTON, R.S. GOODMAN & D.W. ORME-JOHNSON, Eds. 1998. The Transcendental Meditation program in offender rehabilitation and crime prevention. *Journal of offender rehabilitation*. In press.
- [2] O'CONNELL, D.F. & C.N. ALEXANDER, Eds. 1994. Self-recovery: Treating addictions using Transcendental Meditation and Maharishi Ayur-Veda. *Alcoholism Treatment Quarterly* 11.

[22] KAM-TIM SO & DAVID W. ORME-JOHNSON. **Three randomized studies of the effects of the Transcendental Meditation technique on intelligence: Support for the theory of pure intelligence as a field underlying thought and action.** David Orme-Johnson, 191 Dalton Drive, Seagrove Beach, FL 32459, U.S.A.
<davidoj@emeraldcoast.com>.

In the Vedic tradition from which Maharishi Mahesh Yogi derived the Transcendental Meditation technique, consciousness is considered a field underlying individual intelligence and indeed all material expressions in nature [1]. In recent years, a variety of tests of this theory have been performed and the evidence generally has been supportive. In this study, we have attempted to determine if the results of introducing the technique at the level of high school produces effects that are sufficiently broad-based to support the theory of consciousness as an underlying field responsible for emotions, thoughts and actions.

We compared change in mental abilities over six or twelve months in 362 high school students in Taiwan who were randomly assigned to either the Transcendental Meditation (TM) technique, Napping or No Treatment, or with each of two self-selected groups, Contemplation Meditation and No Interest in meditation. Measures included

general intelligence, inspection time, constructive thinking, field independence, creativity, and anxiety. Multivariate Analysis of Variance on all measures showed that the TM groups increased significantly more than Contemplation Meditation ($F = 6.8, p < .000001$), Napping ($F = 3.19, p < .004$), two No-Treatment control groups ($F = 3.65, p < .002$), ($F = 5.48, p < .00003$) and the No Interest group ($F = 3.69, p < .001$). In the most stringent test, students randomly assigned to the TM technique improved more than randomly assigned controls on all individual measures. The results appear to support the wide range of other empirical, experiential, and theoretical evidence for a field of "pure intelligence" or "pure consciousness" underlying all thought and action. Results of this study and others may indicate this field is enlivened through practice of the Transcendental Meditation program and suggest it is this distinctive effect of the TM program that makes it more efficient at promoting a wide range of benefits in the area of psychological and physical health [2].

References

- [1] DILLBECK M.C., K.L. CAVANAUGH, T. GLENN, D.W. ORME-JOHNSON & V. MITTFELDELDT. 1987. Consciousness as a field: The Transcendental Meditation and TM-Sidhi program and changes in social indicators. *The Journal of Mind and Behavior* 8:67-104.
- [2] ORME-JOHNSON D.W. & K.G. WALTON. 1998. All approaches to preventing or reversing effects of stress are not the same. *American Journal of Health Promotion* 12:297-299.

[23] RAKOVIĆ¹, D., M. TOMAŠEVIĆ², E. JOVANOVIĆ^{1,3}, V. RADIVOJEVIĆ⁴, P. ŠUKOVIĆ¹, Ž. MARTINOVIĆ⁴, M. CAR⁵, D. RADENOVIĆ¹, Z. JOVANOVIĆ-IGNJATIĆ⁶ & L. ŠKARIĆ¹. **Electroencephalographic (EEG) correlates of some activities which may alter consciousness: The Transcendental Meditation technique, musicogenic states, microwave resonance relaxation, healer/heelee interaction, and alertness/drowsiness.**

¹Faculty of Electrical Engineering, P.O. Box 35-54, 11120 Belgrade. ²"Vinča" Institute of Nuclear Sciences, P.O. Box 522, 11001 Belgrade. ³Institute "M. Pupin", Computer Systems Department, P.O. Box 15, 11000 Belgrade. ⁴Institute for Mental Health, Department of Clinical Neurophysiology, Palmotićeveva 37, Belgrade. ⁵Institute for Biological Research, Center for Multidisciplinary Studies, 29. Nov. 142, Belgrade. ⁶Private Medical Practice "LAV", Vlade Zečevića 14, Belgrade, Yugoslavia.

<rakovic@buef31.etf.bg.ac.yu>.

A key problem of finding the most complete and useful theory of consciousness may revolve around how to empirically determine different styles or states of consciousness and how to incorporate these within a single paradigm. This was our motivation to start examination of EEG correlates of some activities or substates of consciousness which occur spontaneously or are induced artificially.

Our investigations demonstrated more or less characteristic features in 25 subjects practicing the Transcendental Meditation program (increased beta power in prefrontal region, increased theta power in left frontal and right temporal regions, increased alpha power in both temporal regions, and correlation between increased alpha power and decreased correlation dimension), 6 subjects with 4 types of spiritual music provided to induce musicogenic states (with significant changes in only 3 cases out of 24, where increased theta and alpha power was observed in only those subjects who have described their musical experiences as very pleasant), 28 subjects of relaxation induced by microwave resonance therapy applied to corresponding acupuncture points (with slightly decreased EEG power in all frequency bands, especially in the left central region, which can be ascribed to higher activation of the stimulated left circulatory part of the acupuncture system; it should be noted also that persons not previously subjected to this treatment responded stronger, presumably as a consequence of the more imbalanced acupuncture system), 5 healer/heelee, noncontact interactions (with increase in the maximum mean coherence of their EEG patterns in the alpha band observed only in short 4s time intervals), and 30 subjects for monitoring alertness/drowsiness level (with implemented automatic procedure of the neural network classifier to assess the correlation between EEG power spectrum fluctuations related to changes of vigilance level, demonstrating linear separability of the states of alert wakefulness and drowsy wakefulness, allowing very fast data processing and possible real time applications in clinical practice).

New technologies applied to the EEG may permit rapid and reproducible identification of different styles or states of consciousness. Such a tool might be useful in evaluating the effectiveness of different techniques for stress reduction and for altering expressions of consciousness.

[24] RACHEL SPIGEL GOODMAN. **International peace initiatives through Maharishi Mahesh Yogi's consciousness-based technology.** Psychology Department, DB1157, Maharishi University of Management, Fairfield, IA 52557, U.S.A.

<rgoodman@mum.edu>

<<http://www.mum.edu>>

This talk will introduce Maharishi Mahesh Yogi's Vedic approach to peace and its practical applications in the international arena. The central thesis of this approach is that consciousness itself is basic to thought and action, and thus basic to all the events which constitute conflict resolution and peace [15]. The view that consciousness is basic, even to material objects, also has been expressed by physicists, such as the father of quantum theory, Max Planck, who remarked, "I regard consciousness as fundamental. I regard matter as derivative from consciousness" [13].

According to Maharishi, pure consciousness is the state of least excitation of awareness in the mind; it is beyond the finest level of thinking activity and can be systematically located, experienced, and utilized through the Transcendental Meditation (TM) technique and the advanced

Transcendental Meditation-Sidhi program to strengthen thought and action [15] in the direction of greater coherence and more positive evolutionary tendencies for the individual and the society as a whole. Empirical research has indicated that practice of the TM and TM-Sidhi programs produces tangible results for the individual, including improved health [5, 16], enhanced intellectual capacity [4, 21], alleviation of stress-related conditions [18], and improvement in inter-personal relations [1].

The individual and the community in Maharishi's Vedic perspective have a reciprocal relationship: the consciousness of a society is a wholeness (known as collective consciousness) created by all the individuals living therein [10, 14]. The degree of peacefulness or coherence of any society depends on the level of coherence (or stress) generated by the individuals comprising that society [15]. In this context, coherence refers to the level of integration of the diverse concerns of the individual with the needs of the whole society. According to this perspective, development of consciousness in the individual is the essential means for creating the conditions conducive to coherence and progress in the society. The level of harmony in society, in turn, influences the individual.

Based on Maharishi's early prediction in the 1960's [17] and paralleled by principles of physics related to coherence in other natural systems, researchers have found that only one percent of a society's population practicing the TM technique (known as the "Maharishi Effect") or a group as small as the square root of 1% of a society's population practicing the TM and advanced TM-Sidhi program together (known as the "Extended Maharishi Effect") is necessary to influence society in the direction of greater coherence. Over forty research studies, including prospective intervention studies, have supported Maharishi's initial prediction in the areas of the economy [3], quality of life [2, 20, 22], crime rate [11, 12], support of the government by the constituency [8, 9], and international interactions [6, 19].

In this talk, I will give a brief summary of the research on Maharishi Effect initiatives related to conflict resolution and international relations. In particular I will describe studies which have empirically tested Maharishi's theoretical perspectives by utilizing time series analysis of events to elucidate the coherence-creating impact of large groups practicing the TM and TM-Sidhi programs on U.S.-Soviet interactions [6, 7] and U.S.-World relations [8, 9] during the Reagan administration.

References

- [1] ALEXANDER, C.N., G.C. SWANSON, M.V. RAINFORTH, T.W. CARLISLE & C.C. TODD. 1993. A prospective study on the Transcendental Meditation program in two occupational settings: Effects on stress-reduction, health, and employee development. *Anxiety, Stress, and Coping* 6:246-262.
- [2] ASSIMAKIS, P.D. & M.C. DILLBECK. 1995. Time series analysis of improved quality of life in Canada: Social change, collective consciousness, and the TM-Sidhi program. *Psychological Reports* 76:1171-1193.
- [3] CAVANAUGH, K.L. 1987. Time series analysis of U.S. and Canadian inflation and unemployment: A test of a field-theoretic hypothesis. In *Proceedings of the American Statistical Association, Business and Economics Statistics Section*. Pp. 799-904.
- [4] CHANDLER, H. M. 1991. Transcendental meditation

and awakening wisdom: A 10-year longitudinal study of self-development (Doctoral dissertation, Maharishi International University, 1991). Dissertation Abstracts International 51:5048B.

[5] DILLBECK, M.C. & D.W. ORME-JOHNSON. 1987. Physiological differences between Transcendental Meditation and rest. *American Psychologist* 42:879-881.

[6] GELDERLOOS, P., K.L. CAVANAUGH & J.L. DAVIES. 1990. The dynamics of US-Soviet relations, 1979-1986: Effects of reducing social stress through the Transcendental Meditation and TM-Sidhi program. *In Proceedings of the American Statistical Association*. Alexandria, VA.

[7] GELDERLOOS, P., M.J. FRID, P.H. GODDARD, X. XUE & S.A. LOLIGER. 1988. Creating world peace through the collective practice of the Maharishi Technology of the Unified Field: Improved U.S.-Soviet relations. *Social Science Perspectives Journal* 2:4:80-94.

[8] GOODMAN, R.S. 1997. The Maharishi Effect and government: Effects of a national demonstration project and a permanent group of Transcendental Meditation and TM-Sidhi program practitioners on success, public approval, and coherence in the Clinton, Reagan, and Bush Presidencies (Doctoral dissertation, Maharishi University of Management, 1997). Dissertation Abstracts International 58:06A:2385.

[9] GOODMAN, R.S., D.W. ORME-JOHNSON, M.V. RAINFORTH & D.H. GOODMAN. 1997. Transforming political institutions through individual and collective consciousness: The Maharishi Effect and government. *In Proceedings of the 1997 Annual Meeting of the American Political Science Association*. Washington, D.C.

[10] HAGELIN, J. S. 1987. Is consciousness the unified field? A field theorist's perspective. *Modern Science and Vedic Science* 1:29-87.

[11] HAGELIN, J. S., D. ORME-JOHNSON, M. RAINFORTH, K. CAVANAUGH & C. ALEXANDER. 1994. Results of the demonstration project to reduce violent crime and improve governmental effectiveness in Washington, D.C., June 7-July 30, 1993. Preliminary Technical Report ITR-94:1. Institute of Science, Technology and Public Policy, Maharishi University of Management. Fairfield, IA, U.S.A.

[12] HATCHARD, G.D., A.J. DEANS, K.L. CAVANAUGH & D.W. ORME-JOHNSON. 1996. The Maharishi Effect: A model for social improvement. Time series analysis of a phase transition to reduced crime in Merseyside metropolitan area. *Psychology, Crime and Law* 2:3:165-174.

[13] KLEIN, D.B. 1984. The concept of consciousness: A survey. University of Nebraska Press. Lincoln, NE, U.S.A.

[14] MAHARISHI MAHESH YOGI. 1986. Life supported by natural law. Maharishi International University Press. Fairfield, IA, U.S.A.

[15] MAHARISHI MAHESH YOGI. 1995. Maharishi's absolute theory of government: Automation in administration. Age of Enlightenment Printers. India.

[16] ORME-JOHNSON, D.W. 1987. Reduced health insurance utilization through the Transcendental Meditation program. *Psychosomatic Medicine* 49:493-507.

[17] ORME-JOHNSON, D.W. 1992. Theory and research on conflict resolution through the Maharishi Effect. *Modern Science and Vedic Science* 5:1-2:76-98.

[18] ORME-JOHNSON, D.W. 1994. Transcendental Meditation as an epidemiological approach to drug and alcohol abuse: Theory, research, and financial impact evaluation. *Alcoholism Treatment Quarterly* 2:1-2:119-168.

[19] ORME-JOHNSON, D.W., C.N. ALEXANDER, J.L. DAVIES, H.M. CHANDLER & W.E. LARIMORE. 1988. International peace project in the Middle East: The effect of the Maharishi technology of the unified field. *Journal of Conflict Resolution* 32:776-

812.

[20] ORME-JOHNSON, D.W., P. GELDERLOOS & M.C. DILLBECK. 1988. The effects of the Maharishi Technology of the Unified Field on the U.S. quality of life (1960-1984). *Social Science Perspectives Journal* 2:4:127-146.

[21] SO KAM-TIM. 1995. Testing and developing holistic intelligence in Chinese culture with Maharishi's Vedic Psychology: Three experimental replications using Transcendental Meditation. (Doctoral Dissertation, Maharishi International University, 1995). Dissertation Abstracts International 56:06A:2178.

[22] REEKS, D. 1990. Improved quality of life in Iowa through the Maharishi Effect. (Doctoral Dissertation, Maharishi International University, 1990). Dissertation Abstracts International 51:12:6155B.

[25] DANIEL MEYER-DINGKRAFE. Theatre as an aid to development of higher states of consciousness.

Department of Theatre, Film and Television Studies. University of Wales Aberystwyth, 1 Laura Place, Aberystwyth, Ceredigion SY23 2AU, Wales, U.K.

<dam@aber.ac.uk>.

<<http://www.aber.ac.uk/~drawww/>>.

Theatre involves consciousness on many levels: the performer's emotions, the author's process of writing, the effects of the play itself, or even the lighting, set, costume, or sound on performers and audience, and the reception process in the theatre. Thus, the theatre environment affords myriad opportunities by everyone involved to affect changes in the many types of experiences of the audience and the participants.

A model of consciousness contained in the Vedic tradition has been represented by the teachings of Maharishi Mahesh Yogi. This model proposes that an unbounded field of "pure" consciousness underlies all thought and events. When this field of pure or "transcendental" awareness is systematically experienced through technologies such as the Transcendental Meditation (TM) technique, the qualities of the field (i.e. holistic, dynamic, creative) are enlivened. Maharishi's conceptualisation of consciousness, in conjunction with the Indian treatise on drama and theatre, the *Natyashastra*, has proved valuable in understanding and explaining the specific relationship between consciousness and theatre.

Based on extensive scientific research on Vedic technologies brought out by Maharishi, I propose that the application of these technologies will improve the quality of life of all people involved in the theatre, including measurable improvements in performance-specific parameters in the theatre-related activity of playwrights, actors, producers and designers. I also propose that theatre itself, if following the guidelines provided in the *Natyashastra*, may be an effective addition to these Vedic technologies for improvement of quality of life.

For theatre itself to be useful as an aid to attaining higher states of consciousness, a thorough retranslation and reassessment of the *Natyashastra* from the perspective of Maharishi's teachings is needed. I hope that the hypotheses presented in this paper will stimulate colleagues interested in collaborating with me on further research in

this area.

6. Consciousness in Philosophy and Cognitive Science

26 IMANTS BARUSS. **Overview of consciousness research.** Department of Psychology, King's College University of Western Ontario, 266 Epworth Ave., London, Ontario, Canada N6A 2M3. <baruss@uwo.ca>.

An examination of definitions of consciousness by Baruss has revealed that they fall into four general categories: consciousness as the variable degree to which an entity can make discriminations and act in a goal-directed manner, metacognition demonstrated behaviourally, subjective experience characterized by intentionality and a subjective sense of existence. Historically the effort has been to operationalize subjective experience as metacognitive behaviour and, conversely, to infer subjective experience when an entity passes the Turing test for intelligence. Subjective aspects of consciousness pose problems for science to the extent that science is confined to that which is objectively observable. The resolution of first and third person perspectives is tied, more generally, to beliefs about consciousness and reality of consciousness researchers. Using a survey instrument designed for that purpose, in 1986 Baruss and Moore found a material-transcendent dimension underlying the beliefs about consciousness and reality of a sample of 334 academics and professionals with a potential interest in consciousness. This dimension was borne out in a subsequent survey of 212 participants at the academic conference *Toward a Science of Consciousness 1996 "Tucson II"*. Those tending toward the material pole emphasize emergent and informational aspects of consciousness, while those tending toward the transcendent pole emphasize the subjective nature and primacy of consciousness. Scores in a transcendent direction are associated with existential and spiritual interests and claims to having had extraordinary experiences such as out-of-body experiences. Thus, the personal experiences and beliefs of investigators play a role in their understanding of consciousness and create diversity in the research community. All three domains of discourse—the experiential, cognitive and physiological—must be engaged and knowledge integrated from both normal and altered states for an adequate theory of consciousness.

27 MIHALY LENART & ANA PASZTOR. **Pragmatic paradoxes and the hard problem of consciousness.** Department of Biomedical Engineering, University of Miami, 1251 Memorial Dr., MCA 219A, Miami, FL 33146, U.S.A. & School of Computer Science, Florida International University, University Park, Miami, FL

33199. Tel. (305)348-2019. Fax (305)348-3549. <pasztora@cs.fiu.edu>. <<http://www.cs.fiu.edu/scspage/professor/Pasztor.html>>.

The overall concern of our research is human communication. In this lecture we will zoom in on paradoxical communication.

There are three types of paradoxes [Watzlawick, 67], each corresponding to the three main areas of communication theory, namely syntax, semantics, and pragmatics. First, there are the so called antinomies, which arise in mathematical systems, like e.g., "the class of all classes which do not contain themselves." Then, there are semantic antinomies or paradoxical definitions, like e.g., "the least undefinable ordinal," and finally, paradoxes which arise in ongoing interactions between people and are called pragmatic paradoxes.

Our concern is this latter type of paradoxes, in particular so called double binds [Bateson, 72]. The ingredients for double binds are 1. a strong human relation of some kind of dependency, 2. a negative injunction, like e.g., "if you do not do so and so I will punish you," 3. a second injunction conflicting with the first at a more abstract level and enforced by punishments, and 4. a negative injunction or a context prohibiting the depending person(s) in the relationship to step outside the frame of communication and dissolve the paradox.

Pragmatic paradoxes, particularly double binds, can have a destructive effect on people's behaviour and their sanity by challenging their belief in the consistency of their universe. This leads to the extreme importance of fully understanding pragmatic paradoxes.

Both logico-mathematical and semantic antinomies are mostly well understood and have been the subject of extensive mathematical research. Not so pragmatic paradoxes. The difficulty in understanding them goes back to the problem of qualia or the alleged intrinsic first-person character of subjective experience—the famous *Hard Problem of Consciousness*, suggested by David Chalmers.

In this lecture we will present a model of first-person experience, i.e. of subjective experience, by decomposing it to those cognitive processes which serve as its building blocks [Damasio, 94; Pasztor, 98]. In order to function, these cognitive processes need the help of certain bodily and physiological processes for consolidation and expression. These physical reactions are important for their external observation and confirmation. The primary behavioral elements involved are: body posture, accessing cues, gestures, eye movements, and language patterns. After presenting each category in detail, we will show how they allow us to recognize the building blocks of subjective experience mentioned above, and how we can utilize them to understand the true nature of pragmatic paradoxes in general, and double binds in particular.

References

- [1] [Bateson, 72] BATESON, G. 1972. *Steps to an Ecology of Mind*. Balantine Books.
- [2] [Damasio, 94] DAMASIO, A.R. 1994. *Descartes' Error*. Grosset/Putnam.
- [3] [Pasztor, 98] PASZTOR, A. 1998. *Subjective Experi-*

ence divided and conquered. *Communication and Cognition* 31:1. In E. Myin, Ed. *Approaching Consciousness*. Part II. 73-102.

[4] [Watzlawick, 67] WATZLAWICK, P., J.H. BEAVIN & D.D. JACKSON. 1967. *Pragmatics of Human Communication*. W.W. Norton & Co.

[28] JEREMY HORNE. Information processing efficiency of logical operators. 15 Cooper Hill Court, Durham, NC 27713.
<jhorne1@cris.com>.

Underpinning the logic applied to structure arguments and display mathematical relationships is a philosophy explaining why it works. It can be argued that "Logic [is] the language of innate order in consciousness" (Horne 1997). "Logic ... is the theory of order," said James K. Feibleman (Feibleman, Assumptions 89). "It is the ... theory of abstract structures" (Feibleman, Assumptions 14). Stuart Kauffman in his *Origins of Order* (Kauffman 182-235) relies upon it to explicate his view that there is structure inherent in seemingly randomly generated logical operations. Wheeler, Thorne, Misner, and Piaget (as cited below) say that logic symbolizes inherent order in our universe. Binary logic is a reduction of expression to its simplest components. This paper focuses upon a fundamental component, the operator and its role in revealing order in ungrouped expressions.

I suggest that information processing efficiency can rank the order of logical operators in a parenthesis-free (ungrouped) expression. Efficiency is dependent upon how rapidly and accurately an operator manages the complexity of the other functions it processes. Similarly, efficient logical thinking often means how fast a person can run through logical alternatives to arrive at a correct conclusion.

Each of the 16 operators, or functions, in a two variable system is a self-maintaining (homeostatic) automaton in logical space. The homeostatic character of the function is displayed when outputs are fed back as inputs so as eventually to repeat, thus providing a basis for measuring how efficiently it processes other information in the system. Prioritization schemes in a parenthesis-free notation might be based on such an efficiency order to measure overall computational efficiency in the system. Before talking about the nature of operators in terms of computational efficiency, we need to examine, of course, the issue of prioritizing the operators.

[29] JAZZ RASOOL. The Nature of the mind-body gateway. 53 Broomgrove Gardens, Edgware, Middlesex HA8 5RH, U.K.
<jazras@hotmail.com>.
<<http://members.xoom.com/empower2000>>.
<<http://www.geocities.com/Area51/Cavern/5717/sum.html>>.

For many years interactions between the mind and body of

human beings have been limited in their description to the relationship between the psychological, neurological and immunological systems of the physical body. Even when the physics of mind-matter interactions has been discussed then these subjects in turn are not adequately covered. These investigations also have not taken into account the processes of physiological transformation brought about by non-local influences such as those attributed to absent healing. They also fail to adequately describe the connection between shifts in perspective within consciousness and subsequent physiological transformation.

It is the authors hypothesis that the physical representation, activity and evolution of emotion within the spacetime environment of an individual determines the nature of physiological correlates of mind. It is proposed that focused and coherent states of consciousness associated with states of emotion and conscious intention manifest in specific physiological forms through the fundamental formative mediums of matter that are specifically the quantum vacuum and the quantum gravitational field associated with it. It is proposed by the author that interactions between mind and body processes are facilitated by consciousness cooperating with quantum properties of gravity to transform fundamental aspects of space and time and the associated emergent physiological and material processes.

A model is presented that outlines the nature of interaction between consciousness and the local spacetime continuum and the biological structures or gateways that mediate such interactive processes. The gateways are defined according to structures that currently mediate the firing patterns of brain neurons. Concentrations of substances such as Gamma Amino-Butyric Acid (GABA) released by glial components within neuronal architectures are known to affect neuronal sensitivity to firing and control of such substances is central to the function of many anti-epileptic, anti-convulsant and anaesthetic drugs. The burst firing of cerebral neurons that accompanies epilepsy leads to transformations in consciousness and physiology—there are mind and body changes (change in 'aura', loss of consciousness and physiological convulsions). The release of GABA and other such substances can be influenced by a change in the electron flows along cytoskeletal microtubule protein architectures which have recently been implicated in playing important roles in the qualitative and quantitative aspects of conscious functioning. The author proposes that these microtubule structures and the water contained within them are the common ground upon which all mind-body transformations take place.

It is proposed by the author that that effective forms of alternative and complementary medical practices utilise specific forms of burst firing initiated by changes in the electrical properties of the microtubule architectures of not just neurons but all eukaryotic cells. Such alterations lead to simultaneous transformations in traits and properties of both mind and body.

The author proposes development of concepts and the initiation of research in this area of consciousness science that will facilitate more effective forms of medical treatment which add increased precision and quality to health-care systems that purport to maintain states of wellbeing within society.

30 MITJA PERUŠ. **Neural and quantum complex system dynamics as the background of consciousness.** National Institute of Chemistry, Hajdrihova 19, SI-1001 Ljubljana, Slovenia.
 <mitja.perus@uni-lj.si>
 <<http://kihp6.ki.si/~mitja/index1.html>>.

First, a summary of results of my research of neuro-quantum information-processing analogies will be presented shortly. Then, my Hopfield-neural-net-like quantum information processing “algorithm”, well tested by computer simulations, will be shown. It was developed as a relatively “natural” and biologically-plausible hypothesis.

Quantum Hopfield-like network model bases on the Feynman’s path-integral approach where a Hebb-like Green-function propagator is used as the kernel of the integral equation:

$$\Psi(\vec{r}_2, t_2) = \iint \left(\sum_{k=1}^P \psi^k(\vec{r}_1)^* \psi^k(\vec{r}_2) \right) \Psi(\vec{r}_1, t_1) d\vec{r}_1 dt_1 \quad (1)$$

In a metaphorical description, the overall interaction-web (“all quantum points for one, one point for all”) is combined with the Hebb-like “self-interference” projection operator (projects to memory subspaces), and the system is exposed to inputs. During learning phase, inputs become encoded as “network’s attractors”, or eigenfunctions ψ^k , respectively. During recall-phase, one of them ($\psi^{k=1}$) is selectively reconstructed if the new input (“key” Ψ') is similar to that memory-pattern $\psi^{k=1}$. Thus, we split the right side of Eq. (1) into memory subspaces and get the most similar and relevant memory-pattern ψ^1 out:

$$\left. \begin{aligned} & \left(\int \psi^1(\vec{r}_1)^* \Psi'(\vec{r}_1, t_1) d\vec{r}_1 \right) \psi^1(\vec{r}_2) + \\ & + \left(\int \psi^2(\vec{r}_1)^* \Psi'(\vec{r}_1, t_1) d\vec{r}_1 \right) \psi^2(\vec{r}_2) + \dots \\ & + \left(\int \psi^P(\vec{r}_1)^* \Psi'(\vec{r}_1, t_1) d\vec{r}_1 \right) \psi^P(\vec{r}_2) \doteq \\ & \doteq \psi^1(\vec{r}) + \text{noise} \end{aligned} \right\} \quad (2)$$

The quantum density matrix of the form

$$\rho(\vec{r}_1, \vec{r}_2) = \sum_{k=1}^P \sum_{h=1}^P \rho_{kh} \psi_k(\vec{r}_1)^* \psi_h(\vec{r}_2)$$

could also be used for description of quantum memory.

A discussion on biological implementation of the proposed information processing procedure will follow. It will be argued that neural nets are needed to “collapse” the wave-function, as in (2), and to discretize the quantum dynamics in order to enable the read-out of results of quantum “computation”. Probably, brain’s neural networks have a role of an interface between the classical environment and quantum-based consciousness, thus regulating the so-called “access consciousness” (informational aspect of phenomenal consciousness). However, as I will argue, for associative binding into conscious perceptions quantum dynamics is needed, but seems not yet to be enough to explain qualia.

References

For details on Hopfield neural nets see:

AMIT, D. 1989. *Modeling Brain Functions*. CUP. Cambridge.
 PERUŠ, M. 1995. *In* K. Sachs-Hombach, Ed. *Bilder im Geiste*. 183–194. Rodopi. Amsterdam/Atlanta.
 PERUŠ, M. & P. EČIMOVIČ. 1998. *Int. J. Applied Sci. & Computat.* 4:283–310.

For details on transformation of the Hopfield model to quantum information-processing formalism see:

PERUŠ, M. 1996. *Informatica* 20:173–183.
 PERUŠ, M. 1998. *Open Sys. & Inform. Dyn.* 5. In press.
 PERUŠ, M. 1998. *Advan. Synerget.* In press.
 PERUŠ, M. 1998. *Z. Angew. Math. & Mech.* In press.
 PERUŠ, M. 1997. *In* M. Gams et al., Eds. *Mind Versus Computer*. 156–170. IOS Press. Amsterdam.

For discussion of implications for consciousness studies see:

PERUŠ, M. 1997. *Informatica* 21:491–506.
 PERUŠ, M. 1998. *Informatica* 22:95–102.
 PERUŠ, M. 1997. *Noetic J.* 1:108–113.
 PERUŠ, M. 1997. *World Futures: J. Gen. Evol.* 51:95–110.
 <<http://goertzel.org/dynapsyc>>.
 <<http://www.iway.fr/sc/tribune/articles/peru1.htm>>.

7 Hand-out Topics of Consciousness

31 CYRUS F. NOURANI. **Intelligent Trees and Consciousness Science.** META AI South California, P.O. Box 278, Cardiff by the Sea, CA 92007, U.S.A.
 <Project_META AI@compuserve.com>.

1. Introduction. Multiagent Computing Intelligent Syntax, Visual Linguistics, Perception and Multiagent Cognition, are the bases for a tree intelligent Consciousness Science with computational models. The projects consist of Intelligent Trees, Intelligent Languages, Multiagent Computing on Multiboards, Visual-Computational Linguistics, Double Vision Computing, Multimedia Linguistics, Multiagent Cognition, and Multiagent Computational Logic. Intelligent Multimedia techniques and paradigms are defined. The computing techniques, the MIM deductive system and its model theory are presented in brief [1, 5].

Basic application areas we start with as examples are explicating visual perception and consciousness cognition, e.g. [7]. The second application area is based on AI planning. Reasoning and planning are applied to define scene dynamics consciousness based on scene descriptions and compatibility relations. The project allows us to predict scene dynamics. We apply our recent Intelligent Language paradigm and intelligent visual computing paradigms to define the IM multiagent multimedia computing as a thought paradigm. The phenomenological and philosophical issues are reviewed in brief and addressed in part by IM. Intelligent Trees and Intelligent Multimedia Models are defined as a basis for thought towards a Consciousness.

Science. A new computing area is defined by Artificial Intelligence principles for multimedia cognition. The principles defined are practical for artificial intelligence and its

applications to cognition and consciousness. What is new is a multiagent tree computing basis for consciousness science and cognition analogous to a Gestalt. Computing, cognitive, epistemological, metamathematical, and philosophical aspects are treated and reviewed in brief. It is further a start for Multimedia Linguistics.

Intelligent syntax languages Nourani[1, 2] are defined and their linguistic parsing theories outlined. A brief overview to context abstraction shows how context free and context sensitive properties might be defined by intelligent syntax. The preliminaries to a new computing logic termed MIM-Logic is defined with a brief model theory. The mathematical term for MIM is Morph Gentzen. The application areas are based on advanced Artificial Intelligence available techniques. There are at least a few areas to start with. Artificial Intelligence reasoning and planning can be applied to define content based on personality descriptions and compatibility relations being viewed. The project allows us to predict scene dynamics before viewing. Some of the applicable techniques, for example G-diagrams for models and AI applications have been invented and published by the author over the last decade. Multimedia thought models can be combined with intelligent trees and objects to stage and scene definition mind models.

2. Context. A preliminary overview to context abstraction and meta-contextual reasoning is presented from Nourani [2, 3]. Abstract computational linguistics with intelligent syntax, model theory and categories is presented in brief. Designated functions define agents, as in artificial intelligence agents, or represent languages with only abstract definition known at syntax. For example, a function F_i can be agent corresponding to a language L_i . L_i can in turn involve agent functions amongst its vocabulary. Thus context might be defined at L_i [2, 3]. Generic diagrams for models are defined as yet a second order lift from context. The techniques to be presented have allowed us to define a computational linguistics and model theory for intelligent languages. Models for the languages are defined by our techniques. The role of context in knowledge representation systems, particularly in the process of reasoning, is related to the diagram functions defining relevant world knowledge for a particular context. The relevant world functions can proliferate the axioms and the relevant sentences for reasoning for a context. A formal computable theory can be defined based on the functions defining computable models for a context.

3. Consciousness Awareness. Heidegger's phenomenology has been applied in our papers since 1993 [11] towards AI Modes for Thought and a computational epistemology for visual consciousness [6]. Heidegger had indicated the resources by which we conduct our day to day activities do not usually require consciousness awareness. Resources can present themselves in different modes of being: Available/Unavailable, and Concurrent with respect to day to day activities. There has been an intuition for years that consciousness is not or does not exist as an ordinary state or process itself, but it consists in the awareness of such state and processes. Locke epitomizes this intuition "Consciousness is the perception of what process is a man's own mind". Introspective consciousness is a perception-like awareness of the current states and activ-

ities in our own mind, e.g., sense perception., e.g. [12]. Our project presents morphing as subconscious activities by applying the IM Computing Logic as Way to Discovery [11]. The Morph Gentzen Computing Logic, The IM Multimedia computing techniques [9] have a computing logic counterpart. The basic principles are a mathematical logic where a Gentzen or natural deduction system is defined by taking multimedia objects coded by diagram functions.

We have put forth a basis for a mind computing model in our papers during the last decade for mental leaps [10, 12].

4. Intelligent Models. Intelligent syntax languages are defined and their linguistics parsing theories outlined. A computational logic for intelligent languages is presented in brief with a soundness and completeness theorem. A brief overview to context abstraction shows how context free and context sensitive properties might be defined.

The Gentzen system defined on MIM can be assigned an intelligent model theory. The mathematics is presented in [5].

4.1. Diagrammatic and Analogical Models. The IM Morphed Gentzen logic is a foundation for a consciousness logic which can be defined for specific reasoning models, for example analogical reasoning. Analogical reasoning rules are specific deductive designs which can be embedded by the logic. The multiagent cognition project had started in 1993 with the Double Vision Computing project [6]. Since there has been the 1994-95 Abstract Linguistic and Meta-Contextual Reasoning, Intelligent Multimedia, MIM logic, and its consequent paper on Consciousness Science.

4.2 Fundamental Mind as a Philosophical Question. Kant states consciousness operates in the dichotomy between thinking objects and the thought object. There are two components of knowledge: spontaneity and receptivity. In cognition Kant's spontaneity is not in the void. If acts of thought have objective bearing it is because they are filled with intuition of something that is given to the person, towards which the person is receptive. Whenever we know something through intuition something is given to us. Kant calls it sensibility. Intuition is never anything but sensuous [8]. The explanatory question is—can consciousness be explained by physical theories. The ontological question: Is consciousness itself physical. The Consciousness science formalized by our project since 1993 have applied positivist thoughts. However, we have taken Kant's sensibility with metamathematical precision up to infinitary models and logics, diagrammatic techniques, a Multiagent Cognitive Theory, dynamic epistemics, and the Morph Getnzen or MIM logic towards such science.

References

- [1] NOURANI, C.F. 1998. Intelligent trees, consciousness science, thought models, and intelligent discovery. Private communication.
- [2] NOURANI, C.F. 1996. Slalom Tree Computing. AI Communications. December 1996. IOS Press. Amsterdam.
- [3] NOURANI, C.F. 1998. Intelligent languages—A Preliminary syntactic theory. MFCS Grammatics. Prague.
- [4] GENTZEN, G. 1943. Beweisbarkeit und Unbeweisbarkeit

von Anfangsfällen der transfiniten Induktion in der reinen Zahlentheorie. *Math Ann* 119:140-161.

- [5] NOURANI, C.F. 1998. *MIM-Logik*. Summer Logic Colloquium. Prague.
- [6] NOURANI, C.F. 1995. *Double Vision Computing*. IAS-4. Intelligent Autonomous Systems. Karlsruhe, Germany.
- [7] CHALMERS, D.J. *The Conscious Mind*. 1996. In *Search of a Fundamental Theory*. Oxford University Press. Oxford, U.K.
- [8] KANT, I. 1990. *Critique of Pure Reason*. Translated by J.M.D Meiklejohn.
- [9] NOURANI, C.F. 1996. *Intelligent multimedia—A new computing technique and new AI paradigms*.
- [10] HOLYOAK, K.J. & P. THAGARD. *Mental Leaps. Analogy in Creative Thought*. MIT Press, Cambridge.
- [11] NOURANI, C.F. 1994. *Towards Computational Epistemology—A Forward*. Summer Logic Colloquium. Clairemont-Ferrand, France.
- [12] LINAS, R. & P.S. CHURCHLAND, Eds. 1996. *The Mind and Brain Continuum*. MIT Press.

32 TONY WRIGHT. **The expansion and degeneration of human consciousness.** c/o Steve Charter, Lockyers Farm, Peak Lane, Compton Dundon, Somerset, Somerset. TA11 6PE. England.
 <steve.charter@virgin.net>
 <<http://www.eclipse.co.uk/fresh/ecrbw>>

Whether in ancient texts, oral or mythic traditions virtually all cultures preserve a story that suggests humanity was once very different. Operating at a higher level of perception, long lived, free from disease, naked, forest dwelling, fruit eating, residing in 'paradise'. This golden age ends with catastrophe on a global scale—those that survive and subsequent generations increasingly suffer from the kind of difficulties we recognise as normal today. Some traditions talk of several ages of man each ending in disaster, with each new age bringing a further slide back to our animalistic origins.

A new theory suggests these traditions tell a very real story. Drawing on observations and anomalies in modern science, ancient wisdom and experimental techniques a picture emerges that is simple at its core yet virtually impossible to detect. A catch twenty two with an unexpected twist—at first sight perhaps difficult to assimilate with our apparently advanced technological, scientific culture.

In simple terms, the evidence suggests that we have suffered a devastating long term decline in brain function affecting the most delicate aspects of neural structure. The damage is primarily restricted to one half of the brain—however, unfortunately this is always the dominant half. This creates a severely distorted experience of who or what we are that is usually impossible to escape from. Basically the equipment we use to assess how we feel or what we experience, etc. is flawed and is unable to detect its own damage. As the immune system, hormone system, assimilation and just about everything else is effectively run by the brain a somewhat disturbing picture emerges.

The primary cause of this damage? Testosterone: one of

our own powerful growth regulating hormones produced by both males and females. Unleashed from millions of years of powerful inhibiting chemicals it now wreaks havoc on the new brain, distorting its unique structure and the biochemistry necessary for optimum function. The damage starts and is at its greatest in the earliest phase of development in the uterus and continues unabated throughout a life time, rapidly accelerating with puberty in males and menopause in females. Damage is greater in males as more testosterone is produced.

When our distant tree living ancestor switched from eating insects to fruit, flowers, leaves and shoots its own hormone regime was sufficiently affected by a powerful and unique cocktail of thousands of chemicals. This complex mixture had many properties; very rich in simple sugar (rocket fuel for the brain), chemicals that enhance neurotransmitter activity, relatively low in protein, high pH, etc. In particular such a cocktail is rich in chemicals that can inhibit steroids such as oestrogen and in particular testosterone. This is a very significant effect as these powerful steroids effectively dictate what parts of the DNA code are read—this in turn governs exactly how an organism develops and grows. The effects may have been minor at first, however what emerged was a subtle loosening on the restrictive effects of the steroids allowing a gradual somewhat chaotic tumour like proliferation of the relatively sensitive brain cells. In addition the same chemicals begin to extend the juvenile period prior to sexual maturity when those same steroids effect a range of physical and chemical changes. An extending period of juvenility is key to brain development as this is the period that the brain continues to rapidly grow. Humans are regarded as having an extremely long period of juvenility and retain some traits through to maturity.

What emerged over many millions of years was a new layer of brain with a different structure. As the brain grew larger and began to dominate the earlier more structured layers, it began to take effective control of the whole organism including the internal hormone system. In what was to become the direct ancestor of modern humans the new brain reached a critical mass in terms of size and ability to function sustainably and cohesively at highly energised levels. This may have been linked to an increasing specialisation in fruit eating—the most chemically optimum of foods for gradual, high quality brain growth and function.

At some point, perhaps two or three million years ago, the new level of brain function and activity began to induce key internal changes in the production of hormones—particularly the pineal gland—increasing the amount and period of melatonin production. This hormone has many powerful effects on longevity, immune function, juvenility etc. in part because it has a strongly inhibiting effect on testosterone.

There were now two layers of testosterone suppression, one a sustained external effect as part of an immensely complex bio-chemical cocktail that provided the raw materials for building the human brain/body for millions of years. The other, perhaps the key to many unique human features, an extremely powerful and increasing internal addition, entirely dependant on a new level of brain activity and underpinned by the sustained flood of external chemistry. These were the major elements that set human evo-

lution apart: primarily a brain/hormone led development that created its own increasingly powerful feedback loop. The new brain tripled in size in two million years.

Then not for the first or last time, ecological catastrophe on a global scale. The thin, solidified top layer of the earth pulled by centrifugal force, slipped a few degrees around the liquid centre. This creates rapid movement of climatic zones, massive inundation of low lying areas as oceans spill over their basins, earthquakes, volcanic activity and extreme wind speeds. The relatively fragile tropical forest ecosystems are virtually destroyed and with them the only source of year round fruit. The crucial chemical foundation in human evolution had become an Achilles heel. The external inhibition of testosterone and optimum chemistry for brain function is lost, which in turn reduces the production of melatonin; a double blow. Damage through increased exposure to testosterone is greatest in the uterus as the brain develops, producing successive generations with a brain/hormone regime where each is less able to maintain continued expansion. Although the damage is primarily restricted to one half of the brain its rapid loss of ability equates to increasing fear and a drive to dominate and control what can no longer be experienced or understood. The domination by the dysfunctional half of the brain merely accelerates the decline as it is much less able to maintain the necessary chemistry for continued development. Historically this may have been a slow war of attrition with the remaining function ever more dominated and suppressed by dysfunction. This imbalance severely distorted the unique level of consciousness and led eventually to physical symptoms such as handedness. The same basic patterns still occur very early on during a lifetime. In this way the period of exponential development rapidly spiralled into self perpetuating degeneration.

In a bid to halt or reverse the damage a range of ingenious techniques were developed (e.g. mystical, shamanic, etc.). They centred around replacing some of the missing bio-chemistry, replacing the key dysfunctional elements and various approaches to induce greater function in the less damaged half of the brain while reducing the dominance of the severely damaged half. A combination of these techniques may have slowed or halted the damage in the individual creating a sense of profound shift in experience, however without a clear understanding of the underlying causes and the need to prevent hormonal damage in the uterus a sustainable solution did not emerge. In more recent times (the last ten-fifteen thousand years) the relic and fragmented techniques that have survived can do little more than hint at something beyond understanding. Relatively profound and life altering, these experiences are now more an indicator of the degree of dysfunction that is considered 'normal', rather than significant experience of a lost state of being.

We are now dominated to varying degrees by cleverly disguised fear, a desperate need for familiarity/sense of safety and driven to great lengths to control any perceived threat to those damaged needs. A bleak picture indeed. But perhaps the first step to redemption is acknowledging the severity and nature of the problem. With the depth of specialist understanding in modern science brought to bear in an holistic approach the symptoms and underlying cause could be sustainably reversed within a generation.

If there is the slightest possibility that our self perception equipment is damaged then attempting to consider anything other than repairing the damage as an absolute priority would perhaps be the most telling symptom of the severity of the damage ... ?

Where does this leave us? Certainly, it leaves us with a need to undertake more, and possibly very challenging, research. Clues that support or at least do not seem to contradict some kind of severe dysfunction crop up in a whole range of places. One unlikely source being human sleep, where orthodox teaching suggests how important sufficient sleep is, yet the research into its function paints a less than clear picture. Looking for clarity is not helped by apparently wise people who advocate the use of short or extended periods without sleep in order to access greater function.

Who will assemble the gods unto thee, that thou mayest find the life which thou seekest? Come, do not sleep for six days and seven nights. (From the Epic of Gilgamesh.)

Despite (or in spite of) the general perception that plenty of regular sleep is necessary for normal function a range of clues suggest all may not be as it seems. From sleep walking to the deep insights that can emerge through dreams and the sleeping prophet (Edgar Cayce) many anomalies remain unexplained. Would the use of reduced or no sleep in many spiritual approaches make sense if there were a greatly distinct difference in the need for sleep between each half of the brain?

The mature, rapidly evolving new brain capable of increasingly highly energised states no longer had any need or a greatly reduced need for re-charging (sleep). What little, if any, that was required may have been taken by each half of the brain in turn in a similar way to dolphins. As dysfunction progressed there was a regression to a primitive need to shut down for increasingly long periods, as ability to maintain sustained activity was eroded. As this was primarily a problem for only one half of the brain then it is not difficult to see a means of exploiting this weakness. By starving the damaged half of its recharge time sufficiently long enough to run its batteries down its dysfunction increases along with its ability to suppress and maintain control. The opportunity to stimulate and re-engage the potential functional half while free from its jailer may be the basis of many powerful techniques now mostly lost.

Anything like fully cohesive function would only be possible if the missing evolutionary bio-chemistry had been replaced. Without necessary chemistry (e.g. diet) the primary experience is of increasing dysfunction and tiredness with a few hints of unusual experience. So, in reality it may be that 'normal' sleep keeps the dysfunctional half of the brain in control.

Call for Papers

Advances in the Theory and Practice of Natural Language Processing

Special Issue of Informatica 22 (1998) No. 4

Informatica, an International Journal for Computing and Informatics, announces the Call for Papers for the issue of an interdisciplinary volume dedicated to the theoretical and practical aspects of natural language (NL) analysis and generation.

This special issue is intended to be a forum for presenting, first of all, the theoretical ideas proved to be effective in the design of NL processing systems or promising to considerably extend the sphere of successful NLP applications.

TOPICS: Original papers are invited in all subareas and on all aspects of NLP, especially on:

1. The current state and advancements in the last five years in particular subfields of NLP.
2. Natural-language-like knowledge and meaning representation formal systems.
3. Formal approaches to describing conceptual structures of complicated real discourses (pertaining, e.g., to medicine, technology, law, business, etc.).
4. New logics for NLP.
5. Semantics-oriented methods of natural language analysis, conceptual information retrieval in textual data bases.
6. Computational lexical semantics, ontologies for NLP.
7. Understanding of metaphors and metonymy.
8. Anaphora resolution.
9. Generation of natural language discourses.
10. Parallel conceptual processing of natural language texts.

11. Intelligent text summarization.

12. New directions in NLP.

Informatica 22 (1998) No. 4, in an enlarged volume, is fixed as the special issue.

Time Table and Contacts

The deadline for the paper submission in four copies is July 30, 1998.

Printed-paper mail address:

Prof. A.P. Zeleznikar, Jozef Stefan Institute, Jamova c. 39, SI-1111 Ljubljana, Slovenia.

Correspondence (e-mail addresses):

— anton.p.zeleznikar@ijs.si

Prof. Anton P. Železnikar, Slovenia

— vaf@nw.math.msu.su

Prof. Vladimir A. Fomichov, Russia

— kitano@csl.sony.co.jp

Prof. Hiroaki Kitano, Japan

Format and Reviewing Process

As a rule, papers should not exceed 8,000 words (including figures and tables but excluding references. A full page figure should be counted as 500 words).

Ideally 5,000 words are desirable.

Each paper will be reviewed by at least two anonymous referees outside the author's country and by the appropriate editors.

In case a paper is accepted, its author (authors) will be asked to transform the manuscript into the Informatica \LaTeX style (available from ftp.arnes.si; directory: /magazines/informatica).

For more information about the Informatica and the Special Issue see

FTP: ftp.arnes.si

with anonymous login or

URL:

<http://turing.ijs.si/Mezi/informat.htm>.

Second Call for Papers Special Issue on

SPATIAL DATABASES

INFORMATICA

an International Journal of Computing and Informatics

<http://orca.st.usm.edu/informatica>

Edited by:

Maria A. Cobb
University of Southern Mississippi
Department of Computer Science & Statistics
Hattiesburg, MS 39406-5106
maria.cobb@usm.edu

Frederick E. Petry
Center for Intelligent and Knowledge-Based
Systems
Tulane University
New Orleans, LA 70118
petry@eecs.tulane.edu

A special issue of Informatica on the topic of spatial databases is planned for publication as Vol. 23, No. 4. Max Egenhofer (University of Maine, NCGIA) has agreed to write a special introductory article for the issue. Others who have tentatively agreed to contribute to the issue include Robert Laurini and Alan Saalfeld. We extend invitations for papers in all areas related to spatial databases, including, but not limited to the following:

- spatial data and relationship models
- distributed spatial databases
- visualization
- querying
- uncertainty management
- GIS
- spatio-temporal reasoning

Dates:

Full papers are due on December 1, 1998
Author notification set for March 31, 1999.

Final revisions of accepted papers are due September 30, 1999 (in LaTeX format).

Authors interested in submitting a paper for the issue should contact one of the guest editors listed above for submission details.

Machine Learning List

The Machine Learning List is moderated. Contributions should be relevant to the scientific study of machine learning. Mail contributions to ml@ics.uci.edu. Mail requests to be added or deleted to ml-request@ics.uci.edu. Back issues may be FTP'd from ics.uci.edu in `pub/ml-list/V<X>/<N>` or `N.Z` where X and N are the volume and number of the issue; ID: anonymous PASSWORD: <your mail address> URL: <http://www.ics.uci.edu/AI/ML/Machine-Learning.html>

CC AI

The journal for the integrated study of Artificial Intelligence, Cognitive Science and Applied Epistemology.

CC-AI publishes articles and book reviews relating to the evolving principles and techniques of Artificial Intelligence as enriched by research in such fields as mathematics, linguistics, logic, epistemology, the cognitive sciences and biology.

CC-AI is also concerned with development in the areas of hard- and software and their applications within AI.

Editorial Board and Subscriptions

CC-AI, Blandijnberg 2, B-9000 Ghent, Belgium.

Tel.: (32) (9) 264.39.52,

Telex RUGENT 12.754

Telefax: (32) (9) 264.41.97

e-mail: Carine.Vanbelleghem@RUG.AC.BE

Preliminary Call for Contributions

First Southern Symposium On Computation

December 4-5, 1998

University of Southern Mississippi
Hattiesburg, Mississippi

Organized by:

School of Mathematical Sciences of the University of Southern Mississippi in cooperation with the Department of Mathematics and Statistics and the Department of Computer Science of the Mississippi State University and the Department of Computer Science of Louisiana State University

The aim of the conference is to bring together researchers in ALL areas of computation and in an informal atmosphere attempt to develop links between its various threads.

Keynote Addressess (Preliminary List):

- G. Fairweather (Colorado School of Mines) - tentative
- S. Iengar (Louisiana State University)
- P. Lax (Courant Institute) - tentative
- F. Petry (Tulane University)
- S. Piacsek (Stennis Space Center)
- T. Skjellum (Mississippi State University)
- L. Welch (UT Arlington) - tentative
- J. Zhu (Mississippi State University)

Program Committee:

- A. Al-Dhelaan (King Saud University, Saudi Arabia)
- I. Banicescu (Mississippi State University, USA)
- I. Bar-On (Technion University, Israel)
- V. Berinde (North University of Baia Mare, Romania)
- A. Blazhievskiy (University of Podillia, Ukraine)
- C. Breshears (Waterways Experiment Station MSRC, Vicksburg, USA)
- I. Gladwell (Southern Methodist University, USA)
- M. Gams (Jozef Stefan Institute, Slovenia)
- J. Leszczynski (Jackson State University, USA)
- X. Li (Loyola University, New Orleans, USA)
- L. Mate (Technical University of Budapest, Hungary)
- N. Mastorakis (Hellenic Naval Academy, Greece)
- F. Mazzia (University of Bari, Italy)
- S-I. Morar (DeMontfort University, UK)
- S. Oppenheimer (Mississippi State University, USA)
- P. Schmidt (University of Akron, USA)
- I. Sivergina (Jekaterinburg Tech, Russia)
- P. Stpiczynski (UMCS, Poland)
- T. Taha (University of Georgia, USA)

- J. Tyler (Louisiana State University, USA)
- P. Yalamov (University of Rouse, Bulgaria)
- W. Zhang (Louisiana Tech, USA)

Organizing Committee:

M. Paprzycki (Chair) A. Ali, D. Ali, C. Burgess, M. Cobb, J. Ding, J. Kolibal, J. Lee, M. Mascagni, R. Necaie, R. Pandey, L. Perkins, W. Russell and R. Seyfarth.

We invite contributions covering all areas of computation. Special sessions are welcome. Contributions from graduate students are also invited.

Important Dates:

Special session proposals due: September 30, 1998 Extended abstracts (1 page) due: October 31, 1998 Acceptance: November 15, 1998 Papers for proceedings: November 30, 1998

Electronic proceedings will be published.

Information about the meeting will be updated on the conference WWW site to be located at: <http://pax.st.usm.edu/cmi/fsc98.html> To obtain more information about the meeting send e-mail to: fsc98@pax.st.usm.edu

Call For Papers

8th International Conference on Computer Analysis of Images and Patterns CAIP'99

Ljubljana, Slovenia, 1-3 September 1999

Conference Cochairs

Franc Solina, Aleš Leonardis
University of Ljubljana
Faculty of Comput. and Inform. Science
Tržaška 25,
1001 Ljubljana, Slovenia
Tel:386 61 1768 389,
Fax:386 61 1264 647,
E-mail:
franc.solina,ales.leonardis@fri.uni-lj.si

Program Committee

S. Ablameyko, Belarus
J. Arnspang, Denmark
R. Bajcsy, USA
I. Bajla, Slovakia
A. M. Bruckstein, Israel
V. Chernov, Russia
D. Chetverikov, Hungary
A. Del Bimbo, Italy
J. O. Eklundh, Sweden
V. Hlaváč, Czech Republic
J. Kittler, United Kingdom
R. Klette, New Zealand
W. Kropatsch, Austria
A. Leonardis, Slovenia
R. Mohr, France
M. Schlesinger, Ukraine
W. Skarbek, Poland
F. Solina, Slovenia
G. Sommer, Germany
L. Van Gool, Belgium
M. A. Viergever, Netherlands
S. W. Zucker, USA

Call for Papers

The CAIP conference is a traditional Central European Conference devoted to all aspects of computer vision, image analysis, pattern recognition and related fields.

The conference is sponsored by IAPR, Slovenian Pattern Recognition Society, IEEE Slovenia Section, Faculty of Computer and Information Science at the University of Ljubljana and Hermes SoftLab.

The scientific program of the conference will consist of plenary lectures by invited speakers, contributed papers presented in two parallel sessions and posters. The CAIP proceedings are published by Springer Verlag in the series Lecture Notes on Computer Science and will be distributed to the participants at the conference.

Scope of the Conference

- Image Analysis
- Computer Vision
- Pattern Recognition
- Medical Imaging
- Network Centric Vision
- Augmented Reality
- Image and Video Indexing
- Industrial Applications

Instructions to authors

Authors who wish to present a paper at the conference should send five copies of their paper to one of the two conference chairs marked CAIP'99. To enable double blind review there should be two title pages. The first with title, author's name, affiliation and address, telephone, fax and e-mail, abstract of 200 words and up to three keywords. The second title page should consist only of title, abstract and keywords. The papers excluding the title pages should not be longer than 10 pages. On a separate page the authors should answer the following three questions about their paper: (a) what is the original contribution?, (b) what is the most similar work?, (c) why is their work relevant to others?

A L^AT_EX template for the camera-ready version will be available on the conference home page. During the CAIP'99 review period the authors should not submit any related paper with essentially the same content to any other conference.

Deadline for submission of papers: 15 January 1999

Registration

Information on registration will be available on the conference home page.

Venue

The conference will be held at the Faculty of Computer and Information Science at the University of Ljubljana. Ljubljana is the capital of Slovenia. The city which is a lively mixture of Mediterranean and northern influences offers all amenities within short distance. Alpine resorts, the Adriatic coast and several natural spas are close to Ljubljana.

The conference homepage is:

<http://razor.fri.uni-lj.si/CAIP99>

Errata

Due to an error, in the paper "Advanced in Computer Assisted Image Interpretation" by W. Mess and C. Perneel, the bibliography was omitted. We sincerely apologise.

Preliminary Call For Papers

TIME-99:

Sixth International Workshop on Temporal Representation and Reasoning

Orlando, Florida, USA

May 1-2, 1999

The purpose of this workshop is to bring together active researchers in the area of temporal representation and reasoning in Artificial Intelligence. Previous workshops have been successful at bridging the gaps between theoretical and applied research in temporal reasoning. The workshop is planned as a two day event to immediately precede FLAIRS-99 (12th Annual Florida Artificial Intelligence Research Symposium, May 3-5). Visit the TIME99 webpage at <http://www.doc.mmu.ac.uk/STAFF/C.Dixon/time99/time99.html> or the FLAIRS-99 webpage at <http://erau.db.erau.edu/towhid/flairs99.html> for more details.

TIME-99 will be conducted as a combination of paper presentations, an extended poster session, and invited talks. Submission of high quality papers describing mature results or on-going work are invited for all areas of temporal representation and reasoning, including, but not limited to:

- temporal logics and ontologies
- temporal constraint reasoning
- temporal languages and architectures
- expressive power versus tractability
- belief and uncertainty in temporal knowledge
- temporal databases
- spatiotemporal reasoning and spatiotemporal databases
- temporal learning and discovery
- reasoning about actions and events
- time and nonmonotomism
- time in problem solving (e.g. diagnosis, scheduling,...)
- time in multiple agents, communication, and synchronization other applications

To maximize interaction among participants, the size of the workshop will be limited. Accepted papers will be invited for full presentation or a poster presentation. All submissions must be received by December 6, 1998. Prospective participants should submit a 6-8 page paper (indicating the selected areas), preferably electronically. To do so, send a postscript file (generated from a .dvi file) to one of the following email addresses

C.Dixon@doc.mmu.ac.uk

M.Fisher@doc.mmu.ac.uk

Alternatively, send 5 hard copies of your submission to:

Clare Dixon or Michael Fisher

TIME-99 Program Chairs

Department of Computing and Mathematics

Manchester Metropolitan University

Manchester M1 5GD

United Kingdom

Publication

All accepted papers will be published in the workshop proceedings. As well, a selected subset of the papers will be invited for inclusion (subject to refereeing) in a book or in a special issue of a journal.

Organizing Committee

James Allen General Chair

Clare Dixon PC Co-Chair

Michael Fisher PC Co-Chair

Howard Hamilton Treasurer/FLAIRS Liaison

Scott Goodwin Organizing Committee Member

Program Committee (Preliminary List)

Frank Anger National Science Foundation, USA

Mark Boddy Honeywell Systems and Research Centre, USA

Iliano Cervesato Stanford University, USA

Luca Chittaro Universita' di Udine, Italy

Jan Chomicki Monmouth University, USA

Philippe Dague Universite Paris-Nord, France

Mark Denecker Katholieke Universiteit Leuven, Belgium

Clare Dixon Manchester Metropolitan University, UK

Marcelo Finger Universidade de Sao Paulo, Brasil

Michael Fisher Manchester Metropolitan University, UK

Dov Gabbay Kings College, London, UK

Scott Goodwin University of Regina, Canada

Hans Guesgen University of Auckland, NZ

Peter Haddawy University of Wisconsin-Milwaukee, USA

Patrick Hayes University of West Florida, USA

Lina Khatib Florida Institute of Technology, USA

Peter Ladkin Universitaet Bielefeld, Germany
Gerard Ligozat Univerite Paris-Sud, France
Ron van der Meyden University of Technology,
Sydney, Australia
Angelo Montanari Universita' di Udine, Italy
Bernhard Nebel Albert-Ludwigs-Universitaet
Freiburg, Germany
Han Reichgelt University of the West Indies, Jamaica
Mark Reynolds Murdoch University, Perth, Australia
Abdul Sattar Griffith University, Australia
Yuval Shahar Stanford University, USA
Paolo Terenziani University of Torino, Italy
Andre Trudel Acadia University, Canada
Thierry Vidal National Engineering School of Tarbes
(ENIT), France
Lluis Vila Technical University of Catalonia, Spain

Summary Of Important Dates

December 6, 1998 Submission deadline May 1-2, 1999
TIME-99 Workshop May 3-5, 1999 FLAIRS-99 Con-
ference

[Dates for notification of acceptance and deadline for
camera ready copy to be decided but will be around
mid February and mid March respectively.]

THE MINISTRY OF SCIENCE AND TECHNOLOGY OF THE REPUBLIC OF SLOVENIA

Address: Slovenska 50, 1000 Ljubljana, Tel.: +386 61 1311 107, Fax: +386 61 1324 140.

WWW: <http://www.mzt.si>

Minister: Lojze Marinček, Ph.D.

The Ministry also includes:

The Standards and Metrology Institute of the Republic of Slovenia

Address: Kotnikova 6, 61000 Ljubljana, Tel.: +386 61 1312 322, Fax: +386 61 314 882.

Slovenian Intellectual Property Office

Address: Kotnikova 6, 61000 Ljubljana, Tel.: +386 61 1312 322, Fax: +386 61 318 983.

Office of the Slovenian National Commission for UNESCO

Address: Slovenska 50, 1000 Ljubljana, Tel.: +386 61 1311 107, Fax: +386 61 302 951.

Scientific, Research and Development Potential:

The Ministry of Science and Technology is responsible for the R&D policy in Slovenia, and for controlling the government R&D budget in compliance with the National Research Program and Law on Research Activities in Slovenia. The Ministry finances or co-finance research projects through public bidding, while it directly finance some fixed cost of the national research institutes.

According to the statistics, based on OECD (Frascati) standards, national expenditures on R&D raised from 1,6 % of GDP in 1994 to 1,71 % in 1995. Table 2 shows an income of R&D organisation in million USD.

Objectives of R&D policy in Slovenia:

- maintaining the high level and quality of scientific technological research activities;
- stimulation and support to collaboration between research organisations and business, public, and other sectors;

Total investments in R&D (% of GDP)	1,71
Number of R&D Organisations	297
Total number of employees in R&D	12.416
Number of researchers	6.094
Number of Ph.D.	2.155
Number of M.Sc.	1.527

Table 1: Some R&D indicators for 1995

	Ph.D.			M.Sc.		
	1993	1994	1995	1993	1994	1995
Bus. Ent.	51	93	102	196	327	330
Gov. Inst.	482	574	568	395	471	463
Priv. np Org.	10	14	24	12	25	23
High. Edu.	1022	1307	1461	426	772	711
TOTAL	1565	1988	2155	1029	1595	1527

Table 2: Number of employees with Ph.D. and M.Sc.

- stimulating and supporting of scientific and research disciplines that are relevant to Slovenian national authenticity;
- co-financing and tax exemption to enterprises engaged in technical development and other applied research projects;
- support to human resources development with emphasis on young researchers; involvement in international research and development projects;
- transfer of knowledge, technology and research achievements into all spheres of Slovenian society.

Table source: Slovene Statistical Office.

	Basic Research		Applied Research		Exp. Devel.		Total	
	1994	1995	1994	1995	1994	1995	1994	1995
Business Enterprises	6,6	9,7	48,8	62,4	45,8	49,6	101,3	121,7
Government Institutes	22,4	18,6	13,7	14,3	9,9	6,7	46,1	39,6
Private non-profit Organisations	0,3	0,7	0,9	0,8	0,2	0,2	1,4	1,7
Higher Education	17,4	24,4	13,7	17,4	8,0	5,7	39,1	47,5
TOTAL	46,9	53,4	77,1	94,9	63,9	62,2	187,9	210,5

Table 3: Incomes of R&D organisations by sectors in 1995 (in million USD)

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 post-graduates, 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^ovnia). The capital today is considered a crossroad between East, West and Mediterranean 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, 61000 Ljubljana, Slovenia
Tel.:+386 61 1773 900, Fax.:+386 61 219 385
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 WWW: turing.ijs.si/Mezi/informatica.htm

Referees:

Witold Abramowicz, David Abramson, Kenneth Aizawa, Suad Alagić, Alan Aliu, Richard Amoroso, John Anderson, Hans-Jurgen Appelrath, Grzegorz Bartoszewicz, Catriel Beeri, Daniel Beech, Fevzi Belli, Istvan Berkeley, Azer Bestavros, Balaji Bharadwaj, Jacek Blazewicz, Laszlo Boeszoermyenyi, Damjan Bojadžijev, Jeff Bone, Ivan Bratko, Jerzy Brzezinski, Marian Bubak, Leslie Burkholder, Frada Burstein, Wojciech Buszkowski, Netiva Caftori, Jason Ceddia, Ryszard Choras, Wojciech Cellary, Wojciech Chybowski, Andrzej Ciepielewski, Vic Ciesielski, David Cliff, Travis Craig, Noel Craske, Tadeusz Czachorski, Milan Češka, Andrej Dobnikar, Sait Dogru, Georg Dorfner, Ludoslaw Drelichowski, Matija Drobnič, Maciej Drozdowski, Marek Druzdzel, Jozo Dujmović, Pavol Ďuriš, Hesham El-Rewini, Pierre Flener, Wojciech Fliegner, Terrence Forgarty, Hans Fraaije, Hugo de Garis, Eugeniusz Gatnar, James Geller, Michael Georgiopolus, Jan Goliński, Janusz Gorski, Georg Gottlob, David Green, Herbert Groiss, Inman Harvey, Elke Hochmueller, Rod Howell, Tomáš Hruška, Alexey Ippa, Ryszard Jakubowski, Piotr Jędrzejowicz, Eric Johnson, Polina Jordanova, Djani Juričić, Sabhash Kak, Li-Shan Kang, Roland Kaschek, Jan Kniat, Stavros Kokkotos, Kevin Korb, Gilad Koren, Henryk Krawczyk, Ben Kroese, Zbyszko Krolikowski, Benjamin Kuipers, Matjaž Kukar, Aarre Laakso, Phil Laplante, Bud Lawson, Ulrike Leopold-Wildburger, Joseph Y-T. Leung, Alexander Linkevich, Raymond Lister, Doug Locke, Peter Lockeman, Matija Lokar, Jason Lowder, Andrzej Małachowski, Peter Mercer, Andrzej Marciniak, Witold Marciszewski, Vladimir Marik, Jacek Martinek, Tomasz Maruszewski, Florian Matthes, Timothy Menzies, Dieter Merkl, Zbigniew Michalewicz, Roland Mittermeir, Madhav Moganti, Angelo Montanari, Tadeusz Morzy, Daniel Mossé, John Mueller, Hari Narayanan, Elzbieta Niedzielska, Marian Niedźwiedziński, Jaroslav Nieplocha, Jerzy Nogiec, Stefano Nolfi, Franc Novak, Antoni Nowakowski, Adam Nowicki, Tadeusz Nowicki, Hubert Österle, Wojciech Olejniczak, Jerzy Olszewski, Cherry Owen, Mieczysław Owoc, Tadeusz Pankowski, Mitja Peruš, Warren Persons, Stephen Pike, Niki Pissinou, Ullin Place, Gustav Pomberger, James Pomykalski, Gary Preckshot, Dejan Raković, Cveta Razdevšek Pučko, Ke Qiu, Michael Quinn, Gerald Quirchmayer, Luc de Raedt, Ewaryst Rafajłowicz, Sita Ramakrishnan, Wolf Rauch, Peter Rechenberg, Felix Redmill, David Robertson, Marko Robnik, Ingrid Russel, A.S.M. Sajeev, Bo Sanden, Vivek Sarin, Iztok Savnik, Walter Schempp, Wolfgang Schreiner, Guenter Schmidt, Heinz Schmidt, Denis Sever, William Spears, Hartmut Stadtler, Janusz Stokłosa, Przemysław Stpicznyński, Andrej Stritar, Maciej Stroinski, Tomasz Szmuc, Zdzisław Szyjewski, Jure Šilc, Metod Škarja, Jiří Šlechta, Zahir Tari, Jurij Tasič, Piotr Teczynski, Stephanie Teufel, Ken Tindell, A Min Tjoa, Wiesław Traczyk, Roman Trobec, Marek Tudruj, Andrej Ule, Amjad Umar, Andrzej Urbanski, Marko Uršič, Tadeusz Usowicz, Elisabeth Valentine, Kanonkluk Vanapipat, Alexander P. Vazhenin, Zygmunt Vetulani, Olivier de Vel, John Weckert, Gerhard Widmer, Stefan Wrobel, Stanisław Wrycza, Janusz Zalewski, Damir Zazula, Yanchun Zhang, Robert Zorc

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.

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 five 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).

QUESTIONNAIRE

Send Informatica free of charge

Yes, we subscribe

Please, complete the order form and send it to Dr. Rudi Murn, Informatica, Institut Jožef Stefan, Jamova 39, 61111 Ljubljana, Slovenia.

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:

.....

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
E-mail: anton.p.zeleznikar@ijs.si
WWW: <http://lea.hamradio.si/~s51em/>

Executive Associate Editor (Contact Person)

Mätjaz Gams, Jožef Stefan Institute
Jamova 39, 61000 Ljubljana, Slovenia
Phone: +386 61 1773 900, Fax: +386 61 219 385
E-mail: matjaz.gams@ijs.si
WWW: <http://www2.ijs.si/~mezi/matjaz.html>

Executive Associate Editor (Technical Editor)

Rudi Murn, Jožef Stefan Institute

Publishing Council:

Tomaž Banovec, Ciril Baškovič,
Andrej Jerman-Blažič, Jožko Čuk,
Jernej Virant

Board of Advisors:

Ivan Bratko, Marko Jagodič,
Tomaž Pisanski, Stanko Strmčnik

Editorial Board

Suad Alagić (Bosnia and Herzegovina)
Shuo Bai (China)
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 (Austria)
Ante Lauc (Croatia)
Jean-Pierre Laurent (France)
Jadran Lenarčič (Slovenia)
Ramon L. de Mantaras (Spain)
Svetozar D. Margenov (Bulgaria)
Magoroh Maruyama (Japan)
Angelo Montanari (Italy)
Igor Mozetič (Austria)
Stephen Muggleton (UK)
Pavol Návrat (Slovakia)
Jerzy R. Nawrocki (Poland)
Marcin Paprzycki (USA)
Oliver Popov (Macedonia)
Karl H. Pribram (USA)
Luc De Raedt (Belgium)
Dejan Raković (Yugoslavia)
Jean Ramaekers (Belgium)
Paranandi Rao (India)
Wilhelm Rossak (USA)
Claude Sammut (Australia)
Walter Schempp (Germany)
Johannes Schwinn (Germany)
Branko Souček (Italy)
Oliviero Stock (Italy)
Petra Stoerig (Germany)
Jiří Šlechta (UK)
Gheorghe Tecuci (USA)
Robert Trappl (Austria)
Terry Winograd (USA)
Claes Wohlin (Sweden)
Stefan Wrobel (Germany)
Kindong Wu (Australia)

Informatica

An International Journal of Computing and Informatics

Contents:

Introduction		253
<hr/>		
Matrix Multiplication on Processor Arrays with Optical Buses	M. Middendorf	255
BPC Permutations on the OTIS-Hypercube Optoelectronic Computer	S. Sahni C.-F. Wang	263
<hr/>		
A Framework Supporting Specialized Electronic Library Construction	D.J. Helm J.W. Cogle Jr. R.J. D'Amore	271
A Study in the Use of Parallel Programming Technologies in Computer Tomography	E.G. Sukhov	281
Topological Informational Spaces	A.P. Železnikar	287
Control Mechanisms for Assuring Better IS Quality	M. Pivka	309
Authentic and Functional Intelligence	M. Radovan	319
LFA+: A Fast Chaining Algorithm for Rule-Based Systems	X. Wu G. Fang M. Gams	329
Forecasting from Low Quality Data with Applications in Weather Forecasting	B. Li J. Liu H. Dai	351
Reverse Engineering and Abstraction of Legacy Systems	M. Postema H.W. Schmidt	359
<hr/>		
Reports and Announcements		373