Volume 17 Number 2 August 1993 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Profiles: Jiří Šlechta
Memory Traces in the Brain
Multiple Perceptions
The Fifth Generation
Reports: AAAI'93, ML'93

The Slovene Society Informatika, Ljubljana, Slovenia

Informatica Volume 17 Number 2 August 1993 pp. 107-214

Informatica

An International Journal of Computing and Informatics

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, 61000 Ljubljana, Slovenia.

The subscription rate for 1993 (Volume 17) is
- DEM 50 for institutions,
- DEM 25 for individuals, and
- DEM 10 for students

plus the mail charge.

Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

LaTeX Technical Support: Borut Žnidar, DALCOM d.o.o.
Stegne 27, 61000 Ljubljana, Slovenia.

Lectorship: Fergus F. Smith, AMIDAS d.o.o., Cankarjevo nabrežje 11, Ljubljana, Slovenia.

Printed by Biro M, d.o.o., Žibertova 1, 61000 Ljubljana, Slovenia.

Orders for subscription may be placed by telephone or fax using any major credit card. Please call Mr. R. Murn, Department for Computer Science, Jožef Stefan Institute: Tel (+38) 61 159 199, Fax (+38) 61 158 058, or use the bank account number LB 50101-678-51841.

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 (Matjaž Gams)

The issuing of the Informatica journal is financially supported by the Ministry for Science and Technology, Slovenska 50, 61000 Ljubljana, Slovenia.

KNOWLEDGE—THE NEW INFORMATIONAL PARADIGM

The word *knowledge* has become a new informational paradigm, although it has come a long way from the traditional domains of artificial intelligence (e.g., intelligent data bases, expert systems, knowledge engineering, fifth generation computer systems, electronic dictionaries, knowledge archives, etc.). Why is it still a *new* paradigm and why is the hermeneutic way to its foundations (being, entity, essence) necessary? Does this mean that the recent excursion to the roots and application of knowledge was only a mechanical form of representation in a static data–base sense of the word.

The other form of knowledge understanding is still in the very early stage of research and seeks the informational ground for the framework of evolutionary–cybernetic philosophy. This philosophy seems to be becoming a region of as yet marginal agents in the framework of the European regionalism leaving, so far, the globally dominant R&D centers and producers of informational technology (U.S.A, Japan) uninvolved. As usual, traditional rationalism together with logical and computational tradition seems to form the main obstacle (metaphysical, hermeneutic circle) to a research breakdown and technological breakthrough. However, time already brings new challenges and today traditionally oriented fortresses of AI are beginning to change their paradigms even in the most notorious American technological research centers (M.I.T, Stanford University), where interdisciplinary teams of "computer scientists, linguists, logicians, philosophers, psychologists, and artificial intelligence researchers, seeking solutions to problems in their own disciplines, turned to one another for help" [see CSLI, Informatica 17 (1993), pp. 85–100]. This manner of interdisciplinary research in the realm of knowledge would be highly recommendable not only to the internationally linked European regionalists but also to research authorities in (small) European countries, where specific knowledge remains hidden from the 'eyes of the world'. A similar shift to a new research orientation can also be expected during the knowledge archive project (ICOT) in Japan.

In this issue of Informatica, the reader will find some of the mentioned reminiscences. For instance, what can be said about phenomenal connection between energy and information, on models (knowledge) of quantum processes impacting this interaction in the living brain (a profile of Jiří Šlechta and his article). A final overview of the FGCS Project (Koichi Furukawa) reveals the previously hidden intention of the project to become a technological basis (hardware with operating system) for future globally oriented knowledge archive systems and the industrial production of extremely technologically complex knowledge machines. Finally, within the 12th European Meeting on Cybernetics and Systems Research (EMCSR'94), the symposium *Cybernetic Principles of Knowledge Development* will deal with the problems of creation and evolution of knowledge—an international project and an internationally connected e-mail net of several hundred researchers across the globe—starting from a new informational–cybernetic paradigm of knowledge. The international journal Informatica can help you to foster new ideas simultaneously concerning traditional logic, computation, and informatics—your knowingly developing paradigm.

—*Anton P. Železnikar*, Editor–in–chief

PROFILES

Jiří Šlechta

Jiří Šlechta,[1] an Active Member of the *New York Academy of Sciences,* since February 1988, and *Who'sWho in the World 1993–94* (Marquis 11^{th} Edition) has been working mainly from his home in U.K., for the 15 last years. He has published 40 papers, presented 30 contributions at conferences, and 18 at international congresses. Recently he has been nominated also for the 2^{nd} edition of the *Who'sWho in the Science and Engineering.*

He received his RNDr at the Charles University, in Prague, in Czechoslovakia, in 1962. In 1964–65, he was a lecturer, and in 1965–69 a senior lecturer, at the Department of the Theoretical Physics of the Charles University.

In Autumn 1969, he emigrated to the United Kingdom where he has been living since. He became a British citizen, in 1984.

In U.K., he has worked as a Research Fellow at the School of Physics of the Warwick University in Coventry, 1969–71, as a Senior Research Associate in the School of Mathematics and Physics at the University of East Anglia (UEA), in 1971–74, as a Research Fellow in the Department of Physics of the University of Leeds, Leeds on 1976–77. Since 1978 he has been working from his Leeds home.

The main areas of his research have been (I) the theory of disordered systems and (II) the theory of the brain and society systems.

I. The Theory of Disordered Systems

(1) He discovered, formulated and developed a new theoretical method of the calculation of the density of states of disordered systems, the self–consistent continued–fraction (SCCF) method [J. Phys. C, 2047–57 (1977); Phys. Stat. Solidi (b) 120, 329–39 (1983)] which has been the most complete solution of the problem. It is suitable especially for the molecular electronics and biophysics (low–dimensional materials such as polymers, DNA, RNA, and biological compounds).

He has applied the SCCF method to:
a) the conductivity of these materials [J. Phys. C 12, 1819–34 (1979)], which shown the possibility of the dependence of the position of the Fermi level upon the structure of these materials, i.e. its tunability, by operating their structural properties, and which resulted into
b) an independent new derivation and justification of the Peierls–Fröhlich metal–nonmetal transitions (in the same work);
c) the theory of the Falicov–Ramirez–Kimball metal–nonmetal transitions and to the magnetic properties [Phys. Stat. Solidi (b) 127, 403–12 (1985)];
d) the Hubbard model [Phys. Stat. Solidi (b) 141, 457–55 (1987)], significant for the theoretical justification of the present experimental methods of decoding the genetic code.

(2) He studied the properties of the amorphous magnetic systems beyond the homogenous molecular field approximation [Phys. Stat. Solidi (b) 67, 595–607 (1975); 70, 531–6 (1975)] which lead to his real space theory of the anomalous Kondo effect [J. Non. Cryst. Solid 18, 137–48 (1975)] and how to utilize it for to gain the information about the higher spatial correlation than the pair ones [Jour. Mag. & Mag. Mater. 22, 1130–34 (1980)].

In all these works he pioneered his unique theoretical treatment of the disordered materials beyond the homogeneous molecular field approximation with the optimally detailed microcore.

II. The Theory of the Brain and Society (TBS)

He developed the TBS (economy) as systems with sparse (distributed) self–organizing (managing) set [Proceedings of the 12^{th} Int. Congress on Cybernetics, Namur, Belgium (1989)]. It derives the psychological responses of the higher animal (human being) from the properties of the molecular representation of the memory traces in the brain—a foundation of the molecular psychology.

He presented it for the first time at the 7^{th} International Congress of Cybernetics held in London, in 1987, and continued to do so at the International Congresses on Cybernetics in Namur, Belgium, in 1989 and 1992. At the later congress he was a chairman of three congress symposia.

He formulated the difference between the self–organization in living and inanimate systems. He has applied and generalized it in the theory of self–organizing systems with $T < 0$ which is a part of his theory of the brain to the noncontact thermodynamics of the society and economics (a theory of Mankind) [within his 21 works at the 13^{th} Int. Congress on Cybernetics, Namur, Belgium (1992), partly published in the Proceedings, recently, and partly to be published in Cybernetica].

In all these recent works he has been pioneering the dynamics of the distributed systems. A.P.Ž.

[1] In this issue of Informatica, for the first time an author's contribution is presented concerning the possibility of the quantum–informational phenomenology, discerning energy and information in the realm of the brain memory traces. In the paper which follows, the physicist Jiří Šlechta presents his quantum-statistical trial which is on the way to an uncommon route of understanding informational phenomena in brains.

ON A QUANTUM–STATISTICAL THEORY OF PAIR INTERACTION BETWEEN MEMORY TRACES IN THE BRAIN

Jiří Šlechta
Member of the New York Academy of Sciences
18 Lidgett Hill, Leeds 8, LS8 1PE, U.K.

Keywords: brain, information, informational contents (coupling, difference, quantum), memory traces, pair interaction, quantum statistical theory

Edited by: Anton. P. Železnikar
Received: April 26, 1993 **Revised:** May 31, 1993 **Accepted:** June 8, 1993

A quantum–statistical theory of pair interaction between memory traces (MTs) in the brain is presented and the basic formulas for its strength are derived.
It is shown that the interaction between two memory traces is proportional to the size of the contents of the (free) pieces of information (FPIs) exchanged between them and the number of such exchanges during the given period.
The Green function of the propagation of both MTs and the quanta of FPIs exchanged between two MTs, and their properties, are introduced and studied by means of an elementary Feynman technique.
It is shown for example, that the 'blown up' brain cells found in the brains of people suffering from schizophrenia may be caused by a resonance interaction with them within a ring of MTs (a laser of them) storing a piece of too-simple (crystal-like) information.

1 Introduction

In [1] there was formulated a set of the properties of memory traces in the brain (MTs) which is sufficiently general to incorporate all the specific mechanisms of their working proposed to date, including the processes in the membranes of neural cells, microtubuluses, vesicular mechanism, etc.

On the basis of this set, a general theory of the functioning of the brain has been developed in [2]–[9], spanning from this set of the properties of MTs to an impressive number of the elementary macroscopic responses of a (higher) animal, including human beings, in their relations with the changing external conditions in its environment. This can be seen fitting into the Pavlovian psychological system ([5]). Because of this, the theory represents a theoretical foundation of molecular psychology.

In [2], it was shown that the Hamiltonian of the system of memory traces can be written in the form

$$H = \sum_i \varepsilon_{0i}\, a_i^+ a_i + \frac{1}{2} \sum_{ij} I_{ij}\, a_i^+ a_j \qquad (1)$$

where ε_{0i} is the information quantum stored at position i and I_{ij} is the interaction between the memory traces at positions i and j, a_i^+ and a_i, the creation and annihilation respectively of the informational quantum $varepsilon_{0i}$ in the MT at position i.

In this work, we will show how to determine I_{ij} quantitatively. An explicit expression for the interaction I_{ij} will be derived.

2 Exchange Interaction I_{ij} between Two MTs

In [1] and [3], it was described how MTs are created by the senses to record information about the world around the living body.

Within such a process are linked the following two aspects:

a) a storage of the information related to such a record;

b) the physicochemical processes connected with it, which make such storage possible.

The link is not uniquely defined because storage of the same amount of information can be done in a number of different ways; the type of the stored information and the related chemistry may be, in general, only loosely related, although, once such a relation has been linked, it forms the basis of a unique MT of it (thus remembered). This is a special representation of the duality of the energy and information ([10]).

Though these two aspects are closely interrelated and mutually inseparable, it is useful to abstract them from each other and introduce their ideal separated images which are then, in their real coexistence, put into an interaction ([10] and Section 3).

Let us have n MTs which exchange between one other, pair-wise, pieces of information related to the same event ([1]).

Let us consider, at first, that there are two subsystems:

a) The system of the physical representation of the MTs, called 'free MTs', henceforth denoted as FMTs, the Hamiltonian of which is given by the expression

$$H_0 = \sum_i \varepsilon_{0i}\, a_i^+ a_i \qquad (2)$$

The first part of the equation (1), and

b) the pool of free carriers of the pieces of information (henceforth denoted as FPIs), for example, of the surroundings of an animal, which is then stored by the 'free MTs'

$$H_1 = \sum_k L_k\, b_k^+ b_k \qquad (3)$$

where b_k^+ and b_k are the creation and annihilation operators respectively of the individual free pieces of information being exchanged, and L_k is the information content of each individual piece of information, k is the index numbering them.

The FMTs (as with MTs) are realized by the molecules, their quantum states, and by the dynamic quantum properties of the autocatalytic metabolic cycle of the processes forming them ([1]), that is by the electronic configurations involved, and therefore due to the Pauli principle involved ([11]), they are fermions (e.g. the same piece of the FPI can be stored there in one copy, only).

On the other hand, FPIs, unlike FMTs, do not have a character of particles but quanta of a quantum field of boson type (e.g. plasmons). They do not need to conserve their number. The circumstances for more complex second quantization (like the fact that system MTs which hold, in a stationary situation, the knowledge of the once memorized FPI, by which property they conserve their number, may, on a longer time scale, nonconserve this property), known in their own right in, for example, molecular crystals ([12]), are beyond the scope of this introductory work. They are absorbed in FMTs, to form MTs. This is assumed to be a reversible process. There is no physical rule against their total destruction (being forgotten).

Therefore, the Hamiltonian of the interaction of the FMTs at i and j, via an exchange of the FPIs b_k will be assumed to have the form

$$H_{I_1} = \sum_{ijk} E_{ijk}\, a_i^+ a_j (b_k^+ + b_k) \qquad (4)$$

where E_{ijk} is the informational coupling connected with such an exchange. The hermiticity of the Hamiltonian H_{I1} requires $E_{ijk} = E_{jik}^*$. The structure of E_{ijk}, that is the detailed local conditions of the absorption of the FPIs by FMTs (such as memorizing the message carried by a particular plasmon by a given molecule) in each individual case, is beyond the scope of this work and is to be discussed elsewhere.

It resembles the Hamiltonian of the interaction between the particles, like electrons, or pseudoparticles, like excitons, on one side, and a quantum field of phonons, or photons, on the other side ([11]–[17]).

The total Hamiltonian of the system of MTs interacting with FPIs has the form

$$H = H_0 + H_1 + H_{I_1} \qquad (5)$$

In the x-representation, the Hamiltonian (4) can be written in the form ([13])

$$H_{I_1} = \int dx\, dy\, dz\, E(x,y,z)\, a^+(x)\, a(y)\, [b^+(z) + b(z)] \qquad (6)$$

Further discussion will focus on its simpler form

$$H_{I_1} = -\Gamma_0 \int dx\, a^+(x)\, a(x)\, (b^+(x) + b(x)) \quad (7)$$

where Γ_0 is the so-called elementary vertex which expresses the strength of the three-parameter interaction, constant in these parameters, otherwise. Then, according to [13] the effective potential energy between two MTs I_{ij} in the equation (1) can be expressed by the formula

$$I_{ij} = U = U_{12,34} - U_{12,43} \quad (8)$$

where

$$U_{12,34} = 2\Gamma_0^2 \int dx\, dx'\, a_4^+(x)\, a_2(x) * D_c(x,x')\, \tfrac{\omega_{42}-\omega_{31}}{2}\, a_3^+(x')\, a_1(x') \quad (9)$$

and

$$\omega_{mn} = (\varepsilon_{0m} - \varepsilon_{0n}) \quad (10)$$

is the relative informational difference between the FMTs at positions i, at the states ε_{0m}, and j, at the state ε_{0n}, communicated by the FPI exchanged between them, and the vertex function $D_c(x,x')$ is given by the formula

$$D_c(x,x') = -i\langle b^+(x), b(x')\rangle_{\mathrm{FPI}} \quad (11)$$

where $\langle\ \rangle_{\mathrm{FPI}}$ denotes averaging over the pool of FPIs; in the case of two MTs, this means to average over the exchanges of the quanta of a message during a given period.

The indices 1 and 2 number the incoming and outcoming flow into the position i repectively, while indices 3 and 4 number the same for position j.

The hermiticity $U_{12,34}$ given by equation (9) gives, for fermions, their symmetry properties in the form $U_{12,34} = -U_{12,43}$.

In this way, the task of determination of the quantitative expression for I_{ij} is fulfilled.

The quantity $D_c(x,x')$ has the meaning of the correlation function ([11]) between the pieces of the information of the given pool at x and x'.

From this, it can be seen that the interaction between two MTs is proportional to the size of the content of FPIs exchanged between two MTs and the number of their exchanges over a given period.

This expresses the fact that FMTs interact not only because of the consequences in the machine of the physical processes which realize them, but also because of the interrelation of FPIs stored in them (an elementary logic of it).

3 On the Duality of the Energy and Information

Let us structure further this framework of the workings of real memory traces to add more inside to what can be seen as 'true' quantum logic realized within the structure, by considering that each FMT, and FPI, carries two mutually independent aspects: an informational one and a physical one.

Let us have a system of physical elements, or particles, which form the FMTs. Each such element at the site i has its unique structure S_i, which stores both an energy quantum ε_{E_i} and the quantum information ε_{I_i} ([1]).

Let the sites i and j interact through informational exchange interaction V_{I_i,I_j} ([1], [2]) and physical interaction V_{E_i,E_j}, each determined by equation (8), either for the particles, or for the FPIs.

According to traditional physics, the Hamiltonian of the energy part of the containment of the system is ([10])

$$H_E = \sum_i \varepsilon_{E_i}\, c_i^+ c_i + \frac{1}{2} \sum_{ij} V_{E_i,E_j}\, c_i^+ c_j \quad (12)$$

where c_i^+ and c_i are the creation and annihilation, respectively, operators of the energy part of the structure at site i, ε_{E_i} is the energy quantum stored at position i, V_{E_i,E_j} is the energetical part of the interaction between positions i and j. It is given by a formula analogous to equation (8), its analogy for physical systems being ([13]).

According to [1], the Hamiltonian of the informational part of the containment of such a system can be written in the form

$$H_I = \sum_i \varepsilon_{I_i}\, a_i^+ a_i + \frac{1}{2} \sum_{ij} V_{I_i,I_j}\, a_i^+ a_j \quad (13)$$

where a_i^+ and a_j are the creation and annihilation operators, respectively, of the quanta of information at site i, V_{I_i,I_j} is the informational exchange interaction between positions i and j given by equation (8). Equation (13) is an analogous form of equation (1) expressed, here, in the more detailed twin formalism, simultaneously catering for both the energy and information part of the system ([10]).

The energy and informational parts of the structure S_i do not need to be related in a fixed

mutual interdependence. The same amount of information can be stored in a number of different structures with different amounts of stored energy related to that informational amount in them.

The same is true of the absorptions and emissions of the energy and information quanta relative to the informational transfer with which they are associated ([10]).

An extreme case is when the information stored at site i is read without destruction, or a modification of the structure S_i, that is also without releasing its physical energy (like reading this page)—a physical realization of the Eccles's mind ([18], [19]).

That is why the propagation and storage of energy and information can be studied, as two idealized cases, as two different idealized structures separately.

The remaining problem is to express quantitatively how these two (sub)systems, and their elementary processes, interact, within the duality of these two aspects of a structure.

Let us, at first, study the interaction of an informational quantum $\varepsilon_{I_{0i}}$ with the physical system defined by the Hamiltonian (12).

Its absorption changes the structure S of the system by adding to it a structural increment informationally equivalent to the informational contents of ε_{I_0} (generally, it need not be a unique solution of it). It may be connected with a change in the total energy of the system of MTs, but it does not need to. It can only change the topological structure of the system (the reversibility of such storage is a subproblem in its own right ([10])).

This can be formally expressed by the term

$$H_{EI} = \sum_i \frac{\partial H_E(S_{I_i}(\varepsilon_{I_i}))}{\partial \varepsilon_{I_i}} a_i^+ a_i \, c_i \qquad (14)$$

where S_{I_i} is the dependence of the structure S_i of the system upon its informational contents. Some informational increments can be realized equi-energetically.

On the other hand, an additional physical excitation ε_{E_0} of the total system also changed its structure and thus its informational contents. Therefore, the term expressing the interaction of the elementary physical excitations $\varepsilon_{E_{0i}}$ with the informational contents of the system is given in the form

$$H_{IE} = \sum_i \frac{\partial H_I(S_{E_i}(\varepsilon_{E_i}))}{\partial \varepsilon_{E_i}} c_i^+ c_i \, a_i \qquad (15)$$

where S_{E_i} is the energy dependence of the structure S_i. Some energy excitations do not need to add any informational change to the inner structure of the system.

The total Hamiltonian of the system is

$$H = H_I + H_E + H_0 + H_{EI} + H_{IE} \qquad (16)$$

where H_0 is the Hamiltonian of the ground state of the system—the unexcited part of the system above which are created the excited two subsystems ([14]).

This duality of energy and information is developed further in a generalized form in [10]. There it is shown that H given by equation (16) has a metric form

$$H \equiv \begin{pmatrix} H_I & H_{IE} \\ H_{EI} & H_E \end{pmatrix} + H_0 \qquad (17)$$

Generally, $H_{EI} <> H_{IE}$ expresses the irreversibility of the exchange of information between the basic systems H_I and H_E.

The framework presented above can also be applied to the interaction between the whole systems of MTs and the lasers ([1]) which record the different events. In that system they replace the FMTs.

Examples of such systems: the interaction between vesicles, an electromagnetic interaction between two systems of MTs, which can happen even at very long distances, between the 'lasers' of MTs of two different people (a case of telepathy between them ([3])), which however needs to be lined with the processes which do not change particles' content ([4]), only, not to trigger the motor centrum ([5]) on the basis of data not acquired from the living body's own sensory experience (the opposite is the cause of a kind of mental illnesses).

4 Propagators of MTs and Messages between Them (FPIs)

The propagator of an FPI between two MTs can be expressed by means of the Green function

([11]–[17])

$$L_0(x,x') = -iT\langle b^+(x), b(x')\rangle \qquad (18)$$

where x is the Humming distance (Humming coordinates) between those two MTs ([2]) and T is the 'time ordering operator' ([14]).

While the propagation of an MT from x to x' is described by the Green function

$$G_0(x,x') = -iT\langle a^+(x)a(x')\rangle \qquad (19)$$

Because of the similarity between the Hamiltonian (5) and those used in solid state physics, one can utilize the Feynman diagram technique developed for electrons, or excitons, and their interaction with photons ([11]–[17]) for a description of the dynamics of the system of MTs and the messages between them.

The most simple case ready to be discussed by this technique is repeated passing of the same, or similar, message, along, or with, a chain. Then one can easily sum up the corresponding series of the graphs.

Examples:

a) Passing the maintaining stream of a communication between MTs belonging to the systems of MTs which store the same piece of information (event) ([1], [3]).

Particularly in the case of a piece of information which repeats under the same conditions, as in the case of a piece of academic knowledge ([1], [3]), the interaction between any two MTs of such a system is the same and the system of MTs is crystal-like.

Then one can easily diagonalize the Hamiltonian (1) into the form ([1])

$$H = \sum_k E_k a_k^+ a_k \qquad (20)$$

which means that the piece of information ε_0 is stored in the set of MTs by a kind of 'exciton' ([1], [3]). The lifetime of such pieces of information is infinite. They belong to the long-term (or, are not less stable than medium-term) MTs ([1], [3]) (an example of such a system is a 'generalized chemical laser' related to the same event ([1, 3]).

b) In the case of the majority of everyday events, the systems of MTs which store them are disordered ([1], [3]) and the Hamiltonian (1) can be diagonalized only by means of more sophisticated methods, like the SCCF method ([20]). The lifetime of its eigenvalues ([1], [3]) is finite. They belong to the system of short-term MTs ([1], [3]) which are constantly forgotten, for example during sleep ([1], [3]), and their sieve builds more stable long-term MTs, already of a crystalline kind (the academic knowledge has this accelerated by a sufficient amount of stiffly controlled (defined) repetitions).

The crystalline situation can be technically further processed by means of the Feynman graph technique used to sum up the series of the graphs related to the quanta of the message ([12], [13]), of the form

$$L \equiv \text{▬▬▬} \qquad (21)$$
$$= \text{- - - -} + \text{- - - -○- - - -} + \text{- - - -○- - - -○- - - -} + \ldots$$
$$= (L_0 - \bigcirc)^{-1}$$

where $\text{- - - -} = L_0$. The dressed propagator of a piece of information

$$L = \text{▬▬▬}$$

([12]) can be evaluated by summing the series to give a propagator with 'bubble'. It is the simplest form of the polarization mass operator ([12], [13]), given by the expression ([13])

$$\bigcirc = \ln \Gamma_0 \int G_0(p-k)\, G_0(k)\, dk \qquad (22)$$

It expresses the fact that the original message stored in the loop of the whole system of MTs related to it is 'heavier', and more senior and important, than its original single form. It is connected with the inertia (e.g. the Lorenz type) of the flow during a single exchange between two MTs. This gives to the original FPI extra stability for its existence.

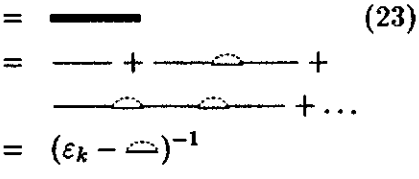
In the case when MTs are too close to each other (e.g. a weak interactive (narrow band) crystalline case) the polarization operator may have its value too close to a given L_k in (1) and the Green function (13) has a singularity and its value diverges to infinity. One gets a 'blow up solution'. This expresses the fact that such a system may catastrophically collapse ([1]).

c) Because such a 'bubble' of the type given by equation (20) is also presented in the interaction operator (9), it means that a repetitive exchange provides a strong force especially in the case of a narrow issue.

The summation of that series gives 'screene-like'–Yukawa type interaction. This means that the large and frequent exchange of information results in a 'short' Humming distance interaction (short Yukawa type forces), which may change the local metabolism.

It may force MTs to explode, as in the case of cells found in the brains of patients suffering from schizophrenia.

d) A similar result can be obtained for the propagator of an MT, like a vesicle, when it propagates along the path upon which it can collect the same, or similar, messages related to its original content. Then the 'dressed' propagator of an 'informed' MT can be written in the form



(23)

where —— $= G_0$. The mass operator of G, its simplest form, has its explicit analytical structure ([13])

$$\text{⌓} = \ln \varepsilon_0 \int G_0(p-k)\, D_0(k)\, dk \qquad (24)$$

where ε_0 is an 'agreement' charge, of the message relative to the original message.

When the mass operator in the denominator is equal to one of the original frequencies of an MT, or related to it, or to one of its original informational quanta, then the denominator becomes zero and the propagator 'blows up'.

The expressivity of the formalism also works in the case of a disagreement (negative resonances).

References

[1] J. Slechta: *On Quantum Statistical Theory of the Brain as a Generalized Laser*, in Cybernetics and Systems: The Way Ahead, ed. J. Rose, 919-24, Thales Publications, Lytham St. Annes, 1987.

[2] J. Slechta: *The Brain as a 'Hot' Cellular Automaton*, in the Proceedings of the 12th International Congress of Cybernetics, Namur, Belgium, 862-69, 1989.

[3] J. Slechta: *On Molecular Foundation of the Thermodynamics of the Thought Processes in the Brain*, in the Proceedings of the 12th International Congress of Cybernetics, Namur, Belgium, 870-77, 1989.

[4] J. Slechta: *On Thermodynamics of the Thought Processes within a Living Body in a Changing Environment*, in the Proceedings of the 12th International Congress of Cybernetics, Namur, Belgium, 878-85, 1989.

[5] J. Slechta: *On Thermodynamics of the Thought Processes within a Living Body which Exchanges both Energy and Matter with Its Environment*, in the Proceedings of the 12th International Congress of Cybernetics, Namur, Belgium, 886-98, 1989.

[6] J. Slechta: *On Spectral Properties of Neural Networks: Their Relation to Psychological Properties of Animal*, in the Proceedings of the 1990 Principle James Williams Congress, Amsterdam, Holland, 1990.

[7] J. Slechta: *On the Molecular Theory of the Biological Time (Circadian Rhythms, Psychological Time)*, presented at the 13th International Congress of Cybernetics, Namur 1992 (to be published).

[8] J. Slechta: *On a Brain Theoretical Cybernetic Concept of the Soul*, Proceedings of the 13th International Congress of Cybernetics, pp. 585–9, Namur, Belgium 1992.

[9] J. Slechta: *On Thermodynamical Theory of the Cybernetic Design of the Mouth, Anus and Genital*, presented at the 13th International Congress of Cybernetics, Namur, Belgium 1992 (to be printed in Cybernetica).

[10] J. Slechta: *On Duality of Energy and Information*, presented at the 13th International Congress of Cybernetics, Namur, Belgium 1992 (to be printed in Cybernetica).

[11] A.S. Davydov: *Quantum Mechanics*, Pergamon Press, Oxford 1976.

[12] B.M. Agranovich: *Theory of Excitons*, 'Science' Publishers, Moscow 1968.

[13] Bonch-Bruevich and L. Tyablikov: *Green Function Method in Statistical Physics*, North Holland Publishing Company, Amsterdam 1962.

[14] R.D. Mattuck: *A Guide to Feynman Diagrams in the Many-body Problem*, McGraw-Hill, London 1967.

[15] A.A. Abrikosov, L.P. Gorkov and I.E. Dzyaloshinski: *Methods of Quantum Field Theory in Statistical Physics*, Prentice-Hall, New Jersey 1963.

[16] A.C. Davydov: *Theory of Molecular Excitons*, 'Science' Publishers, Moscow 1968.

[17] R.D. Mattuck: *Advances in Physics*, **17**, 509-62 (1968).

[18] J.C. Eccles: *Proceedings of the Royal Society*, B 227, 411-28 (1986).

[19] J.C. Eccles: *Psychoneuroendrocrinology*, **7**, 271-83 (1982).

[20] J. Slechta: *Phys. Stat. Solidi (b)*, **120**, 328-39 (1983).

INTEGRATIVE DOMAIN ANALYSIS VIA MULTIPLE PERCEPTIONS

Wilhelm Rossak and Tamar Zemel
Systems Integration Laboratory
Department of Computer and Information Science
New Jersey Institute of Technology
University Heights
Newark, NJ 07102
USA
rossak@pluto.njit.edu
Phone: (201) 596-659

Keywords: domain analysis, systems integration, software engineering

Edited by: Matjaž Gams
Received: June 16, 1993 **Revised:** August 3, 1993 **Accepted:** August 10, 1993

Domain analysis is proposed as an essential activity for the integrated development of large, complex systems within an application domain. As an extension of traditional system-analysis methods, it is used as a means to assure global, inter-project coordination.

Domain analysis provides a universal, comprehensive, and non-constructive domain model. This domain model is used as a common basis for understanding by all developers in the domain and as essential input for the requirements specification phase in each project. Since an application domain is perceived differently by the many entities who have different relations to that domain, we propose building the domain model as an integration of these perceptions. Each perception represents the phenomena of the domain from the viewpoint of a specific group of users, managers, customers, or authorities. To facilitate this type of domain modeling, we propose using a domain modeling schema (domain schema) that consists of pre-specified element-types (modeling primitives) for the domain. This domain schema can be specialized and adapted to support capturing different perceptions and (re)integration of all perceptions into one comprehensive domain model. The proposed approach generalizes and extends existing system analysis methods and is compatible with object-oriented concepts.

1 Introduction

The reduction of hardware prices, the advances in technology and the maturity of customers and developers have led to the development of software systems in many different and new domains. Generally, these systems are large and complex. In many cases they integrate a set of systems into a comprehensive solution [6].

Typically, the constituent systems of these "systems of systems" are developed in an uncoordinated way: they are seen as a local solution to a very specific problem in the domain, reflecting only limited knowledge about the domain and including no preparation for future integration requirements [26]. Therefore, the integration of multiple constituent systems into one system is done in a post-facto manner by ad-hoc and non-systematic solutions [10] and driven by new technologies, e.g., communication or databases, instead of concepts and appropriate methodologies [13]. As these technology-driven approaches do not efficiently solve the difficulties that exist in the development of integrated systems, any addition of a new system, replacement of an existing system, or incorporation of a new technology becomes a tremendous effort.

The problems discussed in the previous paragraphs and the search for a comprehensive engineering methodology have led us to the conclusion that a

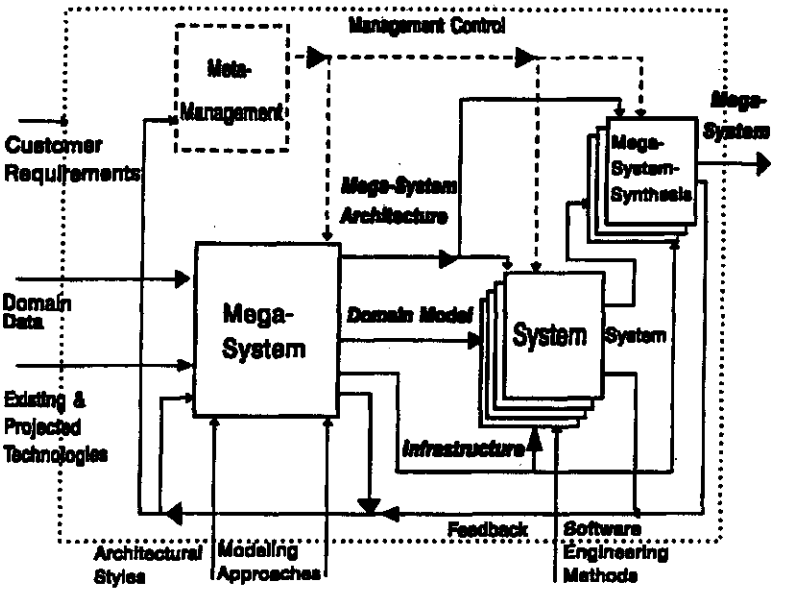


Figure 1: A Process Diagram for Mega-System Development

new approach is required for the development of what we call Mega-Systems - large, complex, and integrated systems which aim at solving a set of related problems in an application domain. A framework for the integrated development of Mega-Systems, incorporating engineering, managerial and technological aspects, has been proposed [12, 13, 14, 15, 16, 17, 18, 26]. At the management level, the framework divides the development of a Mega-System into multiple coordinated projects. It distinguishes between a meta-management task, responsible for the whole development effort and long-term, global objectives, and local management activities, responsible for the smaller projects and local, temporary objectives.

At the engineering level, the framework defines a process model which specifies all tasks, and their deliverables and interrelationships, required for developing Mega-Systems. The engineering process emphasizes the coordination required for developing the constituent systems and proposes domain analysis, Mega-System architecture design and infrastructure acquisition as essential activities to ensure this coordination [18].

Accordingly, the process model includes, in addition to traditional projects (system tasks), a Mega-System task that has the role of project coordination within the application domain and deals with general, long-term objectives (Figure 1). A Mega-System synthesis task integrates single systems into a coherent Mega-System. The whole process is managed by the meta-level management task that controls the other tasks and determines the policy and directions for the whole system.

The Mega-System task includes domain analysis, Mega-System architecture design and infrastructure acquisition (Figure 2). These tasks provide a domain model, a Mega-System (integration) architecture and a common infrastructure to be used by the various projects developing systems in the domain. The domain model is intended to provide a common basis for understanding the domain. Mega-System architecture design defines common design and implementation concepts and overall structure for the Mega-System. The infrastructure is intended to provide a stabilized interface between the applications and enabling technologies. Using these elements, the development of Mega-Systems is pre-planned and based on effective concepts. A project is no longer an isolated activity in a limited part of the domain, but a coordinated activity that fulfills a meaningful role in the domain.

This paper focuses on the role of domain analysis in a framework for the development of Mega-Systems and the process required to achieve an appropriate domain model. Section 2 specifies the fundamental requirements for a domain model. Section 3 discusses the principles of integrative domain analysis. Section 4 describes the concept of a domain schema as a means to facilitate domain modeling. Section 5 defines a process model for domain analysis. An example for integrative domain analysis in the insurance domain, using object-oriented notation as an underlying modeling approach, is described in section 6.

2 Fundamental Requirements

Domain analysis has been a topic of discussion in many different application and research environments [2, 9, 11]. The purpose of domain analysis in our

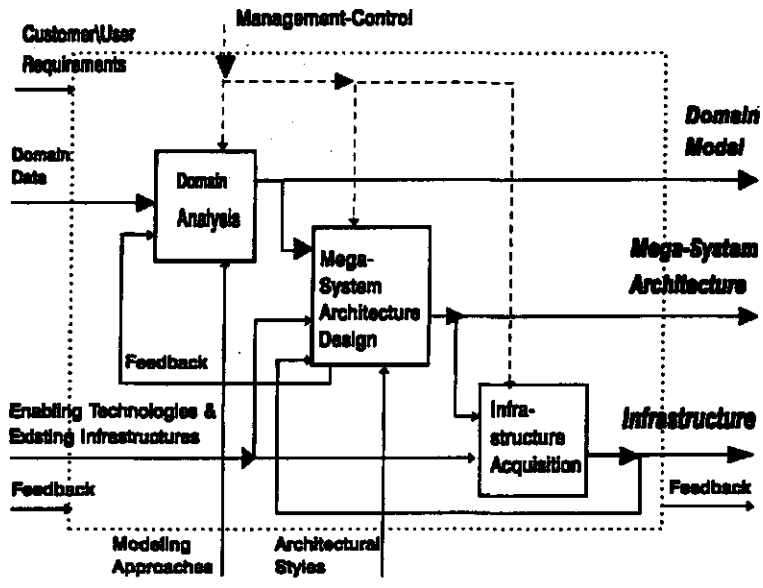


Figure 2: A Process Diagram for the Mega-System Task

framework is to specify a domain model which is used to support the development of software systems in the analyzed domain. The required domain model serves as a common basis for understanding the domain. It is used as a reference model, thesaurus and knowledge base, which captures the essential information required to understand the application domain. Domain analysis is intended to address and rectify the following difficulties in software development: neglect of overall, long-term issues; the need to deal with multiple, unstable requirements of customers with different aims and needs; and coordination and communication problems.

The domain model serves as a basis for refinement or specialization during the requirement specification phase of the various projects which develop systems within the application domain. It is also used as an input for the Mega-System architecture design activity for which it represents the domain. Feedback from system developers and Mega-System architecture design includes recommendations for improvement and corrections of the domain model.

To provide a common basis of understanding, a domain model must be:
- Universal,
- General,
- Comprehensive,
- Nonconstructive, and
- User-friendly.

Universality is required because the model is used by every project developing any system in the domain. For example, a university domain model will be used for the registrar system and the accounting system, as well as the foreign student system. Furthermore, since we are striving for integrable systems, it is essential to identify the relationship of each system to the other parts of its environment. A universal model of the whole domain will facilitate the development of such integrable systems.

Generality is required because the model is not intended to be used for a specific instance (system) in the domain, but rather as a common model for all systems in the domain. For example, a university domain model represents all universities, not a specific university. As a general model, a university domain model includes such concepts as academic year and terms, but the actual number of terms, their lengths and schedules vary with the university and so are not represented in the model. A model that fits only a specific instance or a particular system would not provide a common basis for understanding all systems in the domain. The analysis of a specific instance is, from this viewpoint, merely traditional system analysis.

On the other hand, we must limit the generality of the model to ensure its usability. If the model is too general, it will either include too many alternatives and become unmanageable, or be too abstract and so not provide adequately detailed information. For example, an aircraft carrier domain model should represent all types of aircraft carriers, but not battleships or arbitrary military vessels.

Comprehensiveness is required since the model serves as a common basis for understanding, and so it must include all the essential kinds of information regarding the domain. Thus, the model should include information about the things in the domain, their interactions, concepts, and any useful knowledge.

It is also important for a domain model not to concentrate on constructive aspects, i.e., design and im-

plementation. A conceptual model for an application domain without constructive elements provides a broader basis for systems implementation and improves the reusability of the domain model. Constructive elements usually belong to the solution domain and tend to restrict a model to a specific design or implementation, hiding the essential concepts of the domain. We propose that constructive aspects be dealt with separately, during Mega-System architecture design and infrastructure acquisition [26, 18].

Finally, the domain model must be user-friendly, since it is intended for use by systems analysts, architecture designers, end-users, etc., and not merely by software systems, e.g. application generators. Machine readability is required to support the model by CASE tools, but is not an intrinsic element of the technique.

3 The Principles of Integrative Domain Modeling

An application domain D is a comprehensive, internally coherent, relatively self-contained field or area of action, business, research, etc., supported by software systems. An application domain D consists of phenomena $\{P_1, P_2, \ldots, P_n\}$. For example, universities, banking or military vessels could be considered as application domains. The phenomena in a domain are perceived subjectively by different entities who have different relations to the domain. These entities understand and model a domain and its phenomena according to their perspective, emphasizing or neglecting specific aspects of reality. The following sections describe the underlying principles of integrative domain modeling using the terms phenomenon, perception, and aspect.

3.1 Phenomena

A phenomenon P in an application domain is a concept that abstractly represents instances of a thing, activities, relations or constraints in the domain. The domain model abstracts the phenomena of the domain, and omits details about specific instances of the domain. For example, a domain model for a university might abstract student, department, registration, enrollment in a course, a policy for student acceptance, and the difference between Chemistry and Mathematics. Abstractions of domain phenomena included in the domain model are called domain elements. The characteristics of a phenomenon in the domain are then represented as attributes of this element in the domain model. For example, the attributes of an element representing a student might be: Name, Address, Student-Id, and Grade Point Average (GPA).

Since the domain model must be comprehensive, it must represent both phenomena belonging to the static structure of the domain, e.g., objects and relations, and the dynamic interactions of the domain, e.g., processes and events (c.f. also [19]). The static structure of a domain might include objects (entities) and their relationships [5]. In the object-oriented approaches, objects of the domain with similar characteristics are grouped into object-classes [7]. Objects are also related to other objects in various ways, e.g., by generalization, specialization, aggregation or association. We see these relationships themselves as phenomena belonging to the static structure of the domain.

The dynamic interactions of the domain include behavior patterns of phenomena. The object-oriented methods specify operations that can be applied to instances of a given object-class [7], but we also want to represent processes that may involve more than one object, relationship or activity. Using processes, it is possible to represent the methods and techniques used to solve problems in the domain.

A process is a set of activities operating on or executed by various phenomena in the domain, the results of these activities, and their sequencing. An example of a process in the university domain is registration. In this process, a student selects courses, receives approval from his/her advisor, registers, and is billed. We also propose representing events and state transitions in the domain model as part of the dynamic structure. An example of an event is failure in an exam. An example of a state transition could be a faculty member changing rank from assistant to associate professor.

A general model will also include a variety of other types of qualitative and quantitative information; statistical information; averages and maximums. It might include rationales and constraints.

3.2 Different Perceptions

A domain is perceived differently by entities which have different relationships, roles or concerns with the domain [24]. For example, a student and a registrar have different perceptions of the university domain. These differing perceptions arise from the different relationships of the perceivers to the domain and may include different groups of elements or the same elements under different names or with different attributes and roles. To achieve universality and comprehensiveness, we propose building the domain model by integrating multiple domain perceptions.

First, entities with a significant perception of the domain are identified. These entities may influence the domain or be influenced by it. For example, in the university domain, we might identify faculty, registrar, board of education, student and staff, as entities which have significant perceptions. After iden-

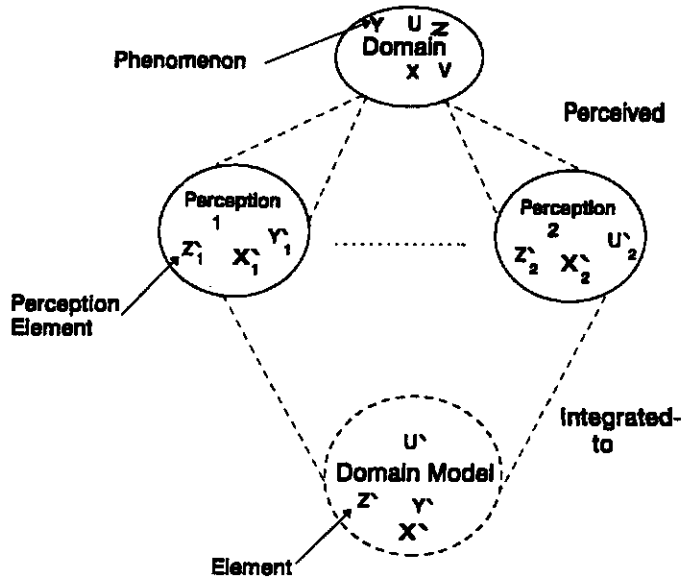


Figure 3: A Domain Model as an Integration of Multiple Perceptions

tifying these entities, it is necessary to build a perception for each of them. Thus, a perception is a representation of the domain as perceived by an entity who has a significant role in or concern with the domain. Phenomena are represented in a perception as perception-elements. For example, perception-elements for a faculty's perception of the university domain might be student, course, department. All the perception-elements for a specific phenomenon, perceived by different significant perceivers, will finally be merged into one integrated element in the domain model. For example, the registrar's student-perception-element, the faculty's student-perception-element, and the student's student-perception-element are integrated into the final student-element in the domain model.

Figure 3 illustrates the integration of several perceptions into a domain model. A domain with phenomena X, Y, Z, U, and V is perceived by some significant perceivers. Perception-1 of perceiver-1 includes perception-elements X_1', Y_1', and Z_1'. Perception-2 of perceiver-2 includes perception-elements X_2', Z_2', and U_2'. Thus, the domain model includes elements X', Y', Z', and U', where element X' integrates both X_1' and X_2' and element Z' integrates both Z_1' and Z_2'.

3.3 Aspects of Phenomena

Any phenomenon has many aspects: physical, structural, dynamic, static, etc. Physical aspects refer to the physical properties of phenomena: dimensions, weight, composition. Structural aspects pertain to the manner in which a phenomenon is organized, or related to other phenomena, e.g., the components of the phenomena or membership. Dynamic aspects describe changes of the phenomenon, e.g., the frequency of change, the originator of change, etc. An aspect usually deals with a specific set of attributes. Aspects are also discussed by Wimmer [25], who calls an aspect a view.

The significance of aspects is domain specific. For example, in the CAD domain, physical aspects are more important than legal aspects, which on the other hand might be more significant in the banking domain. Since a perceiver is often interested in a subset of domain aspects, the significant aspects for different perceivers may be disjointed or they may overlap. For example, the significant aspects of a faculty perceiver in the university domain might be structural, static, and dynamic aspects, while for the physical plant manager they might be the structural, static, and physical aspects.

4 Structuring the Model

4.1 An Overview of the Domain-Schema

The required universality and comprehensiveness of a domain model imply that the domain model must be able to handle a large amount of information. In order to manage this information, support the modeling technique and make uniform the various perceptions, we propose using a domain modeling schema (domain-schema). The domain-schema is used to define the modeling primitives which will later be used to represent the phenomena of the domain as elements. We call the modeling primitives element-types. A similar idea is suggested in [25].

A domain-schema consists of element-types that can be used to represent a group of elements with similar attributes. Such a group of elements that might be represented using the same element-type is called an element-class. Possible element-classes are object, relationship, event, process, etc. The object-element-class, for example, includes all elements that belong to the domain, where each element represents a group of objects in the domain with similar attributes. For example, the object-element-type will be used to represent faculty, student, etc.

As all elements that belong to the same element-class are represented using a specific element-type, we indirectly define a possible set of attributes for these elements. The element-type acts as a template which is filled in with actual attributes for each element. Since every phenomenon has multiple aspects, we divide the attributes into groups based on exactly these aspects. We call these groups element-aspects. Thus, an element-type is a union of element-aspects, where each element-aspect includes the attributes of one aspect for a specific element-class (see also Tables 1, 2, and 3 and the following sections).

The domain-schema can be considered as a meta-schema, and its element-classes and element-types as meta-classes and meta-types: Element-types are used to describe classes of elements, e.g., object-classes and processes. Thus, an element-type does not represent a class of instances of the domain with similar attributes, e.g. student or registration, nor instances of the domain, e.g. J. Smith or the CIS Department.

Beyond element-types such as objects and relations, our schema might also include other element-types, e.g. processes, constraints, or special domain-dependent element-types. Which element-types are included in a specific domain-schema depends on the application domain and the basic modeling philosophy. In our examples we stay close to the object-oriented approach, e.g., [19].

It is important to differentiate the domain schema and schemata of databases. Schemata of databases describe the structure of the database and represent elements of the problem space itself, e.g. student, faculty, department, etc. Domain schemata define the modeling primitives to be used for modeling the domain: objects, relationships, events, etc.

4.2 Dimensions

Rumbaugh et.al. [19] suggest modeling a system from three viewpoints: the object model, the dynamic model, and the functional model. We also recommend dividing the domain model and its elements into orthogonal and interrelated parts, considering each part as a dimension of the domain model. In order to implement this idea, we specify domain-schema dimensions as groups of inter-related element-types. Each group is used for modeling a dimension of the domain model. The number of dimensions and their content depend on both the modeling approach and the domain. For example, a model based on the Entity-Relationship (ER) approach [5] includes only a data dimension with the entities and relationship as the element-types.

Dimensions, aspects, database views, and perceptions are different. The aspects in a domain schema deal with attributes of phenomena and group them into sets. The dimensions of the domain model are groups of interrelated element-types used to simplify modeling by dividing the model into interrelated parts. Views in databases are used to define virtual objects and restrict user access to parts of the data; this is close to the perception concept in our approach. Perceptions are used to model the domain from a specific point of view and include a subset of the phenomena/elements, dimensions, and aspects of the domain.

4.3 More About Element-Types

An element-type is defined in the schema by a set of attributes divided into aspects and represented by a frame-template (see Table 1). Each frame includes actual aspects and their attributes. Composite attributes consisting of other attributes are also allowed and are drawn as split cells, e.g. attribute 21. Multi-valued attributes, which may appear more than once, are designated by a star (*); attributes that appear at least once are designated by a plus (+).

Defining a domain-schema requires identifying element-classes and then defining their element-types with appropriate sets of attributes. The number and kind of element-classes and the content of their element-types depend on the modeling approach, the application domain, and the significant aspects. Similar templates, but with a restricted set of element-types and no explicit division of the attributes into aspects, appear in [4].

Table 2 is an example of an object-element-type. This element-type might be used for representation of objects in a domain model. If the actual aspects in the analyzed domain are physical, structural, static, dynamic, legal and logical, the template includes only attributes of these aspects. The physical aspect includes physical characteristics of objects, e.g. *dimension, weight, color*, etc. The structural aspect includes the *generalizes, specializes, and aggregates* attributes to enable inheritance and aggregation of objects into composite objects. An object can be a generalization of several objects and therefore the *generalizes* attribute is a multi-valued attribute.

Objects have their lifecycle. It is possible to describe the object life cycles by state diagrams [22]. These diagrams include the various states an object might have and the transitions between them. Accordingly, the static aspect might include a state attribute that rep-

Element-Type					
Aspect 1	Aspect 2		Aspect 3	..	Aspect N
Attr. 11: Type 11	Compsite-Attribute 21	Attr. 211: Type 211	Multi-valued attribute31 *: Type 31		Multi-valued Attr. N1 (Not Empty) +: Type N1
		Attr. 212: Type 212			
		Attr. 213: Type 213			
Attr. 12: Type 12	Attribute 22: Type 22				Attr. N2: Type N2
Attr. 13: Type 13					. .
					Attr. Nk: Type Nk

Table 1: A Template for Element-Types

Object					
Physical	Structural	Static	Dynamic	Legal	Logical
Dimension *: Numeric	Generalizes *: Object	State: State	State-diagram: Diagram	Status: Text	ID: Identifier
Weight: Numeric	Specializes *: Object		Method *: Method		purpose: Text
Velocity: Numeric	Aggregates *: Object				Value: Numeric
Color: Text	Part-of *: Object				Status: Text
Material: Text					Role: Text
Temperature: Numeric					

Table 2: An Example of an Object-Element-Type

resents the actual state of the object. The dynamic attributes of an object might include a reference to a state transition diagram that describes the transition between the various states an object can assume. *Method* (Table 2) is a multiple attribute that represents the methods that can be applied to instances of the object-class. Similarily, the other aspects include a list of relevant attributes.

It is important to understand that this is an example only of an object-element-type. A process-element-type or any other element-type will use a different set of attributes. Furthermore, since schemata are domain dependent, it is possible that, for other domains, the object-element-types will have different aspects and sets of attributes.

Using the object-element-type of Table 2, it is possible to represent uniformly the various object-classes in a domain. Each object-class in the domain is defined using the template: the upper part of each cell includes the element-type's attribute specification, while the lower part includes the actual instantiation of the attribute. A representation of a building object-element in a university domain appears in Table 3. In this case, the *weight, velocity, and temperature* attributes of the physical aspect, the *generalizes* attribute of the structural aspects, and other attributes are not used and therefore are designated by "–". The Method attribute includes the *assign* method that assigns a building to a department. The *aggregates* attribute includes all the elements that are aggregated by a building: *floor, room, and elevator*. Similarly, other attributes might be examined and specialized according to the actual element.

4.4 Perceptions

The significance of various domain aspects may vary with each perceiver. It is also possible that a perceiver is interested only in a limited set of element-classes. To simplify modeling we suggest using a perception-schema for each perceiver. A perception-schema is derived from the domain-schema by selecting the perceiver's actual element-classes and restricting the schema to the significant aspects for that perceiver. It is also possible to limit the attributes of an element-aspect to include only a subset of attributes of the element-aspect in the perception-element. Thus, a perception-schema is a sub-schema of the domain-schema determined by the set of element-types, the set of element-aspects of each element-type, and the set of attributes in an element-aspect. Table 4 illustrates a perception-element for a building that includes physical, structural, and static aspects only.

Using the perception-schema as a guideline, a model of the domain is built for each perceiver. This process is similar to analysis approaches and might be supported by analysis notations, e.g. the object-oriented approach. Thus, the relevant phenomena of the domain for a given perception are identified. Later, each phenomenon is classified into one of the element-classes. Using the appropriate perception-element-type, the different attributes of the element are specified. Dimensions can be used to simplify the modeling process by dividing the model into smaller, manageable parts.

We propose that domain experts will build these perceptions supported by domain analysts. The processes that build the perceptions can be executed concurrently by different groups. However, since the domain-schema is used for derivation of all perception-schemata, the resulting perceptions will be both structured and coordinated.

4.5 The Integrated Model

The various perceptions are finally integrated into a domain model. We first find which perception-elements of different perceptions represent the same phenomenon. Later, all the perception-elements for a specific phenomenon are integrated into a unified element. The perception integration process is based on the element-aspects. If different perceivers are interested in different sets of aspects, the final integrated element is the union of the various element-aspects. If an aspect is relevant to more than one perceiver, attributes in the appropriate element-aspect are compared. Conflicts in element-type, names, attributes, roles, etc. are resolved and a unified element-aspect is derived. When conflicts cannot be resolved, the conflicting versions are all incorporated into the element.

Figure 4 illustrates this integration. A domain with phenomena $P = \{P_1, P_2, \ldots, P_m\}$ is perceived by n perceivers. A perception for each perceiver is built: $Perceptions = \{Perception_1, \ldots, Perception_n\}$. Each perception consists of perception elements that represent the significant phenomena for the perceiver. PE_{ij} denotes the perception element of perceiver i for phenomenon j. The domain model integrates all the perception-elements for phenomenon j:$\{PE_{ij} \mid i =$ *1 to n, where phenomenon j is significant to perceiver i*$\}$ into an element E_j.

The process is similar to integrating views or schemata of databases [3, 20, 21, 7, 8]. However, schema integration [20, 7, 8] is done on static structure elements (objects and their relations) only, while here integration is done for elements of all dimensions of the model, including processes, events, etc.

4.6 A Summary of the Domain Schema Concept

A domain schema is in principle nothing but a set of custom-tailored modeling primitives that allow the an-

Building					
Physical	**Structural**	**Static**	**Dynamic**	**Legal**	**logical**
Dimension *: Numeric	Generalizes *: Object	State : State	State-Diagram: state-diagram	Status: status	ID: Alpha-numeric
Height, Width, Length	(--)	(--)	(--)	(Approved, Restricted)	Building-ID
Weight: Numeric	Specializes *: objects		Method *		purpose*: Enumerate
(--)	(--)		Assign		(Teaching, Administration, Sports,Storage, Utilities)
Velocity: Numeric	Aggregates *: object				Value: Numeric
(--)	Floor, Elevator, Room				(--)
Color: Enumerate	Part-of *: object				Status
(--)	Campus				(--)
Material: Enumerate					Role: Alphanumeric
(Wood, Blocks)					
Temperture: Numeric					Role: Alphanumeric
					(--)

Table 3: An Example of a Building-Object-Element

Building-Perception-Element		
Physical	Structural	Static
Dimension*: Numeric	Generalizes*: Object	State: State
Height, Width, Length	(--)	(--)
Weight	Specializes*: Object	
(--)	(--)	
Velocity	Aggregates *: Object	
(--)	Floor, Elevator, Room	
Color	Part-of *: Object	
(--)	Campus	
Material		
(Wood, Blocks)		
Temperature		
(--)		

Table 4: An Example of a Building-Perception-Element

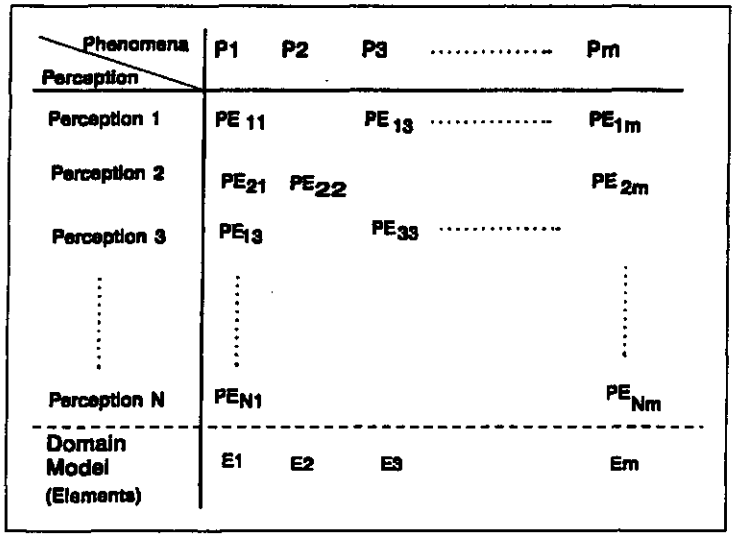


Figure 4: The Integration of Perception-Elements into Elements

alysts to pursue their own investigations and modeling efforts without losing sight of the necessary integration of the different perceptions they are describing.

The basic element in a domain schema is a set of attributes:

$A = \{A_1, \ldots, A_n\}$

As mentioned before, we allow for structured and multivalued attributes. Element-types (ET), the modeling primitives we are striving for, are now simply defined as a set of attributes:

$ET_j = \{ASP_{j1}, \ldots, ASP_{jm}\}$ where
$1 \leq x \leq m$ $ASP_{jx} = \{A_{j1}, \ldots, A_{jl}\}$ and
$1 \leq y \leq l$ $A_{jy} \epsilon A$

Aspects, the ASP_{jx}, are used to group attributes according to their use in describing the physical, structural, static, etc. information of a phenomenon. Which aspects are used in a specific analysis is domain dependent and can be specified by the team of analysts and users, resulting in a set of possible aspects in the model:

$ASP = \{ASP_1, \ldots, ASP_k\}$

Be aware that this is a top-down process, starting with a basic decision on element-types (and interesting aspects), and only then proceeding to the attributes necessary to capture the information for a specific aspect in a specific element-type (see the earlier discussion in this section and section 5). Thus, the same aspect might be described by a different set of attributes for different element-types.

A domain schema (DS) is the set of all defined element-types:

$DS = \cup ET_j$

Based on the domain schema, element-types can be used to model phenomena (P) of the real-world as elements (E) of the domain model. All elements that are modeled by the same element-type form an element class (EC):

$EC_j = \{E \mid derived - from(P, E)$
$\& \; modeled - by(E, ET_j)\}$

The elements E of an element class EC_j can be seen as instantiations of, or as "filled-in", element-type ET_j. An element in the model is a mapping of a phenomenon onto an element-type.

If only one perception, one viewpoint, is involved, a domain model (DM) is specified as the unification of all element-classes derived from the basic domain schema:

$DM = \cup EC_j$

However, in the case of modeling via multiple perceptions, the domain schema is refined to reflect these perceptions before any mapping or actual modeling takes place. After deciding on the involved perceptions, element-types can be adapted to perception-element-types (PET):

$PET_j =\subset ET_j$

The subset relationship is defined as providing the possibility for a perception-element-type to include only a subset of the aspects, and within accepted aspects, a subset of the attributes as specified for the original element-type.

Similarily, a perception-schema (PS) is a subset of the domain-schema, including some or all of the element-types either in their original format or as derived perception-element-types. What should be included in a perception-schema is strictly domain and user dependent.

$PS =\subset DS$

As with element-classes, perception-element-classes (PEC) are defined as:

$PEC_j = \{PE \mid derived - from(P, PE)$
$\& \; modeled - by(PE, PET_j)\}$

Perception-elements (PE) are the result of mapping

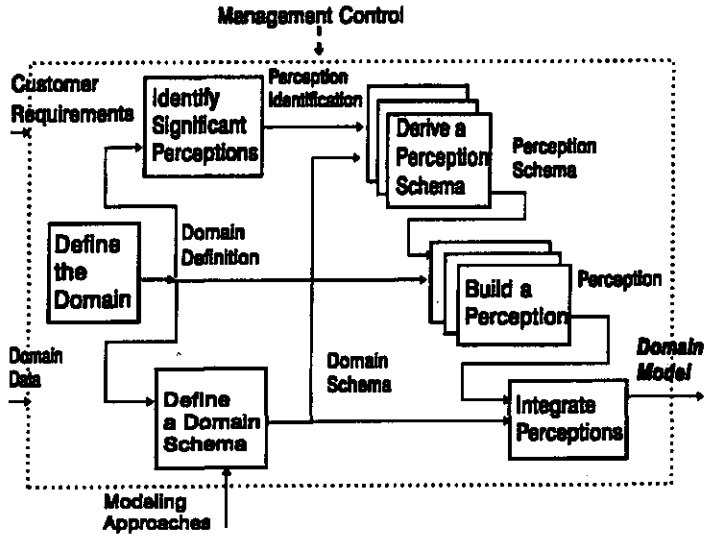


Figure 5: A Process Diagram for Domain Analysis

a phenomenon into a perception-element-type. As we have multiple perception-schemata, one phenomenon can now be introduced into different perception-schemata using similar but specialized perception-element-types (to capture the different types of relevant information about it).

To derive a consistent domain model (DMP), now based on multiple perceptions, it is necessary to integrate the perception-elements of all perceptions (see sections 4.5 and 6):

$DMP = \int PEC_i$

An example and an overview of the domain analysis process are given in the following sections.

5 The Domain Analysis Process

Focusing on the activities required for providing the domain model, we can describe the previous discussion by a process diagram (Figure 5). Based on an initial domain identification, significant perceivers are identified. A domain schema is defined according to the domain characteristics and a modeling approach. This schema consists of element-types that can be divided into dimensions. For each perceiver, a perception-schema is derived and then used in building his/her perception. All perceptions, finally, are integrated into a domain model. Common inputs to all the subprocesses are domain data and customer requirements.

Verification, validation and quality assurance are done as part of every task or sub-task to ensure that the model accurately describes the application domain. Since domains evolve, domain analysis must be a continuous activity. To maintain the effectiveness and usability of the model, essential changes in the domain as well as feedback from the various projects should be evaluated and reflected in the domain model, as required. The process is active as long as systems are being developed and maintained in the application domain.

6 A Partial Example for Integrative Domain Analysis in the Insurance Domain

Insurance is a system that enables a person, business or organization to transfer loss exposure to an insurance company which indemnifies the insured for covered losses and provides for the sharing of the costs of losses among all insured [23].

The objective of integrative domain analysis is to build a domain model as an integration of significant perceptions of the domain. The decision as to what the significant perceptions are is to a certain degree subjective. However, in most cases a preliminary analysis will provide at least a meaningful set of candidates. In the insurance domain, we could identify the insurance company (which we call the insurer), the agent, and the insured as significant perceivers.

The insurer is a company or a person that contracts to indemnify another in the event of loss or damage. The insured is a person, business, or organization that purchases an insurance policy to protect it/himself from losses. Insurance companies usually market their products through agents. The agent serves the insured and represents the insurer. Other significant perceivers of the domain, e.g. the actuary who computes insurance rates, government regulators, and claims adjustors are omitted in this simplified ex-

ample.

In the following sections we describe these perceptions and their integration. For simplicity, we identify elements of the domain model using the object-oriented approach. Since these elements are "self-defined", and because of space limitations, we do not provide the details of each element and the appropriate templates. An extended example can be found in [1].

Each perception is built using multiple dimensions. As dimensions depend on the chosen modeling approach and the actual domain, we select the static (object) and the functional dimensions as used in [19]. We look first at the static dimension for each perception, then at the functional. During the integration phase, the appropriate dimensions of each perception are integrated. For each dimension, the different perception-elements are mapped into actual domain phenomena. Resolution of conflicts and definition of a unified model are then demonstrated for the selected dimensions and perceptions.

6.1 The Static Dimension

The static dimension is illustrated by object diagrams consisting of objects (drawn as rectangles) and their relations (drawn as lines or arrows). Generalizations are designated by the $\bigwedge$ sign. Perception descriptions are typed using italics and upper-case letters to denote relevant objects, relationships, and processes. A line denotes a one-to-one relation, an arrow denotes a one-to-many relation, and double arrows denote a many-to-many relation.

- The Static Dimension of the Insurer Perception

The *Insurer Issues Policies* and is *Represented-by Agents*. The *Insured Purchase* policies *Sold* by *Agents*. Insurers are specialized to *Life*, *Health*, *Property* and *Liability*. An insurer is *Reinsured* by a *Reinsurer*. The insurer *Indemnifies* a *Loss Covered* by a policy (Figure 6).

- The Static Dimension of the Insured Perception

The *Insured Buy Policies* from an *Agent*. Policies are *Issued* by the *Insurer*. The insurer is *Represented-by* agents. Policies *Cover Insurance-items Owned* by the insured. Insurance-items can *Have Losses*. Insurance-item is a generalization of *Car*, *Life*, and *Building*. The insurer *Compensates* losses of an insurance-item (Figure 7).

- The Static Dimension of the Agent Perception

The agent *Represents* the *Insurers* and *Serves Clients*. Clients *Buy Policies*. The agent has *Private*, *Business*, and *Group* Clients. The agent *Sells* policies *Issued* by an insurer. Policies *Cover Losses Indemnified* by the insurer. Policies are specialized to *Life*, *Property*, and *Health*. Property policies are specialized to *Building*, *Motor Vehicle*, and *Property in Transit* (Figure 8).

- Integration of the Static Dimension

We use object and relationship tables to identify which perception-elements belong to the same phenomenon. The object table (Table 5) maps the appropriate perception-elements of each perception to domain objects.

Upon examining the objects in the different perception, we find:

– Objects that appear in all perceptions with the same name, e.g. *Policy*, *Insurer*, and *Loss*. These elements will be included in the domain model using the same name.

– Objects having the same role but with different names, e.g. *Insured*. This is called *Client* in the agent perception. We decide to use *Insured* in the domain model.

– Objects that appear in only one perception, e.g. the *Reinsurer* in the insurer perception and the *Insurance-item* in the insured perception. Both elements are added to the domain model.

– Specializations that do not appear in every perception. We prefer to see all these specialized objects in the domain model. Thus, we include the specialized *Insured* types, i.e. *Group*, *Private*, and *Business*, the specialized *Insurer* types, the specialized *Insurance* items, and the specialized *Policies*.

Upon examining the relationships in the multiple perceptions, we find that:

– Some relationships appear in all perceptions with the same names, e.g. *Issue* and *Sell*. These relationships will be represented in the domain model under the same name.

– Some relationships appear with different names, e.g. *Indemnify* is also called *Compensate*, and *Purchase* is also called *Buy*. We select a name for these relationships and use it in the domain model.

– Several relationships do not appear in all perceptions, e.g. *Reinsure* in the insurer perception, *Serve* in the agent perception, *Own* in the insured perception. These relationships are all included in the domain model.

– A name of a relationship is used in different perceptions between different pairs of objects, e.g. *Cover* appears in both the agent and insurer perceptions between *Policy* and *Loss* and in the insured perception *Cover* appears between *Policy* and *Insured-item*. We choose the name *Insured*: thus, *Cover* will represent the relationship between *Policy* and *Insurance-item*; *Have* will represent the relationship between *Insurance-item* and *Loss*. We do not use the relationship between *Policy* and *Loss*.

The relationship table (Table 6) consists of pairs of objects, their perception names and the name of the relationships in the domain model.

Figure 9 illustrates the integrated static dimension.

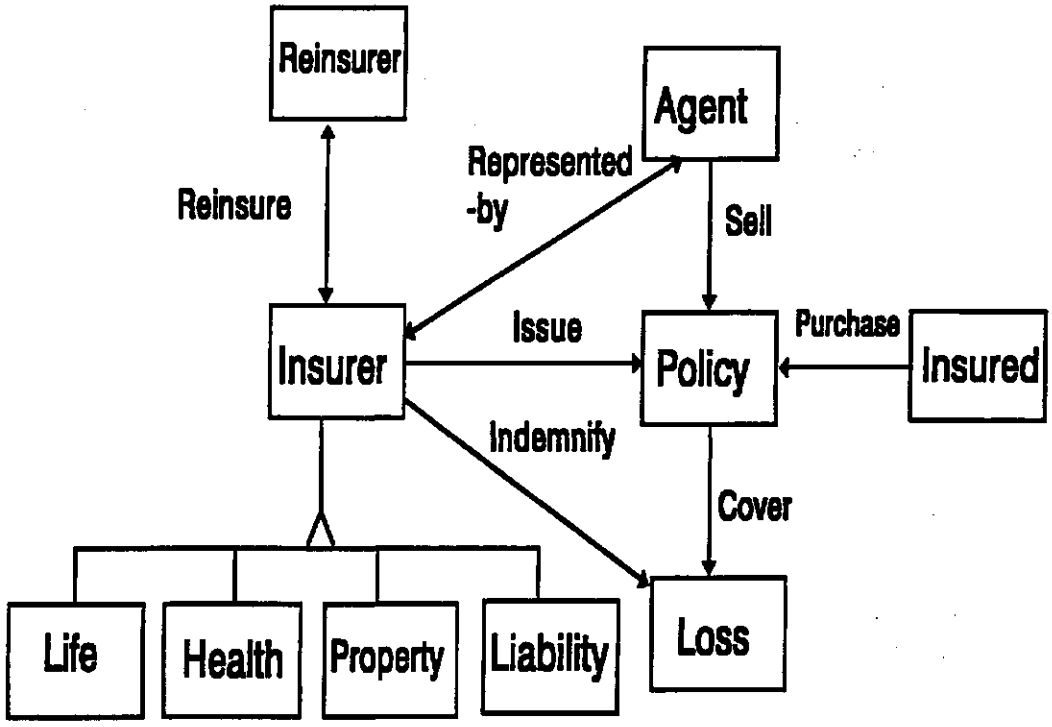


Figure 6: The Static Dimension of the Insurer Perception

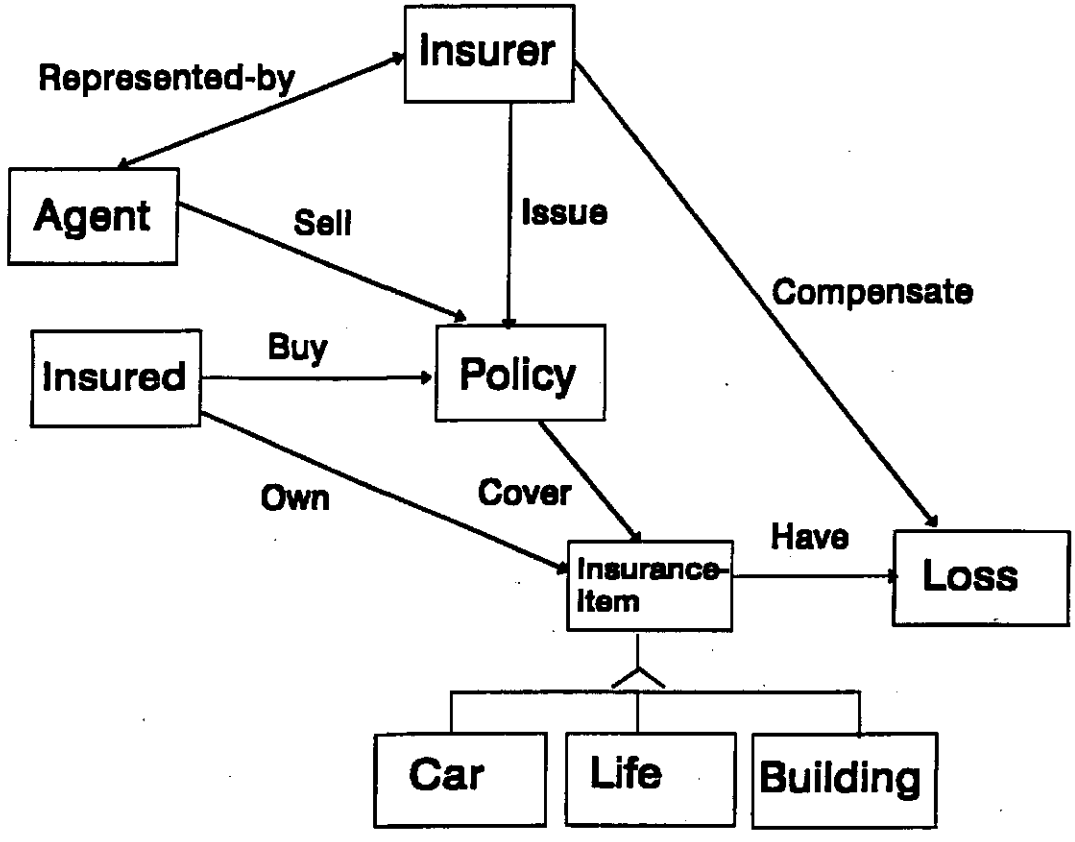


Figure 7: The Static Dimension of the Insured Perception

Insurer Perception	Insured Perception	Agent Perception	Domain Model Objects
Insurer • Life • Health • Property • Liability	Insurer	Insurer	Insurer • Life • Health • Property • Liability
Agent	Agent	Agent	Agent
Insured	Insured	Client • Private • Business • Group	Insured • Private • Business • Group
Policy	Policy	Policy • Life • Property • Health	Policy • Life • Property • Health
Loss	Loss	Loss	Loss
Reinsurer			Reinsurer
	Insurance Item • Car • Life • Building		Insurance Item • Car • Life • Building

Table 5: Mapping of Perception-Elements to Domain Objects

Objects	Insurer Perception	Insured Perception	Agent Perception	Domain Model Relationship
Insurer-Agent	Represented-by	Represented-by	Represent	Represented-by
Insurer-Policy	Issue	Issue	Issue	Issue
Reinsurer-Insurer	Reinsure			Reinsure
Insurer-Loss	Indemnify	Compensate	Indemnify	Indemnify
Policy-Loss	Cover		Cover	Cover
Policy-Insured	Purchase	Buy	Buy	Purchase
Policy-Insurance-Item		Cover		
Agent-Policy	Sell	Sell	Sell	Sell
Insurance-Item-Loss		Had		Had
Insured-Insurance-Item		Own		Own
Agent-Insured			Serve	Serve

Table 6: Mapping of Relationships

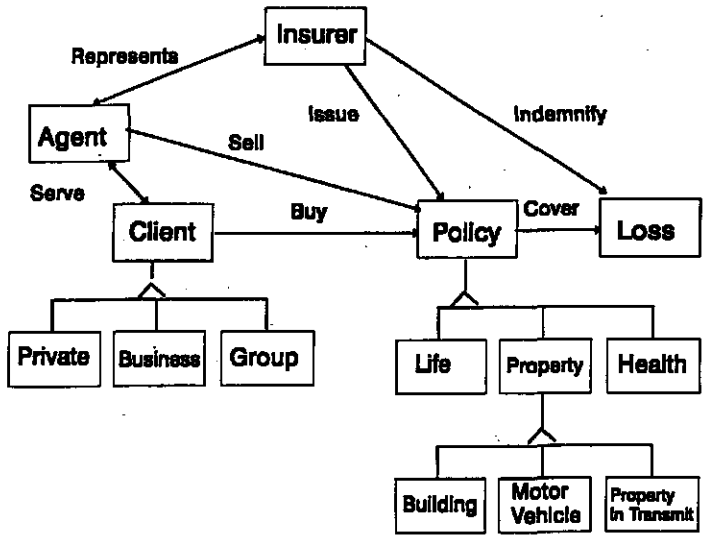


Figure 8: The Static Dimension of the Agent Perception

6.2 The Functional Dimension

The functional dimension includes processes (drawn as bubbles), data and control flows (drawn as solid and dashed arrows), and data stores (drawn as double lines). Sources or terminators are drawn as squares. We use the process of issuing a policy to illustrate the integration of the functional dimension.

- The Functional Dimension of the Insurer Perception

In the insurer perception, the insured *Fill-in an Application* for insurance of an insurance-item, provide *Insurance-item* and *Insured* information, and *Agree* to the insurance terms. In *Underwrite a Policy*, a *Policy* is prepared for approval based on the *Insurance-Rates* computed in *Compute Insurance Rates* by the actuary. These rates are computed according to *Statistical Tables*. After *Approving the Policy*, it is issued to the insured (Figure 10).

- The Functional Dimension of the Insured Perception

In the insured perception, an insured asks for quotes from different agents. The agents *Prepare Quotes* according to the *Insured Item* information. After several iterations, the insured *Agree* and *Fill-in an Application* for insurance. Based on this *Application*, a *Policy* is *Underwritten* and issued to the insured (Figure 11).

- The Functional Dimension of the Agent Perception

According to the agent perception, insured *Ask for a Quote*. The agent *Prepares a Quote*. If the insured *Agrees* to the quote, the agent and the insured *Fill-in an Application*. A *Policy is Underwritten* and is passed for *Approval*. The approved *Policy* is issued to the insured (Figure 12).

- The Integrated Functional Dimension

Similar to the integration of the static dimension, we use tables for mapping the perception-elements into actual domain phenomena.

Table 7 lists the process elements. Examining the process elements we find:

– Processes that appear in each perception, e.g. *Fill-in an Application, Underwrite a Policy*. These processes are included in the domain model.

– Processes that appear only in some perceptions, e.g. *Approve a Policy, Compute Insurance Rates*, and *Prepare a Quote*. These processes are also included in the domain model.

– The *Prepare a Quote* process appears in one perception as a single process and in another perception as a multiple process. In this case we decide to represent it as a multiple process. The difference arises because the insured can ask different agents to prepare quotes and only then selects one offer.

Table 8 includes mapping of the control and data flows, the sources and terminators, and the data stores. Figure 13 represents the integrated functional dimension.

7 Summary

This paper proposes using integrative domain analysis as a means of coordination in a multi-project development environment. The domain model built using this approach provides a basis for understanding the entire application domain with all its different systems.

Integrative domain analysis, as suggested by its name, is based on integration of the significant perceptions of the domain. Each perception consists of elements representing the phenomena of the domain from a specific point of view. Elements are modeled

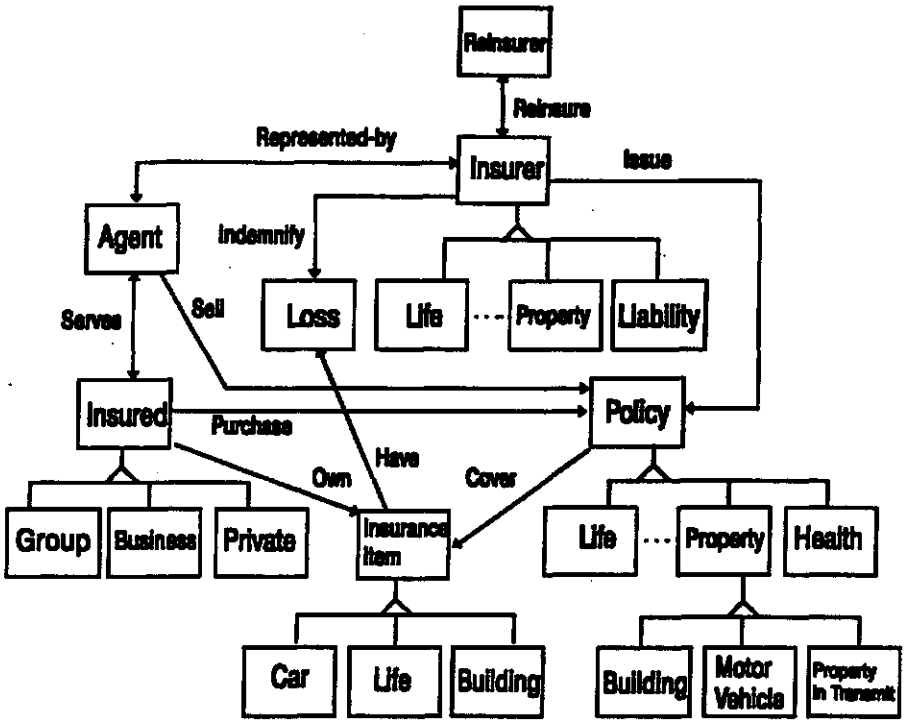


Figure 9: The Integrated Static Dimension

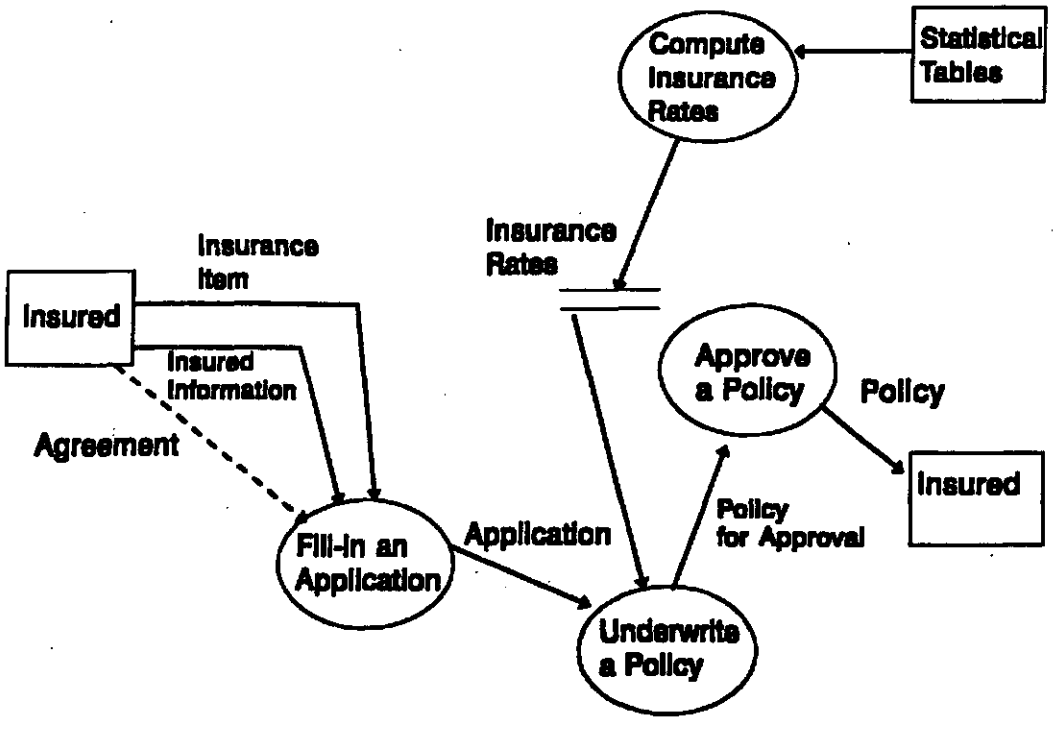


Figure 10: The Functional Dimension of the Insurer Perception

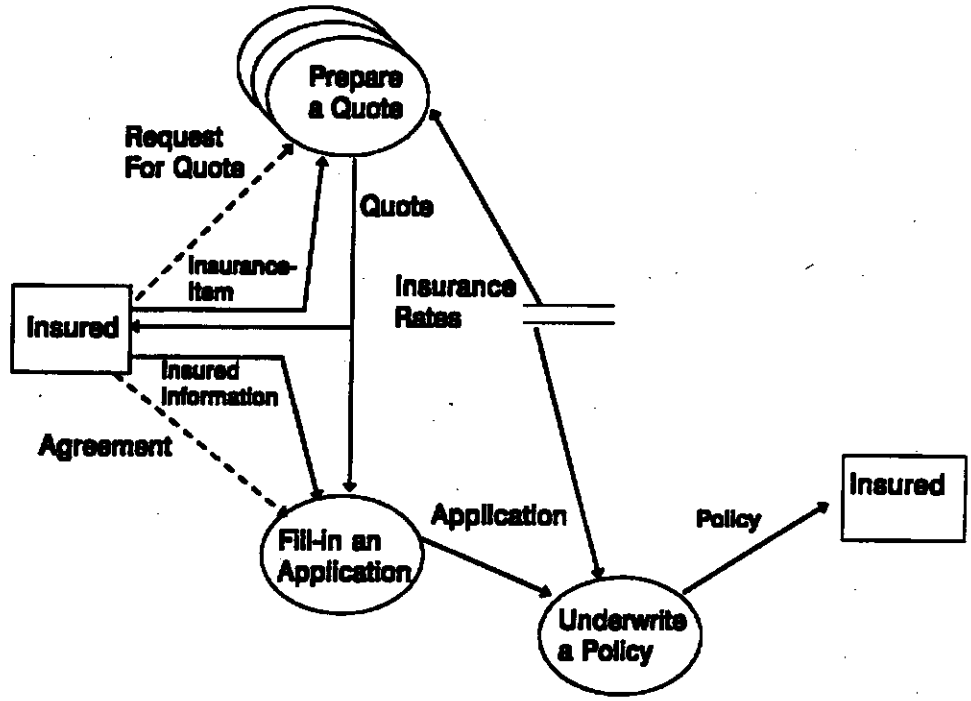


Figure 11: The Functional Dimension of the Insured Perception

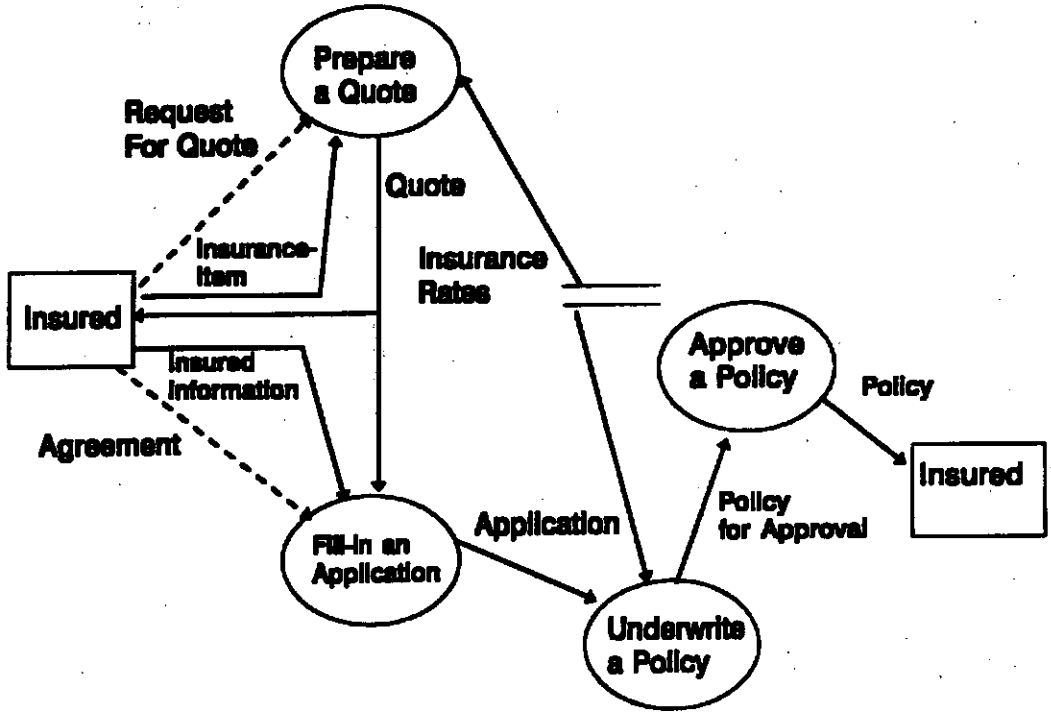


Figure 12: The Functional Dimension of the Agent Perception

Insurer Perception	Insured Perception	Agent Perception	Domain Model Processes
Fill-in an Application	Fill-in an Application	Fill-in an Application	Fill-in an Application
Underwrite a Policy	Underwrite a Policy	Underwrite a Policy	Underwrite a Policy
Compute Premium Rates			Compute Premium Rates
Approve a Policy		Approve a Policy	Approve a Policy
	Prepare a Quote (multiple process)	Prepare a Quote	Prepare a Quote

Table 7: Process Mapping

Insurer Perception	Insured Perception	Agent Perception	Domain Model
Insured	Insured	Insured	Insured
Insurance-Item	Insurance-Item	Insurance-Item	Insurance-Item
Insured-Information	Insured-Information	Insured-Information	Insured-Information
Agreement	Agreement	Agreement	Agreement
Application	Application	Application	Application
Policy for Approval		Policy for Approval	Policy for Approval
Statistics Tables			Statistics Tables
Insurance Rate	Insurance Rate	Insurance Rate	Insurance Rate
	Request for Quote	Request for Quote	Request for Quote
	Quote	Quote	Quote

Table 8: Flows, Sources, Terminators, and Data Stores Mapping

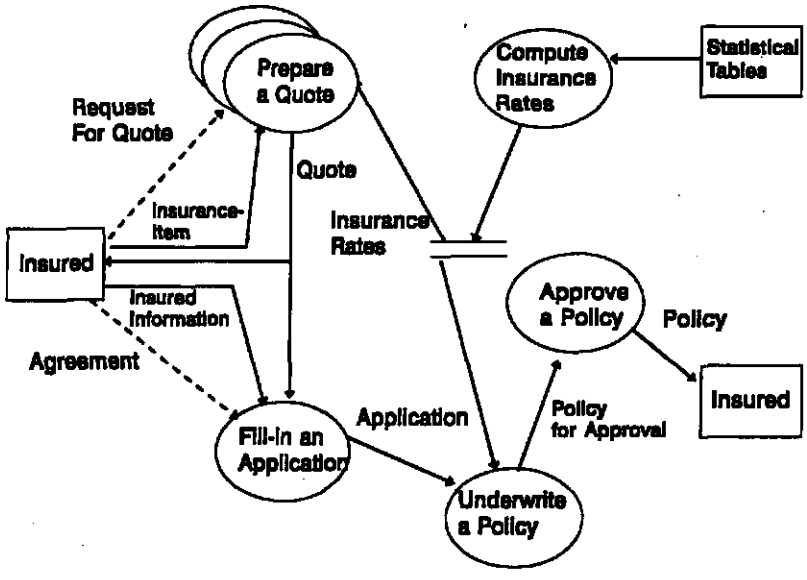


Figure 13: The Integrated Functional Dimension

using element-types that are pre-defined in a domain schema. Perceptions are then integrated into a complete model.

Acknowledgements

The authors wish to thank James McHugh for his many valuable insights.

Special thanks go to Selma Aganovic for her analysis of the insurance domain and to Vassilka Kirova for her suggestions and her help in preparing this document.

References

[1] Aganovic, S., *Domain Analysis for the Insurance Domain*, Master's Project, Department of Computer and Information Science, NJIT, Newark, NJ, USA 1993.

[2] Arango, G., and Prieto-Diaz, R., "Domain Analysis Concepts and Research", in Prieto-Diaz R. and G. Arango (eds), Domain Analysis and Software Systems Modeling, IEEE Computer Press, Los Almitos, CA, USA, 1991, pp. 9-26.

[3] Batini, C., Lenzerini, M., Navathe, S. B., "A Comparative Analysis of Methodologies for Database Schema Integration", ACM Computer Surveys, Vol. 18, No. 4, December, 1986, pp. 323-361.

[4] Booch, G., *Object-Oriented Design with Applications*, Benjamin Cummings, Redwood City, CA, USA, 1991.

[5] Chen, P.P.S., "The Entity-Relationship Model: Towards a Unified View of Data", ACM Trans. on Database Sys., Vol. 1, No. 1, 1976, pp. 9-36.

[6] Eisner, H., Marciniak, J., and McMillan, R., "Computer Aided Systems of Systems (S2) Engineering", Proc. of the 1991 IEEE/SMC International Conference on Systems, Man, and Cybernetics, Charlottesville, VA, USA, IEEE Computer Society Press, October 1991, pp. 531-537.

[7] Geller, J., Perl, Y., and Neuhold, E., "Structural Schema Integration with Full and Partial Correspondence using the Dual Model", Technical Report, Department of Computer and Information Science, NJIT, Newark, NJ, USA, 1991.

[8] Geller, J, Mehta, A., Perl, Y. and Sheth, A., "Algorithms for Structural Schema Integration", Proc. of the IEEE Second International Conference on Systems Integration, Morristown NJ, USA, IEEE Computer Society Press, June 1992, pp. 604-614.

[9] Iscoe, N., Williams, G. B., Arango, G., (eds), *Proc. of the Domain Modeling Workshop*, 13th International Conference on Software Engineering, Austin, TX, USA, May 1991.

[10] Power, L. R., "Post-Facto Integration Technology: New Discipline for an Old Practice", in Proceedings of the First International Conference on Systems Integration, Morristown, NJ, USA, IEEE Computer Society Press, April 1990, pp.4-13.

[11] Prieto-Diaz R. and G. Arango (eds), *Domain Analysis and Software Systems Modeling*, IEEE Computer Press, Los Almitos, CA, USA, 1991.

[12] Rossak, W., and Ng, P. A., "Some Thoughts on Systems Integration - A Conceptual Framework", International Journal of Systems Integration, Vol.1, No. 1, Kluwer Academic Publ., Dordrecht, Holland, 1991, pp.97-114.

[13] Rossak, W., and Prasad, S., "Integration Architectures - A Framework for Systems Integration Decisions", Proc. of the IEEE Internat. Conference on Systems, Man, and Cybernetics, Charlottesville, VA, USA, IEEE Computer Society Press, October 1991, pp. 545-550.

[14] Rossak, W., "Integration Architectures - A Concept and A Tool to Support Integrated Systems Development", Technical Report, Department of Computer and Information Science, NJIT, Newark, NJ, USA, 1991.

[15] Rossak, W., and Ng, P. A., "System Development with Integration Architectures", Proc. of the IEEE Second International Conference on Systems Integration, Morristown, NJ, USA, IEEE Computer Society Press, June 1992, pp. 96-103.

[16] Rossak W., Welch L., Zemel T., and Eder J., "A Generic Systems Integration Framework for Large and Time-Critical Systems", in Halang W., Stoyenko A. (eds): NATO Advanced Study Institute (NATO ASI 910698) on Real-Time Computing, Mullet Bay, Saint Martin, Springer Verlag, October 1992, to appear.

[17] Rossak W., and Zemel, T., "Engineering Large and Complex Systems with Integration Architectures", PD-Vol. 49 - Computer Applications and Design Abstractions, ETCE '93, Houston TX, USA, February 1993, pp. 189-195.

[18] Rossak W., Zemel T., and Lawson H., "A Meta-Process Model for the Planned Development of Integrated Systems", International Journal of Systems Integration, Vol. 3, No. 3/4, Kluwer Academic Publ., Dordrecht, Holland, 1993, pp. 225-249.

[19] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W., *Object-Oriented Modeling and Design*, Prentice Hall, Englewood Cliffs, NJ, USA, 1991.

[20] Sheth, A. P., Larson, J. A., Cornellio, A., and Navathe, S., "A Tool for Integrating Conceptual Schemas and User Views", Proc. of the Fourth International Conference on Data Engineering, Los Angeles, CA, USA, Feb 1988, pp. 176-183.

[21] Sheth, A. P., and Larson, J. A., "Federated Databases Systems for Managing Distributed, Heterogeneous, and Autonomous Databases", ACM Computer Surveys, Vol. 22, No. 3, September 1990, pp. 183-236.

[22] Shlaer, S., Mellor, S. J., *Object Lifecycles - Modeling the World in States*, Yourdon Press Computing Series, NJ, USA, 1992.

[23] Smith, B. D., Triechmann, J. S., Wiening, E. A., *Property and Liability Insurance Principles*, Insurance Institute of America, Malvern, Pennsylvania, 1987.

[24] Thimm, H., "Domain Analysis for Systems Integration", Master's Thesis, Department of Computer and Information Science, NJIT, Newark NJ, USA, 1992.

[25] Wimmer, K., and Wimmer, N., "Conceptual Modeling Based on Ontological Principles", Report, Siemens Munich, BRD, 1992.

[26] Zemel, T., *MegSDF - Mega-Systems Development Framework*, Doctoral Dissertation, Department of Computer and Information Science, NJIT, Newark NJ, USA, 1993.

A Prolog-Based Representation for Integrating Knowledge and Data

Xindong Wu †
Department of Artificial Intelligence, University of Edinburgh,
80 South Bridge, Edinburgh EH1 1HN, U.K.

† Address after 14 July 1993:
(xindong@coral.cs.jcu.edu.au)
Department of Computer Science, James Cook University,
Townsville, QLD 4811, Australia.

Keywords: knowledge representation, Prolog, deductive databases, semantic information

Edited by: Matjaž Gams
Received: May 13, 1993 **Revised:** June 21, 1993 **Accepted:** July 7, 1993

Although the history of data base systems research is one of exceptional productivity and startling economic impact, many advanced applications have revealed deficiencies of the conventional data base management systems (DBMS's) in representing and processing complex objects and knowledge. Object-oriented approaches are currently very popular in processing structurally complex objects, while deductive data bases or logic data bases have been proposed as a solution to those applications where both knowledge and data models are needed. However, it has been characteristic of the current deductive data bases that only actual data is represented explicitly in logic, while the data schema is implicitly described in form of predicates. In this paper, we present a Prolog-based representation. It binds the actual data and data schema together in a natural and flexible way. In addition to expressing all the information which can be represented in the entity-relationship (E-R) model, the representation can represent other kinds of semantic information as well.

1 Introduction

Over the past twenty years, data base research has evolved technologies that are now widely used in almost every computing and scientific field. However, many new advanced applications, including computer-aided design (CAD), computer-aided manufacturing (CAM), computer-aided software engineering (CASE), image processing, and office automation (OA), have revealed that traditional DBMS's are inadequate, especially in the following cases [10]:

- Conventional data base technology has laid particular stress on dealing with large amounts of persistent and highly structured data efficiently and using transactions for concurrency control and recovery. For some applications like CAD/CAM [14] where the data schemata need to vary frequently, new data models are needed.

- In some applications like geographical data and image data, the semantic relationships among data need to be represented in addition to the data itself. Conventional data models ([8], such as hierarchical, network, and relational models) in data base technology cannot support any representation facility for complex semantic information.

- Traditional data base technology can only support facilities for processing data. Along with the developments of other subjects, like decision science and AI, more and more applications need facilities for supporting both data and knowledge management.

To widen the applicability of data base technology to these new kinds of applications, object-oriented data models have been proposed as the data models of next-generation DBMS's [2] to handle more complex kinds of data and objects and deductive data bases have been expected to support a solution to process both knowledge and data models.

In object-oriented approaches [1], complex data structures (e.g. multimedia data) can be defined in terms of objects. Data that might span many tuples in a relational DBMS can be represented and manipulated as a data object. Procedures/operations as well as data types can be stored with a set of structural built-in objects and those procedures can be used as methods to encapsulate object semantics. Containment relationships between objects may be used to define composite or complex objects from atomic objects. An object can be assigned a unique identifier. Relationships between objects can also be represented more efficiently in object-oriented data models by using a more convenient syntax than relational joins. Also, most object-oriented DBMS's have type inheritance and version management as well as most of the important features of conventional DBMS's.

Deductive data base systems provide knowledge management, supporting a number of rules for automatic data inferring and management of integrity constraints between data. Rules in deductive data bases are also called intensional data bases, while the explicitly stored data are called extensional data bases (EDB's). There are several different approaches [10, 3] to implementing deductive data base systems, such as integration and coupling on a physical or a logic level, but their EDB's are mostly relational. As the relational data model and Prolog have a common theoretical foundation [18] and Prolog is a programming language that contains within it the language of relations and can thus be used in a very direct way to implement relational data bases, much of the work on both deductive data base systems and even conventional relational DBMS's has been implemented in Prolog [5, 4, 6, 9], although the implementation is not always very efficient.

The normal way in existing deductive data base systems to model relational data bases in Prolog is based on the following analogies: a relational tuple corresponds to a fact in Prolog, the collection of tuples in a relation corresponds to the facts with the same predicate name, and constraints and queries are represented as Prolog rules. There are two disadvantages to this conventional approach:

- It does not represent data schemata explicitly. Relational data dictionaries are not described in Prolog. Users must remember exactly all structures of different fact collections when, e.g. defining relational operations in Prolog, which means it is impossible to tackle relations as variables. From the view of the DBA (data base analyst), the management of larger applications also becomes more difficult.

- It is inconvenient for data restructuring which presupposes the ability to add, modify and remove schema components and causes corresponding changes in the actual data.

This paper will present a Prolog-based representation. One of the motivations is to represent relational data bases in such a way that the above disadvantages of the conventional approach can be eliminated. The other motivation is the insufficiency of the E-R model, which is a widely adopted data abstract model for the conceptual structure design of data bases, in expressing semantic information. The simple relationship types in the E-R model, such as one to many (1:N) and many to many (M:N), cannot describe well the different explicit semantic features of the relationships among entities, still less the variations and developments of entities in function, performance, structure, status and attributes etc. with time and external variables' variations. The aim of the representation is to integrate knowledge and data in such a natural way that all the information which can be represented in the E-R model and other kinds of semantic information which cannot be described well in the E-R model can be easily expressed and that the semantic information can be used to couple ML facilities with data base and knowledge base technology in order to implement knowledge acquisition from data bases [15].

2 The Representation

Our representation consists of two parts: the first part for relational data bases and the second part for semantic information.

2.1 Representation of the relational model

There are two ways to represent relational tuples. One represents them as labeled n-tuples in the form of a set of $(attribute, value)$ pairs and the other as ordered n-tuples. In the second, an n-tuple is usually represented in the form of $(V_1, ..., V_n)$ where the values $V_1, ..., V_n$ appear in the same order as their field names in the relation schema. As lists are a common form of representation in Prolog, where the relative positions of elements can be taken as important, the representation below is based on the ordered n-tuples method.

The following is a BNF (*Backus Normal form*) notation for representing a relational data base within our representation.

```
<Database> :=
    <Relation> {, <Relation>}*
<Relation> :=
    relation(<Relation Name>
    <Field List>{<Tuples>}_0^1)
<Relation Name> :=
    <Prolog Name>
<Field List> :=
    <Field>{,<Field>}*
<Field> :=
    <Field Name><Field Type>
<Field Name> := <Prolog Name>
<Field Type> :=
    char|string|logical|integer|real|date
<Tuples> := <Tuple>{,<Tuple>}*
<Tuple> :=
    <Element>{, <Element>}*
<Element> :=
    char(Char)|string(String)
    |logical(Boolean)|integer(Integer)
    |real(Real)|date(String)
<Prolog Name> :=
    (any legal Prolog atom)
```

A relation generated by the above BNF notation has the structure of

$$relation(RelationName, FieldList, Tuples) \quad (1)$$

or

$$relation(RelationName, FieldList).$$

Each relation in a relational data base has a unique name, *RelationName*. The predicate *relation* describes all the fields and possible tuples in the relation *RelationName*. Fields in a relation are described by an ordered list, *FieldList*. Their types are identified by the atoms *char, string, logical, integer, real* and *date*, which denote the domain of single characters, character strings, truth-values, integers, real numbers and specific strings for date description. Each field can be uniquely identified as

$$field(RelationName, FieldName, Type). \quad (2)$$

The component *Tuples* in a relation supports a Prolog representation of relational tuples. It contains those tuples of which the relation value consists. In the *Tuples* in a relation, the value of each field appears in the same position as the field name in the field list. It is easy to define structural constraints which check that each tuple confirms to the fields description in a relation and is uniquely defined. This is the way our representation binds relational schemata and relational tuples. In other words, the *Tuples* component describes the relational tuples, whereas the components *RelationName* and *FieldList* belong to the relational data schemata. All of *RelationName, FieldList* and *Tuples* are represented explicitly and can thus be manipulated easily. Constraints between fields and dependency types in relationships will be represented in Section 2.2.

It is convenient to define a predicate:

$$keyfield(RelationName, KeyFieldList)$$

where $KeyFieldList := field\{, field\}^*$ as the key fields of relation *RelationName*. Since in some relational DBMS's (e.g. dBASE), key fields are not explicitly defined, we did not include the *keyfield* predicate in our representation.

2.2 Representation of more semantic information

The E-R model is one of the most successful methods of formulating useful abstract models in the

conceptual structure design of data bases and it is the key design aid for conventional data bases implemented under a wide variety of commercially available systems [4]. By focusing on the entities and their relationships, it structures the way designers approach the problem of creating extensible data bases. However, there are two substantial problems here. One is that transforming an E-R model into a relational model during the logical design of data bases results in loss of some semantic information that exists in the E-R model. In other words, the entities and relationships are not distinguished in the relational data model. It is impossible for the relational data model to describe the changes of relationship(s) and other entities caused by an entity in an E-R model. For example, *age* is an important factor for counting an employee's *salary* in many British institutions. However, we cannot explicitly express whether the employee's salary will increase according to the change of his/her age in the relational data model. The other problem is that the E-R model itself is insufficient in expressing complex semantic information as its relationship types, such as one to many and many to many, are too simple to describe explicitly semantic features of the relationships between entities and within entities themselves. For example, different types of relationships, such as logical implication and conceptual inheritance, cannot be expressed in the E-R model.

The E-R model and the relational data model are successful in those applications where only the ability to deal with large amounts of persistent and fixed-format data efficiently is needed. For new applications, such as those mentioned in the introduction, new representation models are in demand. Object-oriented data models are a new generation of extended data models, based on the relational data model. However, as we can see from their main features, briefly summarized in the introduction, object-oriented models are themselves data models, although some systems (e.g. POSTGRES [2]) have included rule processing facilities. Relational data management, object management and knowledge management are three different problem- solving techniques. They would all be needed in some complex applications.

Knowledge management entails the ability to represent, acquire and enforce a collection of expertise which is part of the semantics of an application. Such expertise describes integrity constraints among data in the application in addition to allowing the derivation of data which is usually called *virtual* data in contrast to the real data stored in the data base(s). The task of knowledge management is a key motivation of deductive data base research.

The representation described in this paper is basically designed for the approach that generates semantic networks from relational data base schemata [17]. Therefore, we have put an emphasis on representing the semantic information which cannot be represented in the relational data model and the E-R model.

Semantic information in the real world includes four different categories:

- descriptive knowledge about entities,
- inherent laws and constraints between attributes or fields in entities,
- relationships among entities which can be further divided into six types [16][1], i.e., hierarchy, fellow member, attribute, role, causality and logical implication, and
- dependency types in the relationships between entities.

The following are some predicates in our representation used to express semantic information. The examples for those predicates will be mainly drawn from the sample data base schemata in Figure 1.

2.2.1 Distinguishing entities and relationships

Each relationship (Relation) is distinguished with a predicate as

$$is - assoc(Relation). \quad (3)$$

[1] In order to give a more precise semantic classification, it is possible to divide one or more of the relationship types here into greater detail. The completeness of a semantic model can only be defined in terms of specific applications. We cannot say whether all the relationships here are necessary for every application. Neither can we say they are complete. However, as we can see from Figure 1, they do exist in the real world.

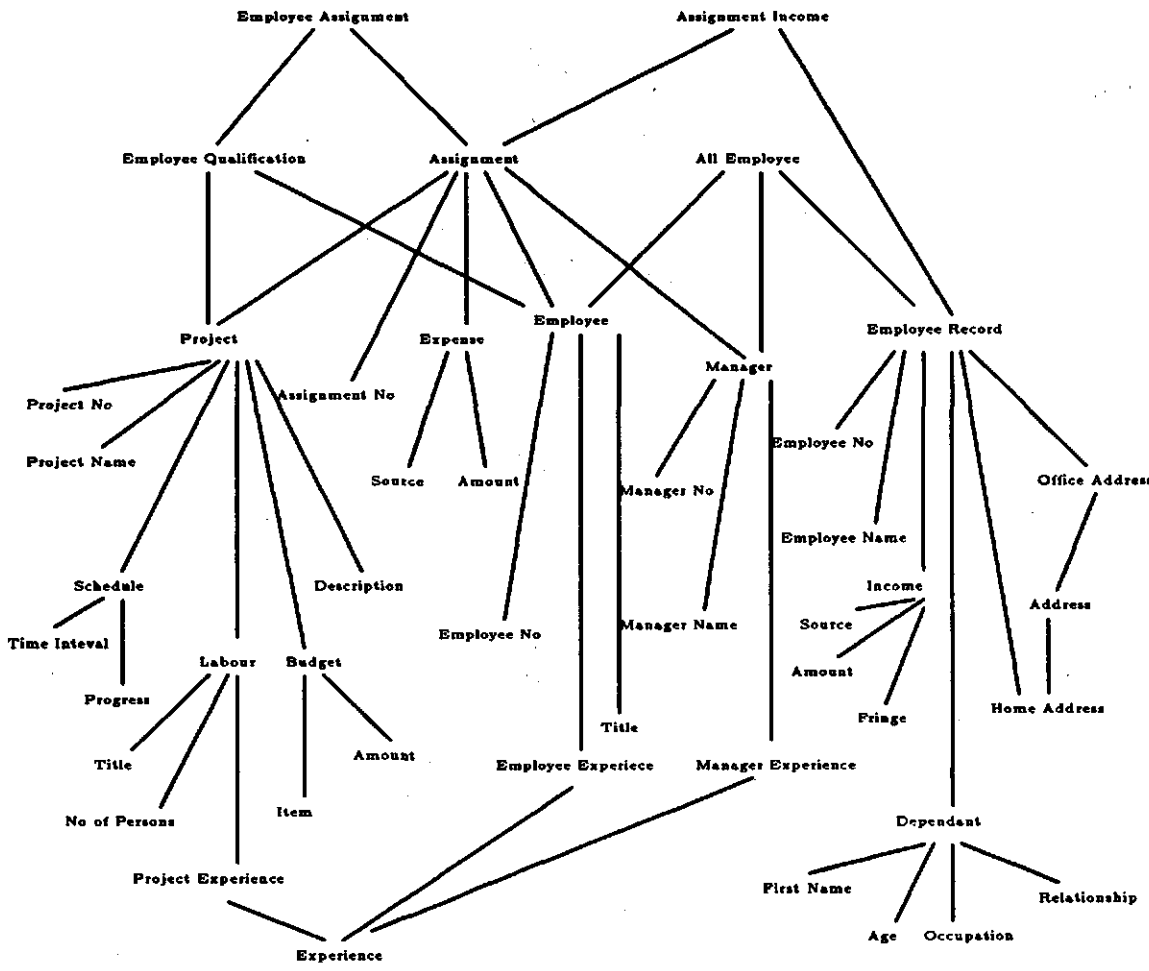


Figure 1: Sample Data Base Schemata (Derived from [7])

In Figure 1, *Dependant* and *Employee* are two entities whereas *Assignment* is a relationship indicating a manager monitors employees to work for a project.

Clearly, each entity (Entity) satisfies the feature below:

entity(Entity) :-
relation(Entity,_,_),
not(is-assoc(Entity)).

Each entity-relationship association is described with predicate *assoc-entity*

assoc-entity(Relation,EntityList,
AssocTypeList) (4)

where $AssocType \in \{1, N\}$, denoting the nature of an entity, is single or multiple valued in an association.

For instance, relationship *Assignment* contains entities *Employee, Manager* and *Project.*

Information about (3) and (4) can be found in the E-R model but it is lost when the E-R model is transformed into the relational data model.

2.2.2 Identifying the semantic type of each relationship

There are examples of six types of relationships in Figure 1:

- hierarchy which indicates conceptual inheritance: the relationships between *Employee* and *All Employee* and between *Home Address* and *Address*,

- fellow member: the relationship between *Home Address* and *Office Address*,

- attribute: *Labour* and *Budget* are two attribute entities of entity *Project*,
- role: *Employee Experience* and *Manager Experience* are two role entities in the *Assignment* relationship,
- causality: the *Labour.Title* of an employee in *Employee Qualification* may be a reason for his/her *Employee.Title* assignment in relationship *Assignment* and
- logical implication: the *Income.Fringe* of an employee can be concluded from his *Project* in *Assignment*, say (Employee No = 14, Project No = 4 → Income.Fringe = 150), in the *Assignment Income* relationship.

The semantic type (AssocType) of each relationship (Relation) is identified by

$$assoc - type(Relation, AssocType). \quad (5)$$

Different types of relationships have different 1) structural features in describing the formulation of the relationships, 2) semantic integrity constraints on data, and 3) operational features or behaviour, such as insertion, deletion, comparison and retrieval, on the data in the relationships [7].

2.2.3 Representing semantic labels in each relationship

Semantic labels are useful for processing natural-language like queries and firing machine learning engines in intelligent data base systems.

For each type of relationship, there are different semantic labels to identify different roles in the relationship. For example, in a causality relationship, there are two kinds of labels, *cause* and *effect*. In a logical implication relationship, there are also two kinds of labels, condition (*if*) and conclusion (*then*). A key entity in a relationship can be given a *key* label to identify the relationship. For example, if a *Project* needs a specific *Assignment*, we say the *Project* entity is a key entity in the *Assignment* relationship.

Each entity's semantic label in each relationship is identified by

$$label(Entity, Relation, Label). \quad (6)$$

For example, in the *Assignment Income* relationship, we have the following labels.

$$label(Project, AssignmentIncome, cause)$$

$$label(Income, AssignmentIncome, effect)$$

2.2.4 Representing deductive knowledge

Knowledge about causality and logical implication is necessary for deductive data bases to establish virtual data. In existing deductive systems, this is often represented as production rules. As there are several disadvantages inherent in conventional production rules, we represent deductive knowledge in the form of "rule schema + rule body" [11, 12]. The Prolog representation is thus

$$schema(Relation, CauseEntityList, ResultEntity), \quad (7)$$

$$body\text{-}left(Relation, No, CauseOrResultEntity, Attri, RelSym, Value), \quad (8)$$

$$body\text{-}right(Relation, No, ResultEntity, Attri, Value) \quad (9)$$

where *No* is used to identify different parts of the same body, *Attri* indicates an attribute and *RelSym* denotes a conventional arithmetic or symbolic relation.

In the example given for the logical implication, we can express it as:

```
schema(AssignmentIncome,Project,
    EmployeeRecord),
body-left(AssignmentIncome,1,
    Project, Project_No, =, 4),
body-left(AssignmentIncome,1,
    EmployeeRecord,EmployeeNo,      =,
    14),
body-right(AssignmentIncome,1,
    EmployeeRecord,IncomeFringe,150).
```

2.2.5 Representing constraints knowledge

Constraints are important in the relational data model. Three sorts of constraints have been classified and represented in our representation. The first is about the integrity of attributes in each relation,

constraint1(Relation,Attribute, RelSym,Value). (10)

For example, in the *Dependant* relation, the *AGE* attribute is supposed to be always less than 120.

The second is the dependency type of each relationship, such as one-to-one (which means a result entity tuple has a unique corresponding tuple of each cause entity, e.g. an *Assignment* tuple corresponds to a unique *Project* tuple), full (which means all possible tuples of the result entity have their corresponding cause entities' tuples, e.g. each *Assignment* tuple must have its corresponding *Project*, *Expense* and *Employee* tuples) or dual (each tuple of a result entity corresponds to a tuple of each cause entity and *vice versa*, e.g. each *Assignment* tuple has its own *Project* tuple and *vice versa*),

constraint2(Relation, MappingType). (11)

The third is the constraint relationship between an attribute in a relation and outer variables,

constraint3(Relation,Attribute, OuterVariableList, ConstraintString). (12)

See the example in Section 2.2.6 where *Year* could be an outer variable of Figure 1.

Here, semantic constraints about relational data have also been explicitly expressed rather than being hidden in application programs. This feature of our representation makes it easier to maintain and adapt application programs.

2.2.6 Representing regularities between attributes

These represent inherent regularities between attributes, for example, the time-dependent function of an attribute, and the function or logical dependency relationship among the attributes,

function((Relation,Attribute), (Rel,Attri),Function)* (13)

where $(Rel, Attri)*$ indicates a list of relational attributes. For instance, if an employee was born in 1950, his age can be computed by the following regular knowledge.

function((Employee, Age), [(Time, Year)], Age=Year-1950)

Functional dependency and multivalued dependency between attributes from the design theory of relational data bases [8] can be represented with Predicates (13) and (11).

3 Discussion

Predicates (1), (2), (3) and (5) above are homologous to the node descriptions in domain semantic networks, while Predicates (4), (6) and (12) are homologous to directed arcs. Predicates (7), (8) and (9) are homologous to reasoning networks in production systems and Predicates (10), (11) and (13) may be used to define deep knowledge[2] of problem domains. It is still difficult to adopt semantic networks to represent reasoning networks and deep knowledge with the existing techniques. The above thirteen predicates have thus formed a Prolog-based representation for complex applications where both knowledge and data management is needed. Such a representation can represent any information that can be expressed in the E-R model.

Also, the representation which consists of the thirteen basic predicates describes explicitly relational schemata as well as relational tuples, thus the disadvantages of the normal method of modelling relational data bases in Prolog discussed in the introduction have been eliminated. A simplified version of the representation has been implemented in *KEshell2* [13], an *intelligent learning data base system*, which provides mechanisms for

1. translating standard (relational) data base information into a form suitable for use by its induction engines,
2. using induction techniques to extract rules from data bases, and
3. interpreting the rules produced to solve users' problems.

[2] In contrast to shallow knowledge (which is directly used for problem solving) in knowledge bases in expert systems, deep knowledge in problem domains can be used to detect inconsistencies in shallow knowledge and data.

Acknowledgements

The work presented in this paper was supported in part by the National Natural Science Foundation of China under Grant No. 68975025 when the author was the grant holder and project head of the 68975025 project and is supported in part by the ORS Award of the United Kingdom and a University of Edinburgh Research Scholarship. The author would like to express warm thanks to Dave Robertson, Robert Rae, and Peter Ross for their useful comments and advice on the work and/or this paper.

References

[1] E. Bertino and L. Martino, *Object-Oriented Database-Management Systems - Concepts and Issues.* Computer 24, 4: 33-47 (1991).

[2] R.G.G. Cattell et al., *Next Generation Database Systems.* Communications of the ACM (special section) 34, No. 10 (1991).

[3] C. Draxler, *Accessing Relational and NF^2 Databases Through Database Set Predicates.* In Pre-conference Proceedings of the 3rd Annual Conference of Association for Logic Programming (UK), Edinburgh, 86–92 (1991).

[4] T. Kazic, E. Lusk, R. Olson, R. Overbeek and S. Tuecke, *Prototyping Databases in Prolog.* In The Practice of Prolog, edited L.S. Sterling, 1–29. The MIT Press (1990).

[5] D. Li, *A Prolog Database System.* Research Studies Press, Letchworth, UK (1984).

[6] T. Nieme and K. Järvelin, *Prolog-Based Meta Rules for Relational Database Representation and Manipulation.* IEEE Transactions on Software Engineering 17, 762–788 (1991).

[7] S.Y.W. Su and D.H. Lo, *A Semantic Association Model for Conceptual Database Design.* In Entity-Relationship Approach to System Analysis and Design, edited P.P. Chen. North-Holand Publishing Company (1980).

[8] J.D. Ullman, *Principles of Database Systems.* Computer Science Press (1982).

[9] J.D. Ullman, *Principles of Database and Knowledge-Based Systems, Vol. 2: The New Technologies.* Computer Science Press (1989).

[10] X. Wu, *A Study on Intelligent Data Base Techniques.* In Proceedings of the First Chinese Joint Conference on Artificial Intelligence, 23–30, Jilin, China (1990).

[11] X. Wu, *Constructing Expert Systems.* Press of the University of Science and Technology of China, Hefei, China (1990).

[12] X. Wu, *KEshell: A "Rule Skeleton + Rule Body" Based Knowledge Engineering Shell.* In Applications of Artificial Intelligence IX (Proceedings of SPIE 1468), edited M.M. Trivedi, 632–639. SPIE Society, Bellingham, USA (1991).

[13] X. Wu, *KEshell2: An Intelligent Learning Data Base System.* In Research and Development in Expert Systems IX, edited M.A. Bramer and R.W. Milne, 253–272. Cambridge University Press (1992).

[14] X. Wu, *A Frame Based Architecture for Information Integration in CIMS.* Journal of Computer Science and Technology 7, 328–332 (1992).

[15] X. Wu, *Knowledge Acquisition from Data Bases, Ph.D. Thesis.* Dept. of Artificial Intelligence, University of Edinburgh, Scotland (1993).

[16] X. Wu and D. Zhang, *A Study of Knowledge Types.* Tech. Report NNSFC-HUT-CS-68975025-90-1, Hefei University of Technology, China (1990). Also Journal of Systems Engineering (in Chinese), to appear.

[17] X. Wu and D. Zhang, *An Approach to Generation of Semantic Network from Relational Database Schema.* Chinese Science Bulletin (English edition) 36, 1222–1225 (1991).

[18] C. Zaniolo, *Prolog: A Database Query Language for All Seasons.* In Proceedings of the 1st International Workshop on Expert Database Systems, edited L. Kerschberg, 219–232 (1986).

WALKING VIABILITY AND GAIT SYNTHESIS FOR A NOVEL CLASS OF DYNAMICALLY-SIMPLE BIPEDS

Jon Kieffer and Ramesh Bale
Interdisciplinary Engineering Program
Australian National University

Keywords: bipeds, equations, simple dynamics

Edited by: Jadran Lenarčič
Received: June 15, 1993 **Revised:** July 12, 1993 **Accepted:** July 20, 1993

This paper introduces a class of three-link, two-motor, planar bipeds that have mass centers invariantly-fixed at the hip axis and bodies that serve as reaction wheels. The principle advantage of these bipeds is that they are governed by exceptionally simple dynamic equations. This paper derives the governing equations for single-leg support and support leg transfer as well as step-to-step boundary conditions for periodic walking. Closed-form periodic gait trajectories are synthesized which ensure that the body's spin does not build up in the course of periodic walking. Examples show that a realistic model can walk on both flat and inclined surfaces.

Nomenclature

F_n, F_t:	Normal and tangential foot forces.
g:	Acceleration due to gravity.
J_b, J_ℓ:	Mass moments of inertia of leg and body about hip.
J_0:	$J_0 = J_\ell + m\ell^2$ = Effective mass moment of inertia of support leg about point of ground contact.
ℓ:	Leg length.
L:	Step length.
m:	Total biped mass (body, 2 legs).
N_F:	Number of feet per leg.
T:	Step period.
$x, \dot{x}, \ddot{x}$:	Horizontal hip position, velocity, and acceleration.
$y, \dot{y}, \ddot{y}$:	Vertical hip position, velocity, and acceleration.
α:	Slope of ground incline.
γ:	Stance angle with respect to ground normal.
$\theta_i, \dot{\theta}_i, \ddot{\theta}_i$:	Position, velocity, and acceleration of joint i ϵ(0,1,2).
μ:	Coefficient of friction between support foot and ground.
τ_i:	Torque at joint i ϵ(0,1,2).
$\psi_i, \dot{\psi}_i, \ddot{\psi}_i$:	Absolute angular position, velocity, and acceleration of link i ϵ(0,1,2) with respect to vertical.

1 Introduction

For biped walking machines built to date [1-7], control complexity increases with the number of links and generality of mass distributions. Most successful machines rely on non-anthropomorphic simplifications to reduce dynamic complexity, and/or controls that ignore high order dynamics. The simple dynamics of Raibert's hopping/running machines [8] allowed him and his coworkers to develop elegant controls and investigate fundamental aspects of legged locomotion with minimal control complexity.

This paper introduces a new class of bipeds that are governed by exceptionally simple dynamic equations and lays the groundwork for their experimental development. Walking machines of this class are expected to offer advantages in terms of control tractability and reduced cost.

We begin in section 2 by describing the proposed class of bipeds and their walking gaits. Section 3 provides the governing equations of motion and inequality constraints. Section 4 gives the conditions for periodic walking. Section 5 derives a closed-form solution to the problem of synthesizing periodic gait trajectories that do not cause the body (behaving as a reaction wheel) to spin faster with each step. Section 6 gives two examples of gait trajectories for a realistic model with

commercial actuators. Section 7 concludes the paper.

2 Machine

2.1 Physical Description

The proposed bipeds are notionally planar, but can be realized in three dimensions by implementations that are stabilized within the plane by passive means such as Raibert's tethering scheme [8], or sufficiently-wide feet. Figure 1 shows two versions of the proposed biped that have sufficiently-wide feet. Both versions are composed of three rigid links (a body, plus two identical legs) that are serially-interconnected at the hip by coaxial revolute joints that are actuated by electric motors. The version shown in Fig.1(a) has one foot per leg, but versions with two feet per leg (Fig.1(b)), or more, are also possible. Several features distinguish the proposed class of bipeds: (1) kneeless planar structure with only three links and two motors, (2) small feet that may be modelled as points of contact in the plane of motion, and (3) mass centers of each link coincident with hip axis h-h. The third feature makes the composite mass center configuration-invariant which simplifies the dynamic equations dramatically.

2.2 Walking Gaits

Without knees, or equivalent mechanisms for shortening the legs, the proposed bipeds will stub their toes if standard anthropomorphic gaits are used. In addition, such gaits would cause "sufficiently-wide" feet to interfere with each other. For these reasons, two alternate gaits shown in Fig.2 are proposed.

Figure 2(a) illustrates the so-called *wheel gait* in which each leg rotates like a wheel and the next support foot is always placed ahead of the current support foot in the direction of travel. If legs are multifooted, then the wheel gait will cycle through the feet on each leg, using each in turn for support.

The so-called *inchworm gait* (Fig.2(b)) uses the same two feet for support regardless of the number of feet on each leg and maintains the ordering of these feet with respect to the ground. It is composed of alternating steps of two types: one that expands the distance between support feet, and another that contracts the distance.

This paper will only consider *singular gaits*, i.e., gaits that have no dual-support phase. Therefore walking consists of single-support phases separated by instantaneous exchanges of support feet.

3 Governing Equations and Constraints

3.1 Single-Support Phase Dynamics

Dynamic equations for the single-support phase were derived in two-steps. Firstly, equations (1)-(3) were derived by Lagrange's approach using the notation shown in Fig.3(a).

$$\tau_0 = J_0\ddot{\theta}_0 - mg\ell\sin\theta_0 + J_b(\ddot{\theta}_0+\ddot{\theta}_1) + J_\ell(\ddot{\theta}_0+\ddot{\theta}_1+\ddot{\theta}_2) \quad (1)$$

$$\tau_1 = J_b(\ddot{\theta}_0 + \ddot{\theta}_1) + J_\ell(\ddot{\theta}_0 + \ddot{\theta}_1 + \ddot{\theta}_2) \quad (2)$$

$$\tau_2 = J_\ell(\ddot{\theta}_0 + \ddot{\theta}_1 + \ddot{\theta}_2) \quad (3)$$

Then torque, τ_0, at the point of foot contact was set to zero, and relative coordinates $(\theta_0, \theta_1, \theta_2)$ were changed to absolute coordinates(ψ_0, ψ_1, ψ_2)(Fig.3(b)).

$$\ddot{\psi}_0 = \frac{mg\ell}{J_0}\sin\psi_0 - \frac{1}{J_0}\tau_1 \quad (4)$$

$$\ddot{\psi}_1 = \frac{1}{J_b}(\tau_1 - \tau_2) \quad (5)$$

$$\ddot{\psi}_2 = \frac{1}{J_\ell}\tau_2 \quad (6)$$

The resulting system of equations (4)-(6) are decoupled. Equations (5) and (6) are linear. Nonlinear equation (4) is the same as an inverted pendulum despite the fact that control torque τ_1 is applied at the tip, rather than the base, of the pendulum! The system has three degrees of freedom, but only two inputs.

3.2 Inelastic Impact and Support Transfer

Let point p (point q) designate the releasing support foot (engaging support foot) at the instant of support transfer. Assume that point q on the

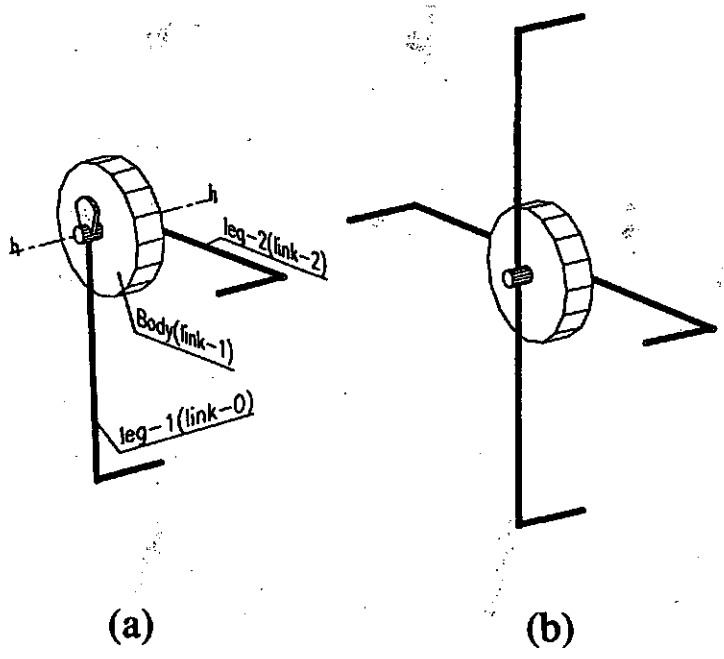


Figure 1: Biped Machines: (a) N_F=1 foot per leg, and (b) N_F=2 feet per leg.

swing leg collides inelastically with the ground at the moment of impact. This means that point q (point p) w hich was free (fixed) before impact becomes fixed (free) after impact. The following momentums are conserved at the instant of impact.

$H_p^{(h)}$= angular momentum of the releasing leg (containing foot p) about the hip point h.

$H_{pb}^{(h)}$= combined angular momentum of the releasing leg and the body about h.

H_{pbq}^{q}= combined angular momentum of all three links about the collision point q.

Conservation of these momentums provide the following impact relations which relate velocities before (-) and after (+) impact.

$$\begin{bmatrix} \dot{\psi}_0 \\ \dot{\psi}_1 \\ \dot{\psi}_2 \end{bmatrix}_p^+ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ C_1(\psi_p^{\pm}) & 0 & C_2 \end{bmatrix} \begin{bmatrix} \dot{\psi}_0 \\ \dot{\psi}_1 \\ \dot{\psi}_2 \end{bmatrix}_p^- \quad (7)$$

where

$$C_1(\psi_p^{\pm}) = -m\ell^2 \cos(\psi_2 - \psi_0)_p^{\pm}/J_0 \quad (8)$$

$$C_2 = J_\ell/J_0 \quad (9)$$

Here, subscripts p indicate that coordinates $\psi_p = (\psi_0, \psi_1, \psi_2)_p$ are referenced to foot p (the support foot before impact), and superscript $(\pm)$ indicates that $\psi_p^{\pm}$ is evaluated at the moment of impact.

After support transfer, point q becomes the support foot. Coordinates ψ_q referenced to point q are related to coordinates ψ_p as follows.

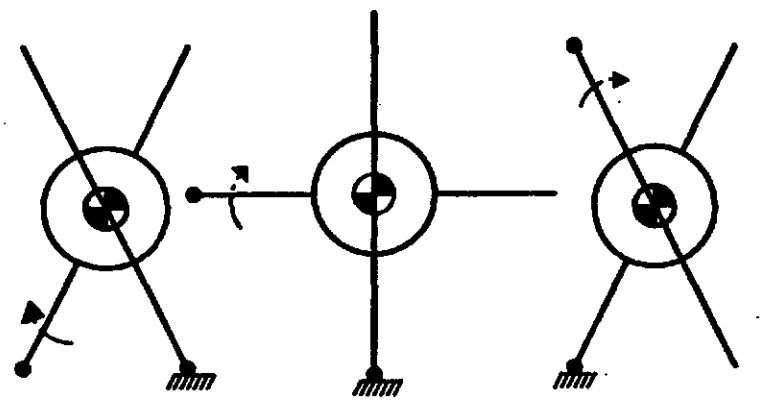
$$\begin{bmatrix} \psi_0 \\ \psi_1 \\ \psi_2 \end{bmatrix}_q = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \psi_0 \\ \psi_1 \\ \psi_2 \end{bmatrix}_p + \begin{bmatrix} \pi \\ 0 \\ \pi \end{bmatrix} \quad (10)$$

Either coordinate system, ψ_p or ψ_q, can describe the configuration of links, regardless of which foot, p or q, currently supports the biped.

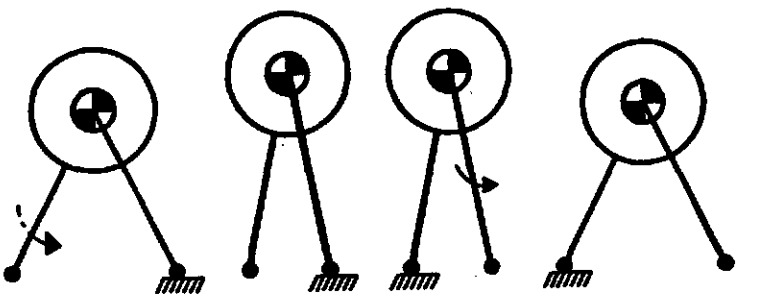
Let p designate the support foot for step k and q designate the support foot for step k+1. Then equation (7) and differentiated equation (10) combine to relate step k final velocities (before impact) to step k+1 initial velocities (after impact).

$$\begin{bmatrix} \dot{\psi}_0 \\ \dot{\psi}_1 \\ \dot{\psi}_2 \end{bmatrix}_{k+1}^{i+} = \begin{bmatrix} C_1(\psi_k^f) & 0 & C_2 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\psi}_0 \\ \dot{\psi}_1 \\ \dot{\psi}_2 \end{bmatrix}_k^{f-} \quad (11)$$

Here subscripts p and q have been dropped and



(a)



(b)

Figure 2: Walking Gaits: (a) wheel gait for biped with N_F=2, and (b) inchworm gait for biped with N_F=1.

it is understood that coordinates ψ_k are referenced to the support foot of step k.

3.3 Foot Force Constraints

The preceding equations are only valid if the support foot remains fixed to the surface. Lifting and slipping will be avoided if the following constraints are satisfied.

$$F_n > 0 \tag{12}$$

$$|F_t| < \mu F_n \tag{13}$$

Here μ represents the coefficient of static friction between the support foot and the surface and F_n and F_t are the normal and tangential foot force components which can be expressed as functions of mass center acceleration and slope incline α, as follows.

$$\begin{bmatrix} F_t \\ F_n \end{bmatrix} = m \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y}+g \end{bmatrix} \tag{14}$$

where

$$x = \ell \sin(\psi_0) \tag{15}$$

$$y = \ell \cos(\psi_0) \tag{16}$$

3.4 Other Constraints

The body and the swing leg must not interfere with the ground or with the stance leg. Actuator torque and velocity limits must be respected.

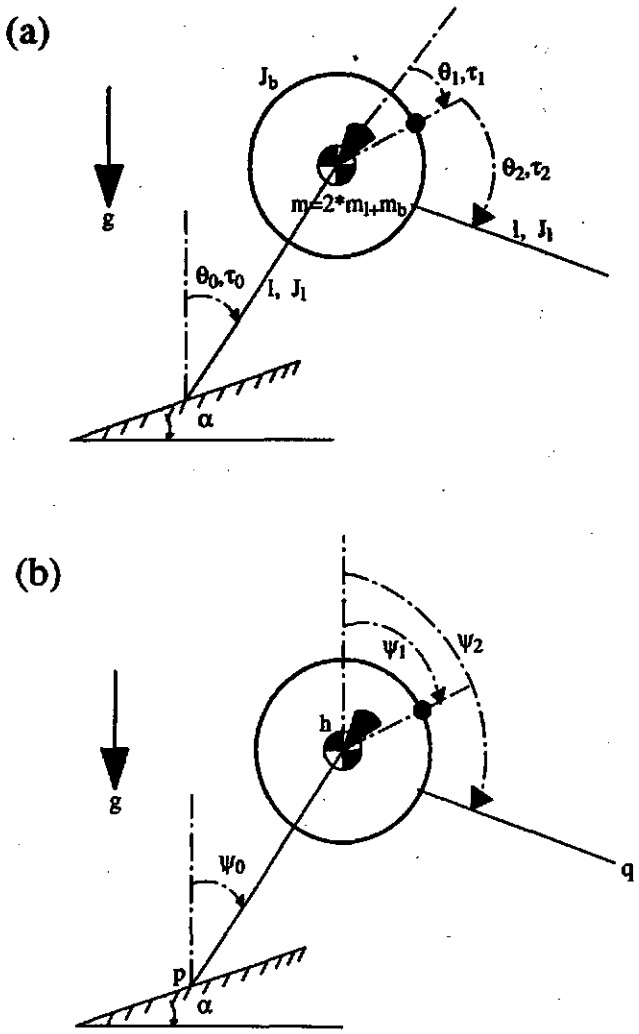


Figure 3: Notation: (a) joint coordinates (θ_0,θ_1, θ_2), (b) absolute coordinates (ψ_0,ψ_1,ψ_2).

4 Conditions for Periodic Wheel Gait Walking

The rest of this paper focuses on proving the viability of periodic wheel-gait walking. This section determines the initial and final conditions for the single-support phase.

For strict periodicity, the functions $\psi_0(t)$, $\psi_1(t)$, and $\psi_2(t)$ should each repeat with every step. However, because the body's position $\psi_1(t)$ has no influence on any governing equations, we choose to ignore it. Similarly, the body's velocity $\dot{\psi}_1(t)$ has no influence on $\psi_0(t)$ or $\psi_2(t)$. But, in contrast to $\psi_1(t)$, it's periodicity cannot be ignored because the body must not be allowed to build up enough speed to violate motor speed limits.

Thus, $\psi_0(t)$, $\dot{\psi}_1(t)$, and $\psi_2(t)$ are required to be step-wise periodic, but we will ignore periodicity of $\psi_1(t)$.

4.1 Initial and Final Positions

For steps of uniform length L, initial and final values of ψ_0 and ψ_2 can be determined as follows.

$$\psi_0^i = -\gamma - \alpha \tag{17}$$

$$\psi_0^f = \gamma - \alpha \tag{18}$$

$$\psi_2^i = \gamma - \alpha + \pi - \frac{2\pi}{N_F} \tag{19}$$

$$\psi_2^f = -\gamma - \alpha + \pi \tag{20}$$

where:

$$\gamma = \arcsin\left(\frac{L}{2\ell}\right)$$
$$(\gamma < \frac{\pi}{N_F} \text{to avoid ground interference}) \tag{21}$$

4.2 Initial and Final Velocities

For step-to-step periodicity, the initial velocities of each step must match, i.e. $\dot{\psi}_k^i = \dot{\psi}_{k+1}^i$. Using support transfer equation (11), this condition translates into the following relation between initial and final velocities of the single-support phase.

$$\begin{bmatrix} \dot{\psi}_0 \\ \dot{\psi}_1 \\ \dot{\psi}_2 \end{bmatrix}^f = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ D_2 & 0 & D_1 \end{bmatrix} \begin{bmatrix} \dot{\psi}_0 \\ \dot{\psi}_1 \\ \dot{\psi}_2 \end{bmatrix}^i \tag{22}$$

where

$$D_1 = -C_1(\psi^f)/C_2 = -m\ell^2\cos(2\gamma)/J_\ell \tag{23}$$

$$D_2 = 1/C_2 = J_0/J_\ell \tag{24}$$

5 Closed Form Solution for Periodic Wheel Gait Trajectories

Just as humans can walk in a variety of ways, bipeds usually have many feasible gait trajectories. The objective here is not to explore all possibilities or to find an optimal solution. It is only to prove that periodic walking is feasible by deriving example gait trajectories that satisfies all governing equations.

The problem of gait synthesis would be trivial, were it not for the fact that the system has more degrees of freedom (ψ_0, ψ_1, ψ_2) than inputs (τ_1, τ_2). We will solve the problem in two steps. First we will derive closed-form trajectories $\psi_0(t)$, $\psi_1(t)$, and $\psi_2(t)$ that satisfy dynamic equations (4)-(6), as well as the boundary conditions (17)-(20), and (22). Then, in section 6, we will simulate examples that satisfy foot force, ground interference, and actuator constraints.

5.1 Synthesis of ψ_0(t)

Let the single support phase begin at t=0 and have duration T. Then equations (4)-(6) can be integrated over the interval [0,T] to obtain the following equations.

$$J_0\Delta\dot{\psi}_0 = mg\int_0^T \ell\sin(\psi_0)dt - \int_0^T \tau_1 dt \quad (25)$$

$$J_b\Delta\dot{\psi}_1 = \int_0^T \tau_1 dt - \int_0^T \tau_2 dt \quad (26)$$

$$J_\ell\Delta\dot{\psi}_2 = \int_0^T \tau_2 dt \quad (27)$$

Here $\Delta\dot{\psi}_j = \dot{\psi}_j^f - \dot{\psi}_j^i$; j=0,1,2. With substitutions from (26) and (27), equation (25) can be expressed as follows.

$$mg\int_0^T \ell\sin\psi_0 dt = J_0\Delta\dot{\psi}_0 + J_b\Delta\dot{\psi}_1 + J_\ell\Delta\dot{\psi}_2 \quad (28)$$

Equation (28) imposes conditions on ψ_0(t) that can be further clarified if $\Delta\dot{\psi}_0$, $\Delta\dot{\psi}_1$, and $\Delta\dot{\psi}_2$ can be expressed as functions of $\dot{\psi}_0^i$ and $\dot{\psi}_0^f$ only. Such functions, shown below, can be derived from equation (22).

$$\Delta\dot{\psi}_0 = \dot{\psi}_0^f - \dot{\psi}_0^i \quad (29)$$

$$\Delta\dot{\psi}_1 = 0 \quad (30)$$

$$\Delta\dot{\psi}_2 = D_2\dot{\psi}_0^i + (D_1 - 1)\dot{\psi}_0^f \quad (31)$$

Substitution of (29)-(31) into (28) and simplification using (23) and (24) provides the following condition which ψ_0(t) must satisfy in addition to $\psi_0(0) = \psi_0^i$ and $\psi_0(T) = \psi_0^f$

$$\int_0^T \ell\sin\psi_0 dt = \frac{\ell^2}{g}[1 - \cos(2\gamma)]\dot{\psi}_0^f \quad (32)$$

To simplify synthesis of ψ_0(t), we change variables,

$$x = \ell\sin(\psi_0) \quad (33)$$

and represent equation (32) as follows.

$$\int_0^T x(t)dt = \frac{\ell[1 - \cos(2\gamma)]}{g\cos(\gamma - \alpha)}\dot{x}^f \quad (34)$$

Here, $\dot{x}^f = \ell\cos(\psi_0^f)\dot{\psi}_0^f$.

The problem is now to synthesize a function x(t) which satisfies equation (34) as well as the following boundary conditions derived from equations (17), (18) and (33).

$$x^i = -\ell\sin(\gamma + \alpha) \quad (35)$$

$$x^f = \ell\sin(\gamma - \alpha) \quad (36)$$

Because there are only 3 conditions, we can choose x(t) to be quadratic in t

$$x(t) = a_0 + a_1 t + a_2 t^2 \quad (37)$$

and use equations (34)-(36), which are linear in (a_0, a_1, a_2), to solve for the unknown coefficients. After back substitution, we arrive at the following final expression for ψ_0(t).

$$\psi_0(t) = \arcsin\left(\frac{a_0 + a_1 t + a_2 t^2}{\ell}\right) \quad (38)$$

where

$$a_0 = x^i \quad (39)$$

$$a_1 = \frac{2}{T}\Delta x - \dot{x}^f \tag{40}$$

$$a_2 = \frac{1}{T}\left(\dot{x}^f - \frac{1}{T}\Delta x\right) \tag{41}$$

and

$$x^i = -\ell \sin(\gamma + \alpha) \tag{42}$$

$$\Delta x = 2\ell \sin\gamma \cos\alpha \tag{43}$$

$$\dot{x}^f = \frac{(x^i + 2/3\Delta x)T}{\frac{l}{g}\frac{[1-\cos(2\gamma)]}{\cos(\gamma-\alpha)} + \frac{T^2}{6}} \tag{44}$$

5.2 Synthesis of $\psi_2(t)$

Once $\psi_0(t)$ has been determined, synthesis of $\psi_2(t)$ is straight forward. There are 4 boundary conditions for $\psi_2(t)$: initial and final positions given by (19) and (20),and the following initial and final velocities derived from (22) considering that $\psi_0(t)$ has been determined.

$$\dot{\psi}_2^i = \dot{\psi}_0^f \tag{45}$$

$$\dot{\psi}_2^f = D_2\dot{\psi}_0^i + D_1\dot{\psi}_0^f \tag{46}$$

Here the values of $\dot{\psi}_0^i$ and $\dot{\psi}_0^f$, depend on the previously-synthesized trajectory $\psi_0(t)$. For trajectories synthesized according to equation (38), the following equations apply.

$$\dot{\psi}_0^i = \frac{a_1}{[\ell\cos(\psi_0^i)]} \tag{47}$$

$$\dot{\psi}_0^f = \frac{\dot{x}^f}{\left[\ell\cos(\psi_0^f)\right]} \tag{48}$$

Where a_1, $\dot{x}^f$ and ψ_0^f are determined from (40),(44) and (18), respectively.

We choose $\psi_2(t)$ to be cubic in t and use the four boundary conditions to determine the unknown coefficients. The results can be expressed as follows.

$$\psi_2(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 \tag{49}$$

where

$$b_0 = \psi_2^i \tag{50}$$

$$b_1 = \dot{\psi}_2^i \tag{51}$$

$$b_2 = \frac{\left[3(\psi_2^f - \psi_2^i) - T(2\dot{\psi}_2^i + \dot{\psi}_2^f)\right]}{T^2} \tag{52}$$

$$b_3 = \frac{\left[-2(\psi_2^f - \psi_2^i) + T(\dot{\psi}_2^i + \dot{\psi}_2^f)\right]}{T^3} \tag{53}$$

Here: ψ_2^i, ψ_2^f, $\dot{\psi}_2^i$ and $\dot{\psi}_2^f$ are determined from equations (19), (20), (45), and (46), respectively.

5.3 Determination of $\psi_1(t)$

Once $\psi_0(t)$ and $\psi_2(t)$ have been determined, $\psi_1(t)$ can be evaluated by integrating the following equation, derived from equations (4)-(6) and (33), twice.

$$\ddot{\psi}_1(t) = \frac{mg}{J_b}x(t) - \frac{J_0}{J_b}\ddot{\psi}_0(t) - \frac{J_\ell}{J_b}\ddot{\psi}_2(t) \tag{54}$$

Using equation (37), the results of each integration are shown below.

$$\begin{aligned}\dot{\psi}_1(t) &= \frac{mg}{J_b}\left[a_0 t + \frac{a_1}{2}t^2 + \frac{a_2}{3}t^3\right] \\ &\quad -\frac{J_0}{J_b}\left[\dot{\psi}_0(t) - \dot{\psi}_0^i\right] \\ &\quad -\frac{J_\ell}{J_b}\left[\dot{\psi}_2(t) - \dot{\psi}_2^i\right] + \dot{\psi}_1^i\end{aligned} \tag{55}$$

$$\begin{aligned}\psi_1(t) &= \frac{mg}{J_b}\left[\frac{a_0}{2}t^2 + \frac{a_1}{6}t^3 + \frac{a_2}{12}t^4\right] \\ &\quad -\frac{J_0}{J_b}\left[\psi_0(t) - \psi_0^i\right] - \\ &\quad \frac{J_\ell}{J_b}\left[\psi_2(t) - \psi_2^i\right] + \psi_1^i + \\ &\quad \left[\frac{J_0}{J_b}\dot{\psi}_0^i + \frac{J_\ell}{J_b}\dot{\psi}_2^i + \dot{\psi}_1^i\right] t\end{aligned} \tag{56}$$

Expressions (54)-(56) can be considered closed-form functions of t based on the following relations.

number of feet per leg	$N_F = 1.$
leg length	$\ell = 0.16$ m.
total biped mass	m = 0.75 kg.
leg mass moment of inertia	$J_\ell = 3.9\times 10^{-3}$ kg.m^2.
body mass moment of inertia	$J_b = 2.8\times 10^{-3}$ kg.m^2.

Table 1: Model Design Parameters.

motor plus gearbox mass	$m_{motor} = 0.09$ kg.
rated output torque	$\tau_{rated} = 0.18$ N.m.
maximum output torque	$\tau_{max} = 0.37$ N.m.
rated output speed	$\omega_{rated} = 9.3$ rad/s.
maximum output speed	$\omega_{max} = 18.4$ rad/s.

Table 2: Actuator Output Specifications based on Harmonic $Drive^{TM}$ model RH-5 servo actuator with a 50:1 reduction ratio.

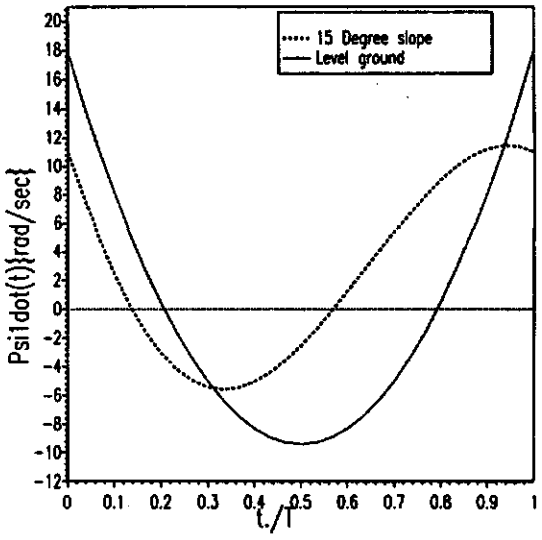


Figure 8: Body Spin Rate Trajectory $\dot{\psi}_1(t)$.

$$\dot{\psi}_0(t) = \frac{\dot{x}(t)}{\ell \cos[\psi_0(t)]}$$

$$\dot{\psi}_2(t) = b_1 + 2b_2 t + 3b_3 t^2 \qquad (57)$$

$$\ddot{\psi}_0(t) = \frac{\ddot{x}(t) + x(t)\dot{\psi}_0^2(t)}{\ell \cos[\psi_0(t)]}$$

$$\ddot{\psi}_2(t) = 2b_2 + 6b_3 t \qquad (58)$$

Here x(t) and ψ_0(t) are given by (37) and (38), and

$$\dot{x}(t) = a_1 + 2a_2 t \qquad (59)$$

$$\ddot{x}(t) = 2a_2 \qquad (60)$$

6 Examples

In this section we apply examples of the derived gait trajectories to a realistic model. The model's design parameters, shown in Table 1, were chosen to suit the commercial actuator specified in Table 2. Two gait trajectories were considered: one for walking on a level surface using step length L=0.05m and step period T=0.9s, and the other for walking up an incline $\alpha = 15°$ with step length L=0.10m, and step period T=0.75s.

From Figs. 4(a) and 4(b), which plot feet and hip paths and intermediate positions of the biped, it is clear that the feet and body do not interfere with each other or the ground for either example. Figures 5-8 plot various trajectories for both examples. In these figures, solid-line plots correspond to the level-surface walking and the dashed-line plots correspond to sloped-surface walking. Figure 5 shows plots of foot force trajectories which verify that the support foot stays in contact with the ground and will not slip if the friction coefficient, μ, is greater than 0.18. Figures 6 and 7 show plots of actuator velocities and torques respectively, which verify that actuator performance limits (Table 2) are respected. The plots of $\dot{\psi}_1$(t) in Fig. 8, verify that the body's angular velocity is step-wise periodic and will not build-up over time.

It is clear that these examples satisfy all conditions for periodic walking, but they also come close to violating actuator performance specifications (Table 2). In particular, speed limits are nearly violated by the level walking example, and torque limits are nearly violated by sloped walking example. This seems to indicate that walking may be only marginally viable, but other factors must also be considered: (1) No attempt was made to optimize actuator selection, biped design,

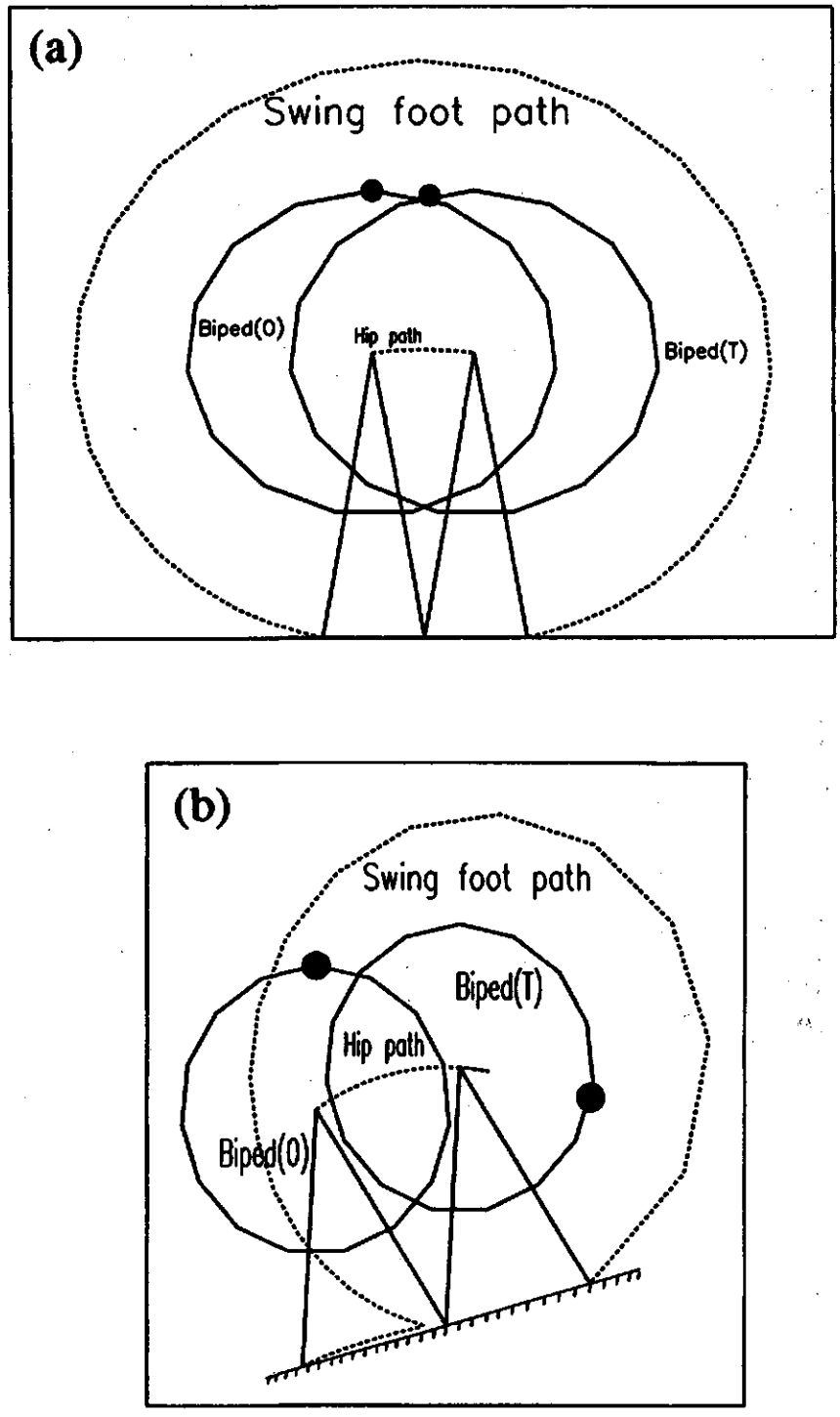


Figure 4: Feet and Hip Paths for a Single Periodic Walking Step: (a) on level ground, (b) up a $\alpha = 15°$ incline.

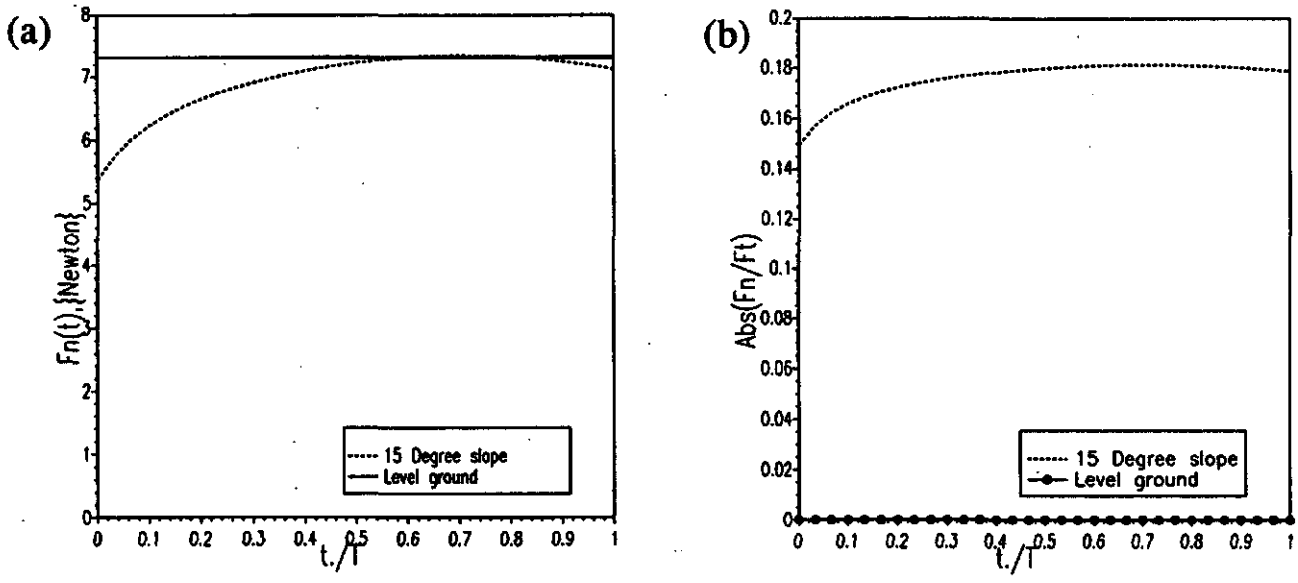


Figure 5: Foot Force Trajectories: (a) normal force $F_n(t)$, (b) required friction coefficient $\mu(t) = |F_t|/F_n$

step length, or period -each was obtained by trial and error. (2) No attempt was made to optimize gait trajectories -the choice of low order polynomials for trajectory equations (38) and (49) was based on mathematical convenience, rather than physics. Considering these factors, we expect that substantial improvement will follow from systematic investigation of them.

7 Conclusion

This study has introduced a class of dynamically-simple bipeds and has laid the groundwork for their implementation. In particular, it has provided three main results: (1) derivation of governing dynamic equations, constraints, and conditions for periodic walking, (2) analytic solution to the problem of gait synthesis for periodic walking (3) realistic examples providing evidence that such machines can be successfully implemented using existing commercial actuators.

References

[1] Miura, H., and Shimoyama, I.: Dynamic Walk of a Biped, *Int. J. Robotics Research*, 3(2):1984.

[2] Mita, T., Yamaguchi,T., Kashiwase, T., and Kawase, T.: Realization of High Speed Biped Using Modern Control Theory, *Int. J. Control*, 40(1):107-119, 1984

[3] Katoh, R., and Mori, M.: Control Method of Biped Locomotion Giving Asymptotic Stability of Trajectory, *Automatica*, 20(4):405-414,1984.

[4] Furusho, J., and Masubuchi, M.: Control of a Dynamical Biped Locomotion System for Steady Walking, *J. Dynamic Systems, Measurement, and Control*, 108:111-118, 1986.

[5] Furusho, J. and Sano, A.: Sensor-Based Control of a Nine-Link Biped, *Int. J. Robotics Research*, 9(2):83-98, 1990.

[6] Zheng, Y.F.: Acceleration Compensation for Biped Robots to Reject External Disturbances, *IEEE Trans. Syst., Man, and Cyber.*, 19(1):74-84,1989.

[7] Kajita, S., Yamaura, T., and Kobayashi, A.: Dynamic Walking Control of a Biped Robot Along a Potential Energy Conserving Orbit, *IEEE Trans. Robotics and Automation.* 8(4):431-438, 1992.

[8] Raibert, M.,H.: *Legged Robots That Balance*, MIT Press, 1986.

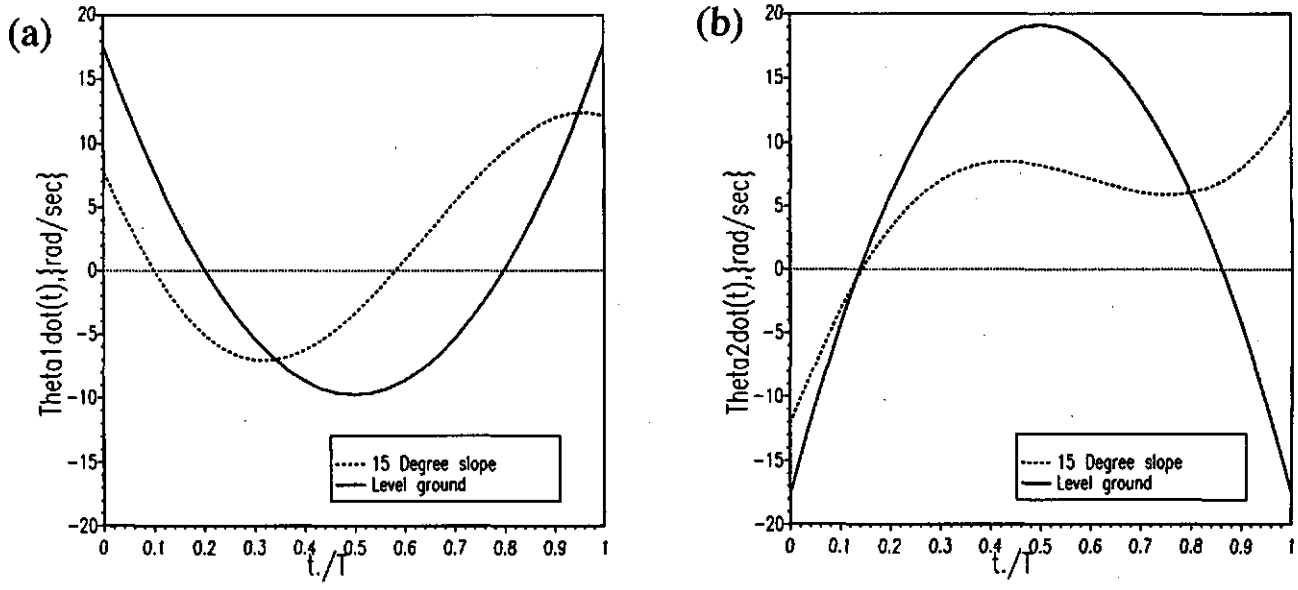


Figure 6: Joint Velocity Trajectories: (a) stance leg hip joint velocity $\dot{theta}_1(t)$, (b) swing leg hip joint velocity $\dot{theta}_2(t)$.

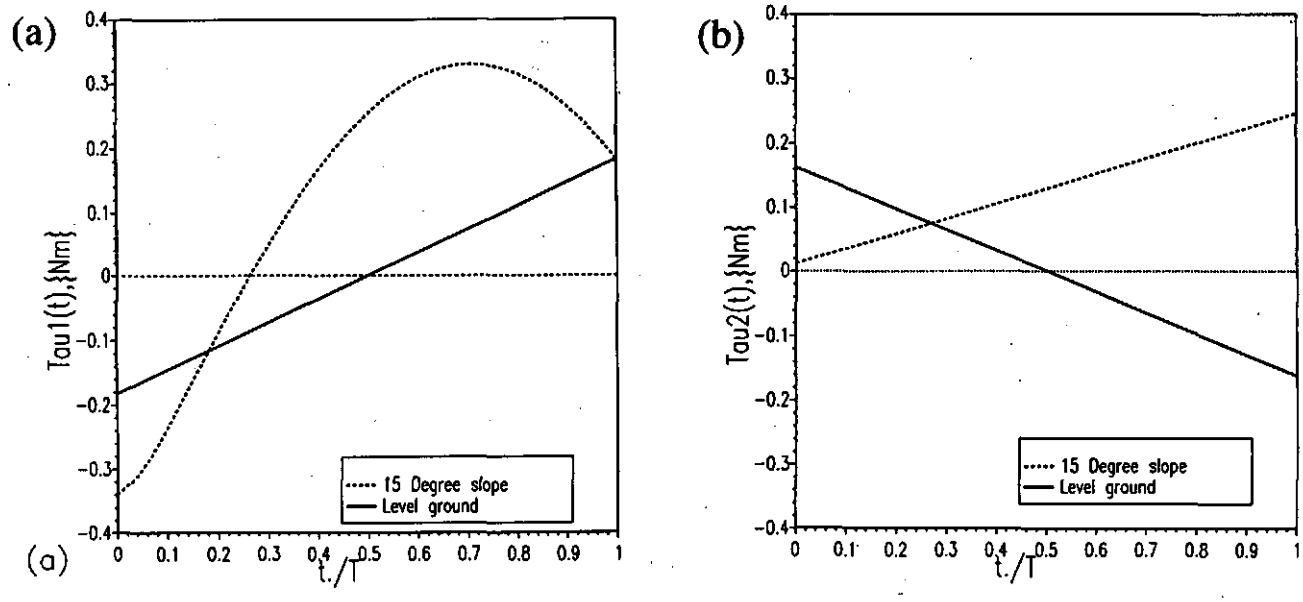


Figure 7: Joint Torque Trajectories: (a) stance leg torque $\tau_1(t)$, and (b) swing leg torque $\tau_2(t)$.

MODELLING BIODEGRADATION BY AN EXAMPLE-BASED LEARNING SYSTEM

Dragan Gamberger, Sanja Sekušak, Aleksandar Sabljić
Rudjer Bošković Insitute, P.O.B.1016
41001 Zagreb, Croatia

Keywords: inductive learning, biodegradation

Edited by: Matjaž Gams
Received: April 19, 1993 **Revised:** June 8, 1993 **Accepted:** July 9, 1993

In this paper a novel rule-generation system for learning from examples and its application for modelling biodegradation of chemicals are presented. Two rules for biodegradation prediction are generated: the first one for all binary descriptors and a learning set of 48 examples, and the second one with some descriptors extended to integer and floating-point values and a learning set of 160 examples. The results of prediction of test examples by the generated rules are compared with the measured values and the results of two known models: classical fitting model, based on molecular connectivity indices, and a neural network model. Besides good prediction results, the generated rules have the unique characteristic of pointing out some logical dependencies that might influence the better understanding of the biodegradation process.

1 Introduction

Ultimate biodegradation of commercial chemicals in natural water, soil and sediment is a complex process, i.e. a sequence of processes which is not yet well understood. It is also one of the key factors in evaluating the environmental fate and possible adverse effects of commercial chemicals, as well as in the exposure-assessment process. Thus the ability to measure or estimate the biodegradation potential of organic chemicals, in semi-quantitative or qualitative terms, is of crucial importance for the environmental risk assessment of commercial chemicals required by environmental laws and regulations in many countries [1]. The results from a batch of laboratory test procedures are used to evaluate the ultimate biodegradability of organic chemicals in the environment. Unfortunately, those laboratory procedures are quite lengthy and sometimes of questionable accuracy [2]. Thus, in many instances expert opinion is sought in addition to the laboratory data. Furthermore, it is not feasible to perform biodegradation tests either for the large number of chemicals in everyday use or even for the High Production Volume (HPV) chemicals [3].

Considerable efforts have been made to develop models that will reliably, in semi-quantitative or qualitative terms, estimate ultimate biodegradability from chemical structure [4], [5], [6], [7], [8], [9]. Unfortunately, only a few of those models can be generally applied to various chemical classes. Quantitative structure-activity relationship (QSAR) models are based either on a group contribution method [8], [9] or on molecular connectivity indices combined with a set of simple rules [4]. Quite recently, neural network (NN) example-based learning methods [5] have also been used to model the ultimate biodegradability of organic chemicals.

In this paper, a novel inductive machine learning method will be presented and its applicability will be tested on modelling the ultimate biodegradation of organic chemicals. The example-based learning method, which includes some elements of the so-called first order inductive learning systems [13], will be applied to two training (learning) sets of biodegradation data in order to develop rules, based on the structural characteristics of chemicals, for estimating their biodegradability. The research of chemical properties by example-based learning methods is very promising, since there

is usually a relatively small set of chemicals for which the property of interest is known and which can be used for learning, and an enormous number of chemicals for which this property should be predicted. In addition, the incomplete understanding of the biodegradation phenomena, even by experts, stimulates at the present time research on the application of example-based learning.

The first learning set consists of biodegradability data which are the result of a survey of expert opinion, conducted by the U.S. Environmental Protection Agency [2], on ultimate biodegradability for a set of complex and structurally diverse chemicals. Apart from the experts' opinions, no experimental biodegradation data is available for this set of 50 chemicals. The results derived from this unique learning set will also provide information on the ability of the employed AI method to simulate expert knowledge and reasoning. Biodegradation rules developed from expert knowledge will be tested on a set of experimental data. This test will allow us to evaluate the quality of the developed rule, to discover its possible limitations, and, if necessary, to formulate an improved biodegradation rule. Special attention will be paid to the problem of unreliable learning examples. Finally, all rules developed in this study will be tested on two small sets of reliable experimental data not included in the learning process. Those results will be compared with the results obtained by the classical modelling method [4] and a neural network-based learning system [5] from the same sets of biodegradation data.

The results of this study should help biodegradation experts to identify essential structural requirements influencing ultimate biodegradation and to direct future experimental research on their verification. These results should also advance human knowledge on the environmental behaviour of xenobiotic chemicals and facilitate construction of an expert system for biodegradation of commercial chemicals. In this way, non-experts in the field of biodegradation will also be able to benefit from the experts' knowledge.

2 Inductive learning system

The artificial intelligence methods applicable to ultimate biodegradability are expert systems and example-based learning [10], [11], [12]. Expert systems are potentially a good solution for such problems and other, similar examples. They can combine explicit and implicit expert knowledge, as well as the results of the example-based learning systems. Moreover, the complexity of biodegradation phenomena points to the need for the development of corresponding expert systems. However, the incomplete understanding of the biodegradation phenomena, even by experts, stimulates research into the application of example-based learning.

The well-known inductive learning systems, like AQ, ID3, CN2 and ASSISTANT, have already been successfuly applied to a few different learning problems [12], [13]. In this paper are presented the results of a novel approach to inductive learning by logical minimization. The task of the example-based learning system is to generate a rule from a given set of examples (instances). The examples are composed of input variables and an output binary variable. The number of input variables is determined by the number of descriptors (attributes) and the value of each variable equals the value or state of the corresponding descriptor. The output variable has value 1 if the example has some property or 0 otherwise. There are three types of input variable: variables of quality (represented by strings), integer variables of quantity and floating-point variables of quantity.

The generated rule is in the form of a disjunction with extracted common factors. Each element of a disjunction is a conjunction of elementary logical tests (ELT).

if (ELT1)(ELT2) ∨ (ELT3)(ELT4) ∨ (ELT5)
then OUTPUT IS 1 else OUTPUT IS 0.

ELTs are not generated by the system in the sense described in [13], but they may have only predefined forms that are automatically generated for each variable and each pair of variables of the same type. For variables of quality, they can be: an input is equal to a constant string, an input is different from a constant string, an input is equal to another input of the same type, an input is different from another input of the same type (examples: inp1=great, inp2≠small, inp1=inp3, inp2≠inp3). For integer variables of quantity, they can be: an input is equal to, different from, greater than or smaller than a constant integer, another input of the same type, or another input of the

same type shifted by a constant integer (examples: inp4=2, inp4≠3, inp4>1 inp4<5 inp4=inp5, inp4≠inp6, inp4>inp6, inp4<inp6, inp4=inp6+1, inp5≠inp6-2, inp5<inp6+3, inp4>inp5-1). For floating point variables of quantity, they can be: an input is either greater than or smaller than a floating point constant, another input of the same type, or another input of the same type shifted by a floating point constant (examples: inp7>1.2, inp8<3.14, inp7>inp9, inp8<inp9, inp7>inp8+0.2, inp8<inp9-2.46).

In this system, the generated rule depends neither on the order of the examples in the learning set nor on the manifold repetition of the same examples, and the generated rule must satisfy all learning examples. This is impossible where there are contradictory learning examples. Before starting the actual rule-generation process, the algorithm tests the learning set for duplicate examples (examples with all equal input and output values) and eliminates the copies. After that, the set is tested for contradictions (examples with all equal input values and different output value). In such cases both contradictory examples are excluded.

It is possible to generate a large number of different rules of the given form so that they satisfy all learning examples. The assumption of this and other learning systems is that the simplest one among them has the greatest chance of having the highest average percentage of correct predictions for test examples different from the learning examples [14]. The conditions for determining the simplest rule can be defined in several different ways. In the realised algorithm, in contrast to other similar inductive learning systems [12], we search in the first step for the minimal set of ELTs with which the rule can be built. In the second step, for the defined set of ELTs, we build a rule in the form of a disjunction of conjunctions, so that we minimize first the length of conjunctions and then their number.

From a set of ELTs, a rule that satisfies all learning examples can be built, if and only if, for any possible 1/0 pair from the learning set, there is at least one element in the set which covers the pair. A 1/0 pair is a pair of examples formed of an example with output value 1 and an example with output value 0. An ELT covers a 1/0 pair if it is true for the first and false for the second example from the pair.

The search for the minimal set of ELTs which satisfies the previous condition can be deterministic or heuristic. The first approach, based on an exhaustive search through all promising subsets of all ELTs, has the advantage that the generated solution is certainly the absolute minimum and that all equally good solutions can be found. The drawback is that the computation time grows exponentially with the number of examples and the number of all possible ELTs. The heuristic approach, selecting first the elements that cover 1/0 pairs of examples with the most similar descriptor values, generates a solution much faster, but it may not be optimal. In the biodegradation case, we used the heuristic approach. Deterministically generated solutions were used occasionally to control how far heuristic results are from the optimum. It was noticed that the number of equally good minimal sets of ELTs generated by this algorithm may be used as a measure of the quality of the learning set. A large number of possible solutions means either that the learning set is too small or that it includes a few examples significantly different from the rest of the set (potentially incorrect examples).

For the selected minimal set of ELTs, a rule can be formed in the following way. For each example with output 1, form a conjunction of all elements from a minimal set that are true for the example. It is obvious that, because of the described condition for the selection of elements in the minimal set, there is at least one element in the conjunction that is not satisfied for any learning example with output 0. The rule is a disjunction of different conjunctions formed for all learning examples with output 1. In the realised algorithm, we first minimize the number of elements in each conjunction so that the condition for dissatisfaction of examples with output 0 remains fulfilled. There may be more than one minimal solution for each example with output 1. In the second step, the minimal number of such conjunctions is selected in such a way that, for any example with output 1, there is at least one conjunction in which all elements are satisfied for the example.

The described algorithm assumes the correctness of all learning examples and generates a rule which satisfies them all. By the use of the heuristic algorithm, we have also built an iterative algorithm for locating potentially incorrect examples.

At the beginning, the algorithm determines the number of necessary ELTs for the whole learning set and after that the same number for learning sets in which some of the examples were omitted. If the exclusion of an example reduces the number of ELTs in the minimal set, we conclude that that example is potentially incorrect. This example is removed from the learning set and the process is repeated until a simple irreducible solution is obtained.

The presented rule-generation method can be a main part of a more general algorithm in which the output variable may be of the quality or quantity type. It is also possible to use this rule-generation method in multiple-output applications.

3 Modelling ultimate biodegradation

The basis for biodegradation modelling is a set of 50 complex and structurally diverse molecules for which a group of experts have estimated the ultimate biodegradation [2]. The measure of biodegradation is expressed by a number in the range 1-4 which represents the average value of expert responses. The value 1 implies very fast, and the value 4 very slow, biodegradation. The standard deviations of expert responses were also computed and were cited in [2]. One can see that the standard deviations are relatively large (0.40-1.06), which indicates the differences of opinion among experts and the complexity of the problem. Nevertheless, the given examples, because of their diverse structures, are considered to be more appropriate for modelling purposes than measured data which may also significantly depend on environmental parameters. Because of this, in both studies [4] and [5], the expert examples have been used as a learning set and two sets of experimental examples have been used as test sets. For this purpose, biodegradation below 2.50 was considered fast and biodegradation above 2.50 was considered slow. In both studies, examples 27 and 29 have not been used because their molecular weight is unknown. Thus the learning set was reduced to 48 examples.

In [5], 11 simple binary descriptors based on structural features appropriate for modelling biodegradation have been introduced. They are listed in Table 1, column A. For comparison purposes, we used the same set of binary descriptors in our first experiment. In the second experiment, we also tried to model biodegradation with these descriptors extended to integer variables of quantity whenever possible (column B). Descriptor "j" for molecular weight was transformed to a floating-point variable so that the real molecular weight was rounded to a multiple of 10. Descriptor names "a"-"k" are used in generated rules to denote the corresponding descriptor.

We have built two different rules for biodegradation prediction. The first one was built for binary descriptors using the same learning set as [4] and [5]. The rule generated from the whole set of 48 learning examples was built of 5 ELTs, and its results on the test sets were 19/23 and 14/17 (19 correct predictions out of 23 examples and 14 correct predictions out of 17 examples). Experiments with the elimination of some learning examples showed that when example 44 was eliminated, a rule with 4 ELTs was built. Its results on test examples were 18/23 and 16/17, which is somewhat better than with the original rule.

The test with the deterministic rule-generation algorithm showed that there is still a large number of different solutions. When we repeated the search for potentially incorrect examples with the extended learning set in which 112 additional measured examples were included, 3 new potentially incorrect examples were detected (28,40,45). We selected example 28 because its elimination meant that the deterministic algorithm generated the least number of different solutions. The number of different solutions was 34, a relatively large value. It can be interpreted as a sign that the remaining 46 examples (essentially only 33 different examples and 13 duplicates) cannot define the rule uniquely and that we cannot expect very good predictions using the generated rule. When both examples 44 and 28 were eliminated, the heuristic algorithm generated Rule 1 with 4 ELTs.

if $(c = g)(c \neq k) \vee$
$\vee(e \neq f)(c = j)[(c = g) \vee (c \neq k)]$
then BD IS FAST else BD IS SLOW.
(Rule 1)

Its results are 20/23 and 15/17 for test sets, which is better than both previous results. The predictions obtained by this rule are recorded in

input name	A - binary descriptors	B - integer descriptors
a	heterocycle N	
b	ester,amide,anhydride	
c	$\geq 2Cl$	number of Cl
d	chemicals with aliphatic fused rings	
e	chemicals only with C, H, N, O	
f	NO_2	
g	$\geq$2 cycles	number of cycles
h	epoxide	
i	primary or aromatic OH	
j	molecular weight $\geq$ 259.6g mol^{-1}	molecular weight
k	C-O bond	number of C-O bonds

Table 1: Descriptor definitions

column 3 of Tables 3-5. Further example eliminations did not result in further rule simplification.

We have also experimented with this rule on the additional set of 112 measured biodegradation data. The result was 96/112 or 86%, which matches the prediction results for the two cited test sets. We have tried to improve Rule 1 by extending the learning set with these 112 measured examples. However, of 112 measured examples, 12 were in contradiction with the learning set. It was a sign that the selected binary descriptors are not completely sufficient to determine the biodegradation property. Because of this, we tried to substitute binary descriptors by variables of quantity whenever possible (Table 1 column B), with the result that only one contradiction remained.

The second rule-generation process started with a learning set of 160 examples and with 3 integer, 1 floating-point and 7 binary descriptors. The directly generated rule for all 160 examples was rather complicated, having 16 ELTs. Its prediction results were 15/23 and 14/17. After eliminating 14 examples, which was a rather straightforward procedure, Rule 2, with only 6 ELTs, was generated.

if $(j < 180)(k \neq 0) \vee$
$\vee(e \neq f)[(g \leq k) \vee (j < 135)(j > 95)]$
then BD IS FAST else BD IS SLOW.
(Rule 2)

Of 14 eliminated examples, 5 were from the starting learning set of 48 examples. Prediction results of this rule are 22/23 and 17/17, and they are given in column 4 of Tables 3-5. The good prediction results can be explained by the large learning set including examples similar to those in the test sets, but also by the fact that the deterministic algorithm executed for the reduced set of 146 learning examples generated only 2 different possible solutions with 6 ELTs. These 2 solutions had 5 identical elements, while the sixth was $(j < 180)$ and $(j < 190)$ respectively.

It is interesting to analyse the ELTs selected for rule generation. In particular, the elements of Rule 2 offer a lot of information about the biodegradation property by themselves. At first we notice that three boundary values for molecular weight (95,135 and 180 g mol^{-1}) are selected and that none of them is 259.6 g mol^{-1}, as selected for the binary descriptor definition in [5]. At the same time, the existence of three elements with a molecular weight descriptor points to its importance for the rate of biodegradation. We can also notice the element $(k \neq 0)$. From it we conclude that the boundary value for the binary descriptor "k" was well chosen. In contrast, the element $(g \leq k)$ opens up the possibility for a novel hypothesis concerning the number of cycles and its importance for the biodegradation process. A similar situation holds also for the element $(e \neq f)$, the only one appearing in Rule 1 and in Rule 2.

4 Comparison of different models

It is relatively difficult to compare prediction results of the four mentioned models. Directly comparable are only NN model (feedforward multilayer NN with 4 neurons in the hidden layer trained by the backpropagation algorithm) and Rule 1, because they used the same learning set and the same descriptors. The QSAR model also used only 48 learning examples but it inherently includes some biodegradation expert knowledge, while Rule 2 is based on the extended learning set of 160 examples.

Table 2 summarises the prediction results for all models. From the first row, it can be seen that models eliminate or incorrectly predict some examples from the learning set. This result strongly supports the hypothesis that some of the learning examples are potentially incorrect. It is interesting to notice that the QSAR model and Rule 2, although completely different in concept, agreed that examples 28,40 and 44 are potentially incorrect.

The second and the third row of the table contain the number of correct answers for the first and the second test set respectively. The QSAR model has substantially better results for the first test set than for the second test set, while the prediction results for the NN model show exactly the opposite characteristic. Rule 1 and Rule 2 are proportionally equally good for both test sets.

In the fourth row, the percentage of good predictions for all test examples are listed. Prediction results for Rule 2 are significantly better. However, this model still needs additional verification on other test examples. Unfortunately, there are no other available biodegradation data on which this prediction may be verified. It is worth noting that three out of four models predicted for example 11 in the first test set fast degradation, in contrast to the measured slow degradation. There is a high probability that this example is incorrect. If future experimental efforts prove this assumption to be correct, it means that Rule 2 has 100% accuracy for both test sets!

The QSAR model also shows rather good prediction results and justifies the effort requireed for its development. We also notice that Rule 1 showed better results than the NN model, but we cannot in any case conclude on the general superiority of the inductive learning over the neural network approach. The advantage of the neural network approach is its flexibility and direct applicability in modelling very different problems. In contrast, the described system can use only predefined patterns of logical elements. Because of that, we expect that this system will, in the first place, be a powerful instrument for experts. The modelling process can in this case be an iterative activity in which the generated rules can, by introducing novel ideas and by pointing out the potentially incorrect examples, direct future theoretical and experimental research. Furthermore, the experts can, by inclusion of new learning examples and verification of potentially incorrect examples, influence the rule- generation process and improve its prediction results.

5 Conclusions

Modelling biodegradation is a good application to test diverse methods for example-based learning. It has the advantages that it is a real and natural problem with a relatively small number of learning examples and with an already-developed model by classical numerical-fitting methods. There also exist test sets with measured biodegradation characteristics. The obtained prediction results of the generated rules developed by the novel inductive learning system are rather encouraging. However, it seems that two other features of the inductive learning system are even more important. The first one is that the generated rules directly point out decisive descriptor values and relations among descriptors. In this way, the rules are very stimulating for further research in the field and they also directly contribute to the overall human knowledge of the topic. The other feature is that the method as shown here for example-based learning in a single binary output application, could be a basic building block for more general learning systems. Two obstacles seem to stand in the way: computational complexity and the problem of general and flexible definition of the starting set of all logical tests that can be used in the rule-generation process.

MODELS*	QSAR	NN	Rule1	Rule2
LEARNING SET	41	47	46	43
TEST SET 1	22	17	20	22
TEST SET 2	14	16	15	17
PREDICTION	90%	82,5%	87,5%	97,5%

* - QSAR: Quantitative Structure Activity Relationship
- NN: Neural Network

Table 2: Prediction results

Acknowledgement *This work was supported by grants Nos. 1-07-159 and 2-06-221 awarded by the Ministry of Science, Technology and Informatics of the Republic of Croatia.*

References

[1] OECD (1989), *Guidelines for testing chemicals, Section 3: Degradation.* Paris, OECD.

[2] R. S. Boethling, B. Gregg, R. Frederick, N. W. Gabel, S. E. Campbell, A.Sabljić, *Expert Systems Survey on Biodegradation of Xenobiotic Chemicals.* Ecotoxicological Environment Safety, Vol.18, 1989, pp.252-267.

[3] *EINECS - European Inventory of Existing Commercial Chemical Substances, Advanced edition.* Commission of the EC, Luxembourg, Office of Official Publications of the European Communities, 1987, (8 volumes).

[4] R. S. Boethling, A. Sabljić, *Screening-Level Model for Aerobic Biodegradability Based on a Survey of Expert Knowledge.* Environmental Science and Technology, Vol.23, 1989, pp.672-679.

[5] B. Cambon, J. Devillers, *New Trends in Structure-Biodegradability Relationships.* Quant. Struct. Act. Relat., Vol.12, No.1, 1993, pp.49-58.

[6] P. H. Howard, A. E. Hueber, R. S. Boethling, *Biodegradation Data Evaluation for Structure/Biodegradability Relations.* Environmental Toxicology and Chemistry, Vol.6, 1987, pp.1-10.

[7] G. J. Niemi, G. D. Veith, R. R. Regal, D. D. Vaishnav, *Structural Features Associated With Degradable and Persistent Chemicals.* Environmental Toxicology and Chemistry, Vol.6, 1987, pp. 515-527.

[8] P. H. Howard, R. S. Boethling, W. M. Stiteler, W. M. Meylan, A. E. Hueber, J. A. Beauman, M. E. Larosche, *Predictive Models For Aerobic Biodegradability Developed From a File of Evaluated Biodegradation Data.* Environmental Toxicology and Chemistry, Vol.11, 1992, pp.593-603.

[9] M. D. Desai, R. Govind, H. H. Tabak, *Development of Quantitative Structure-Activity Relationships for Predicting Biodegradation Kinetics.* Environmental Toxicology and Chemistry, Vol.9, 1990, pp.473-477.

[10] M. Stefik, J. Aikins, R. Balzer, J. Benoit, L. Birnbaum, F. Hayes-Roth, E. Sacerdoti, *The Organization of Expert Systems, A Tutorial.* Artificial Intelligence, Vol.18, 1982, pp.135-173.

[11] H. S. H. Sandell, J. R. Bourne, *Expert Systems in Medicine: A Biomedical Engineering Perspective.* CRC Critical Reviews in Biomedical Engineering, Vol.12, 1985, pp.95-135.

[12] P. Clark, T. Niblett, *The CN2 Induction Algorithm.* Machine Learning, Vol.3, 1989, pp.261-284.

[13] J. R. Quinlan, *Knowledge Acquisition from Structured Data.* IEEE Expert, Vol.6, 1991, pp.32-37.

[14] N. A. B. Gray, *Capturing Knowledge through Top-Down Induction of Decision Trees.* IEEE Expert, Vol.5, June 1990, pp.41-50.

6 Appendix

In the appendix are cited the learning and test examples used in modelling biodegradation. In Table 3 are 48 learning examples. For each example, the values of all 11 descriptors are given. Columns for descriptors are titled a-k as the rows in Table 1. In columns c,g,j,k both binary and quantitative values are given. The next column, titled EST, contains the rate of biodegradation in the range 1-4 estimated by experts. The last four columns contain the results of the prediction by different models:

1 QSAR model

2 neural network (NN) model

3 Rule 1 (48 learning examples, binary descriptors)

4 Rule 2 (160 learning examples, quantitative and binary descriptors).

A result 1 represents fast, and result 0 slow, biodegradation. They are printed in bold if they do not agree with the expert value.

In the same way, in Table 4 and Table 5 the prediction results for the first test set (23 examples) and for the second test set (17 examples) are presented. The only difference is that these tables have a column MES instead of a column EST. In this column, measured data as 1 (fast biodegradation) or 0 (slow degradation) are given.

No	a	b	c	d	e	f	g	h	i	j	k	EST.	1	2	3	4
1	Y	Y	N,0	N	N	N	Y,3	N	N	Y,40	Y,3	2.64	0	0	0	0
2	N	Y	N,0	N	Y	N	Y,2	N	N	N,15	Y,4	1.76	1	1	1	1
3	N	Y	N,0	N	Y	N	N,1	N	N	N,10	Y,4	1.39	1	1	1	1
4	N	N	N,0	Y	Y	N	Y,2	N	N	N,14	N,0	2.58	0	0	0	0
5	N	N	N,0	N	Y	N	Y,2	N	N	N,14	N,0	2.82	0	0	0	0
6	N	N	N,0	N	Y	N	N,0	N	Y	N,10	Y,1	1.73	1	1	1	1
7	N	N	N,0	N	Y	N	N,0	N	N	N, 6	N,0	1.73	1	1	1	1
8	N	N	N,0	N	Y	Y	N,0	N	N	N,12	N,0	2.83	0	0	0	0
9	N	N	Y,3	N	N	N	N,0	N	N	N,13	N,0	3.10	0	0	0	0
10	N	Y	Y,2	N	N	N	Y,2	N	N	Y,31	Y,1	2.81	0	0	0	0
11	N	Y	N,0	N	Y	N	N,1	N	N	N,15	Y,1	1.71	1	1	1	1
12	N	Y	N,0	N	Y	N	N,0	N	N	N,10	Y,1	1.83	1	1	1	1
13	N	N	N,0	N	Y	N	Y,3	N	N	Y,26	N,0	3.00	0	0	0	0
14	N	N	N,0	N	Y	N	N,1	N	N	N,13	N,0	1.83	1	1	1	1
15	Y	N	N,0	N	Y	N	Y,7	N	N	Y,46	Y,4	3.50	0	0	0	0
16	Y	Y	N,0	N	Y	N	Y,5	N	Y	Y,41	Y,2	3.04	0	0	0	0
17	N	N	N,1	N	N	N	N,1	Y	N	N, 9	Y,2	2.09	1	1	1	1
18	N	N	N,0	N	N	N	Y,2	N	N	Y,26	Y,4	2.56	0	0	0	0
19	N	N	N,0	N	N	N	Y,5	N	N	Y,60	N,0	3.09	-	0	0	0
20	N	N	Y,2	N	N	N	N,1	N	N	Y,32	N,0	3.04	0	0	0	0
21	N	Y	N,0	N	Y	N	N,0	N	N	N,10	Y,3	1.87	1	1	1	1
22	N	Y	N,0	N	Y	N	N,0	N	N	N, 9	Y,3	1.64	1	1	1	1
23	N	N	N,0	N	Y	N	Y,2	N	N	N,23	Y,3	2.05	1	1	1	1
24	N	N	N,0	N	Y	N	N,0	N	N	N,11	Y,1	1.92	1	1	1	1
25	N	N	N,0	N	Y	N	Y,2	N	Y	N,12	Y,6	2.36	0	1	1	1
26	N	N	N,0	N	Y	N	N,1	N	Y	N,15	Y,1	2.42	1	1	1	1
28	N	N	N,0	N	Y	N	N,1	Y	N	N,10	Y,2	2.68	1	0	1	1
30	Y	Y	N,0	N	Y	N	N,1	N	N	N,13	Y,3	1.89	1	1	1	1
31	Y	N	N,0	N	Y	N	N,1	N	Y	N,14	Y,2	2.11	0	1	1	1
32	N	N	Y,3	N	N	N	N,1	N	N	Y,37	N,0	2.95	0	0	0	0
33	N	N	N,0	N	N	N	N,0	N	N	Y,31	N,0	3.77	0	-	0	0
34	N	N	N,0	N	Y	N	N,0	N	N	N,24	Y,2	2.08	1	1	1	1
35	N	N	N,0	N	N	N	Y,2	N	N	Y,44	N,0	3.00	0	0	0	0
36	N	N	N,0	N	N	N	Y,2	N	N	N,15	N,0	2.68	0	0	0	0
37	N	Y	N,0	N	Y	N	N,0	N	N	Y,30	Y,10	2.29	1	1	1	1
38	Y	Y	Y,2	N	N	N	N,1	N	N	N,22	Y,3	2.95	0	0	0	0
39	N	N	N,0	N	N	N	N,1	N	N	N,20	N,0	2.59	0	0	0	0
40	N	N	N,0	N	Y	N	Y,2	N	N	Y,28	Y,2	2.54	1	0	0	1
41	N	Y	Y,3	N	N	N	N,0	N	N	Y,33	Y,2	2.71	0	0	0	0
42	Y	N	Y,3	N	N	N	N,1	N	N	N,24	Y,2	3.13	0	0	0	0
43	N	Y	Y,6	N	N	N	Y,3	N	N	Y,37	Y,4	3.80	0	0	0	0
44	Y	N	N,0	N	Y	N	N,1	N	N	N,12	Y,1	2.55	1	0	1	1
45	N	N	N,0	N	N	Y	Y,4	N	Y	Y,57	Y,1	3.10	-	0	0	0
46	N	N	N,0	N	Y	N	N,1	N	N	N,23	N,0	2.32	1	1	1	0
47	Y	N	Y,2	N	N	N	N,1	N	N	N,19	Y,2	3.33	0	0	0	0
48	N	N	N,0	N	Y	N	N,1	N	Y	N,16	Y,1	2.21	1	1	1	1
49	N	N	N,0	N	N	N	Y,3	N	N	Y,36	Y,1	3.05	0	0	0	0
50	N	N	N,0	N	Y	N	N,0	N	N	Y,51	N,0	2.68	0	0	0	1

Table 3: Learning set

No	a	b	c	d	e	f	g	h	i	j	k	MES.	1	2	3	4
1	N	N	N,0	N	Y	N	N,1	N	Y	N,11	Y,1	1	1	1	1	1
2	N	N	N,0	N	Y	Y	N,1	N	Y	N,14	Y,1	1	1	1	1	1
3	N	N	N,0	N	N	Y	N,1	N	N	N,26	Y,3	1	1	0	1	1
4	N	N	N,1	N	N	N	N,1	N	Y	N,13	Y,1	1	1	1	1	1
5	N	Y	N,0	N	Y	N	N,1	N	N	Y,28	Y,6	1	1	1	1	1
6	N	N	N,0	N	Y	N	N,0	N	N	N,23	N,0	1	1	1	1	1
7	N	N	N,0	N	Y	N	Y,2	N	N	N,13	N,0	1	1	0	0	1
8	N	N	N,0	N	Y	N	N,0	N	N	N,25	N,0	1	1	0	1	1
9	N	N	N,0	N	Y	N	Y,2	N	N	N,14	N,0	0	1	0	0	0
10	N	N	N,0	N	N	N	Y,3	N	N	Y,37	Y,3	0	0	0	0	0
11	N	Y	N,0	N	Y	N	N,1	N	N	Y,39	Y,6	0	0	1	1	1
12	N	N	N,1	N	N	N	N,1	N	N	N,11	N,0	0	0	0	0	0
13	N	N	N,0	N	N	N	Y,3	N	N	Y,38	Y,3	0	0	0	0	0
14	N	N	N,0	N	Y	N	Y,3	N	N	N,18	N,0	0	0	0	0	0
15	N	Y	Y,2	N	N	N	Y,2	N	N	Y,31	Y,4	0	0	1	0	0
16	N	N	Y,3	N	N	N	N,1	N	N	N,25	Y,4	0	0	0	0	0
17	N	N	Y,3	N	N	N	N,1	N	Y	N,20	Y,1	0	0	1	0	0
18	N	N	N,0	N	Y	N	Y,4	N	N	N,20	N,0	0	0	0	0	0
19	N	N	N,0	N	Y	N	Y,5	N	N	N,25	N,0	0	0	0	0	0
20	N	N	N,0	N	Y	N	Y,5	N	N	Y,27	N,0	0	0	0	0	0
21	N	N	Y,6	N	N	N	Y,2	N	Y	Y,41	Y,2	0	0	0	0	0
22	N	N	Y,2	N	N	N	Y,2	N	N	Y,27	N,0	0	0	0	1	0
23	N	N	Y,6	N	N	N	N,1	N	N	Y,28	N,0	0	0	0	0	0

Table 4: First test set

No	a	b	c	c	e	f	g	h	i	j	k	MES.	1	2	3	4
1	N	N	N,0	N	Y	N	N,0	N	N	N, 7	N,0	1	1	1	1	1
2	N	N	N,0	N	Y	N	N,1	N	N	N,12	Y,1	1	1	1	1	1
3	N	N	N,0	N	Y	N	N,1	N	Y	N,11	Y,1	1	1	1	1	1
4	N	N	N,0	N	Y	N	N,1	N	Y	N,14	Y,3	1	1	1	1	1
5	N	Y	N,0	N	Y	N	N,1	N	Y	N,17	Y,4	1	1	1	1	1
6	N	N	N,0	N	Y	N	N,1	N	Y	N,15	Y,4	1	1	1	1	1
7	N	N	N,0	N	Y	N	N,0	N	N	N,16	Y,1	1	1	1	1	1
8	Y	N	N,0	N	Y	N	N,1	N	N	N,12	Y,2	1	1	0	1	1
9	N	N	N,0	N	Y	N	N,0	N	N	Y,38	Y,8	1	1	1	1	1
10	Y	N	N,0	N	Y	N	N,1	N	N	N, 9	N,0	0	1	0	1	0
11	N	N	N,0	N	Y	Y	N,1	N	N	N,14	N,0	0	1	0	0	0
12	N	N	N,0	N	Y	Y	N,1	N	N	N,14	N,0	0	1	0	0	0
13	N	N	N,1	N	N	Y	N,1	N	N	N,16	N,0	0	0	0	1	0
14	N	N	N,0	N	Y	N	Y,4	N	N	N,23	N,0	0	0	0	0	0
15	N	N	Y,3	N	N	N	N,1	N	N	N,22	Y,2	0	0	0	0	0
16	N	N	Y,2	N	N	N	N,0	N	N	N,16	N,0	0	0	0	0	0
17	N	N	Y,5	N	N	N	N,1	N	N	N,25	N,0	0	0	0	0	0

Table 5: Second test set

SUCCESSIVE NAIVE BAYESIAN CLASSIFIER

Igor Kononenko
University of Ljubljana, Faculty of Electrical & Computer Engineering,
Tržaška 25, 61001 Ljubljana, Slovenia
e-mail: igor.kononenko@ninurta.fer.uni-lj.si

Keywords: naive Bayesian classifier, successive learning, non-linear problems, empirical learning, empirical evaluation

Edited by: Igor Mozetič
Received: April 21, 1993 **Revised:** May 27, 1993 **Accepted:** July 21, 1993

The naive Bayesian classifier is fast and incremental, can deal with discrete and continuous attributes, has excellent performance in real-life problems and can explain its decisions as the sum of information gains. However, its naivety may result in poor performance in domains with strong dependencies among attributes. In this paper, the algorithm of the naive Bayesian classifier is applied successively enabling it to solve also non-linear problems while retaining all the advantages of naive Bayes. The comparison of performance in various domains confirms the advantages of successive learning and suggests its application to other learning algorithms.

1 Introduction

Let $A_i, i = 1 \ldots n$ be a set of attributes, each having values $V_{i,j}, j = 1 \ldots NV_i$. Let C_j be one out of k possible classes. An object is described with vector $X = (X_1, ..., X_n)$ where X_i may have one of values $V_{i,j}, j = 1 \ldots NV_i$. Let an object with unknown class be described with $X^l = (X_1^l, ..., X_n^l)$. If the conditional independence of attributes with respect to all classes is assumed, the naive Bayesian formula can be used to classify such an object:

$$\hat{P}(C_j|X = X^l) = P(C_j) \prod_{i=1}^{n} \frac{P(C_j|X_i = X_i^l)}{P(C_j)} \quad (1)$$

where prior and conditional probabilities on the right-hand side can be approximated from a set of training examples with known classes. An object is classified by class with maximal probability calculated with (1).

If a limited number of training data are available, the approximation of probabilities with relative frequency becomes unreliable. Cestnik (1990) has shown that instead of using relative frequencies, it is more appropriate to use the *m-estimate* of conditional probabilities:

$$\hat{P}(C_j|X_i = V_{i,J_i}) = \frac{N_{C_j,V_{i,J_i}} + m \times \hat{P}(C_j)}{N_{V_{i,J_i}} + m}, \quad j = 1..k \quad (2)$$

and Laplace's law of succession (Good, 1950) for prior probabilities of k classes:

$$\hat{P}(C_j) = \frac{N_{C_j} + 1}{N + k}, \quad j = 1..k \quad (3)$$

In the above formulas $N_{C_j,V_{i,J_i}}$ represents the number of training instances with value V_{i,J_i} of the i-th attribute and belonging to the j-th class, N_{C_j} and $N_{V_{i,J_i}}$ are interpreted in a similar manner and N is the number of all training instances. The same formula was also used by Smyth and Goodman (1990). Parameter m trades off the relative frequency and the prior probability. A lower setting for m suggests stronger belief in training data, whereas a higher setting implies greater reliance on prior probabilities. In our experiments described in section 4, parameter m was set to 2, which is an empirically verified appropriate choice for a typical learning problem (Cestnik, 1990).

The naive Bayesian formula applies to discrete attributes, while continuous attributes have to be

discretized in advance. It was shown that fuzzy discretization for modeling continuous attributes achieves a better performance and that the results are less sensitive with respect to factual discretization (Kononenko, 1991). In experiments described in this paper, fuzzy discretization was not applied.

Many authors have experimentally verified that the naive Bayesian formula achieves better or at least as good classification accuracy as inductive learning algorithms in many real-world problems (Kononenko et al., 1984; Cestnik, 1990; Smyth and Goodman, 1990) and, surprisingly, the explanation ability of naive Bayes, at least in inexact domains such as medical diagnosis, is better (as estimated by physicians) than that of a decision tree (Kononenko, 1990). The kind of explanation by naive Bayes is the *sum of information gains* by each attribute for/against each class, which is obtained with logarithm of eq. (1):

$$-\log_2 \hat{P}(C_j|X = X^l) = -\log_2 P(C_j)$$
$$-\sum_{i=1}^{n}\left(\log_2 P(C_j|X_i = X_i^l) - \log_2 P(C_j)\right) \quad (4)$$

The explanation can be presented to human experts as a list of attribute values with corresponding information gains for each class that appear in the sum on the right-hand side of equation (4). Human experts appeared to prefer explanations of this type to a single if-then rule for a classified object.

However, the naivety of formula (1) can be too drastic in certain domains with strong dependencies among attributes. A classical non-linear problem which cannot be solved by naive Bayes is exclusive "or" (XOR):

$$Class(X) = \begin{cases} C_1, & X_1 \neq X_2 \\ C_2, & X_1 = X_2 \end{cases} \quad (5)$$

This problem is also hard for other machine-learning algorithms. One class of hard machine-learning problems contains parity problems of higher degrees. This paper describes a method for successive application of naive Bayes which under certain conditions can solve parity problems. In the next section, the theoretical limitations of naive Bayes are briefly discussed and results on well-known problems are compared to those of other learning algorithms. In Section 3, the successive naive Bayesian classifier is described and section 4 gives empirical results on various problem domains. In the discussion, a generalization of the approach to other learning algorithms is proposed.

2 Performance of naive Bayes

Despite its limitations, the performance of the naive Bayesian classifier in many real-world problems is excellent compared to that of other learning algorithms. In table 1 the performance on three well-known medical diagnostic problems - primary tumor, breast cancer, and lymphography - is compared to that of other propositional-logic learning algorithms. We also tested naive Bayes on the problem of finite element mesh design (Dolšak & Muggleton, 1991), which has been a focus of the inductive logic programming (ILP) community (see section 4 for descriptions of learning data). Although it cannot use information about geometric properties of objects in this domain (this holds for all propositional logic algorithms) it outperformed all sophisticated ILP algorithms. The comparison is given in table 2.

Let us now examine more closely the limitations of the naive Bayesian classifier. With appropriate recoding of objects, the naive Bayesian classifier can also be interpreted as a linear function which discriminates between two classes C_i and C_j:

$$Class(Y) = \begin{cases} C_i, & P^{i,j}Y > 0 \\ C_j, & P^{i,j}Y < 0 \end{cases} \quad (6)$$

where
vector $Y = (1, Y_{1,1}, .., Y_{1,NV_1}, ..., Y_{n,1}, .., Y_{n,NV_n})$ is recoded vector X so that each attribute's value corresponds to one vector's component:

$$Y_{i,j} = \begin{cases} 1, & X_i = V_{i,j} \\ 0, & X_i \neq V_{i,j} \end{cases} \quad (7)$$

and $P^{i,j}Y$ is the inner product of vector $P^{i,j}$, which is used to discriminate between classes C_i and C_j, and vector Y. The term $P^{i,j}$ is obtained by subtracting two instances of eq. (4):

$$P^{i,j} = (p_0^{i,j}, p_{1,1}^{i,j}, .., p_{1,NV_1}^{i,j}, ..., p_{n,1}^{i,j}, .., p_{n,NV_n}^{i,j}) \quad (8)$$

where

$$p_0^{i,j} = \log_2 P(C_i) - \log_2 P(C_j)$$

algorithm	reference	prim.tumor	brea.cancer	lymphogr.
Assistant	(Kononenko et al., 1984)	44 %	73 %	77 %
AQ15	(Michalski et al., 1986)	41 %	66 %	80 %
Assistant 86	(Cestnik et al., 1987)	44 %	77 %	76 %
LogArt	(Cestnik & Bratko, 1988)	44 %	78 %	84 %
CN2	(Clark & Boswell, 1991)	46 %	73 %	82 %
naive Bayes		51 %	79 %	84 %

Table 1: Performance of different algorithms in three medical domains.

algorithm	reference	accuracy
FOIL	(Quinlan, 1990)	12 %
mFOIL	(Džeroski, 1991)	22 %
GOLEM	(Dolšak & Muggleton, 1991)	29 %
LINUS	(Lavrač & Džeroski, 1991)	29 %
naive Bayes		33 %

Table 2: Performance of different algorithms in finite element mesh problem.

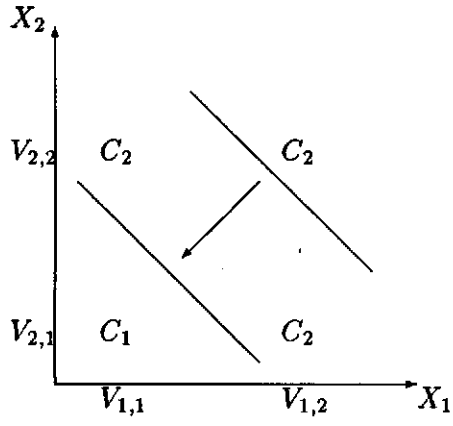


Figure 1 Applying delta learning rule to naive Bayes

and

$$p_{l,m}^{i,j} = \log_2 \frac{P(C_i|X_l = V_{l,m})}{P(C_i)} - \log_2 \frac{P(C_j|X_l = V_{l,m})}{P(C_j)}$$

For each pair of classes we have one linear discrimination function. This derivation confirms that the naive Bayesian classifier is limited to linear decision functions, which cannot solve non-linear problems.

This raises the question of whether it is worth using a delta learning rule, in the sense of perceptrons (Minsky & Papert, 1969), to adapt the discrimination function to discriminate more reliably between classes on training data. This can be achieved by iteratively duplicating training instances not correctly classified by naive Bayes. This alters the probability distribution so that the discrimination function moves in an appropriate direction. This is illustrated in Fig. 1 where, by duplicating instances $(V_{1,1}, V_{2,2})$ and $(V_{1,2}, V_{2,1})$, the discrimination function is changed into a perfect discriminator.

However, such changes in probability distribution are inappropriate for predicting cases unseen during learning. The decision function in fact overfits the training data which affects the performance on unseen cases. We tried the above (delta) learning rule on several medical diagnostic problems. The performance on training data increased (in lymphography it even reached 100% classification accuracy), but the performance on test data of the classifier drastically decreased. This suggests that the decision function given by the original probability distribution is optimal among linear discrimination functions.

One should be careful when changing the representation space. In fact, if Y representation is used instead of X, the space is much sparser, as many points are illegal (an instance cannot have more than one attribute's value). On the other

hand, in the original space (X coding) in general the discrimination function of naive Bayes is not linear. For clarity of illustration, in the next section we will assume approximately linear discrimination functions in the original space.

3 Successive learning with naive Bayesian classifier

If one tries to solve the XOR problem with the naive Bayesian classifier, the result may be one of the decision curves (a or b) from Figure 2.1. The direction of the curve depends on the distribution of training instances. However, if all instances are equally likely, no decision curve appears, as all components of $P^{1,2}$ are equal to 0. In such a case, it is desirable to modify slightly the distribution (e.g., by duplicating one of the instances) to get one of the decision curves (i.e. breaking the symmetry).

To enable the naive Bayesian classifier to solve the XOR problem, the same algorithm may be repeatedly applied, each time on a redefined problem. In each iteration, training instances that are correctly classified by the current discrimination function are assigned to an additional special class C_0 and the other training instances retain their original classes. The resulting learning tasks of the XOR problem (depending on the current discrimination function) and their solutions are depicted in Fig. 2.2a, and 2.2b. There is one discrimination function for each pair of classes (labeled with their indices). In both cases, the discrimination is perfect after two successive learning iterations. However, for parity problems of higher order, more iterations may be needed. In general, it is not always possible to obtain perfect discrimination with such successive learning. On the other hand, perfect discrimination of training instances usually implies overfitting. This is avoided here by keeping all training instances for each new learning problem, thus enabling reliable probability estimates. Overfitting the training data may be interpreted as reliance on unreliable probability estimates from a small number of training instances. Fig. 3 illustrates successive learning in a parity problem involving two three-valued attributes and three classes. in Fig. 3 (1), two discrimination lines (between classes 2 and 3 (2-3) and 2 and 1 (2-1)) overlap.

The above discussion leads to the following learning algorithm:

repeat
    Train naive Bayes;
    Change the class label of correctly classified training instances to C_0;
until all training instances are classified to C_0

Note that the terminating condition does not require perfect classification (i.e. not all training instances need be correctly classified). This algorithm may enter an infinite loop if perfect discrimination is impossible. However, more iterations do not cause overfitting of the training data, as all training instances are used in all iterations. For practical reasons, it is necessary to limit the number of iterations. In our experiments described in the next section, the number of iterations was limited to ten.

When classifying new objects, the discrimination function learned last should be tried first. If the result is class C_0 the next latest function must be tried, while if the result is one of the original classes, it is accepted as an answer. The reverse order for classification follows from the training algorithm, because the classification into a class other than C_0 is more reliable with the latest discrimination function. Eventually, by repeted application of discrimination functions in reverse order, a class $C_i, i > 0$ is obtained as an answer.

4 Experimental results

We applied successive naive Bayesian learning to several data sets from medical diagnostics, chess endgame, criminology, engineering, and one artificial data set. Basic data characteristics are given in table 3. A brief description of each problem follows:

Primary tumor: Locate the position of the primary tumor in the body of a patient with metastases.

Breast cancer: Predict the recurrence of the disease in five years after the operation.

Lymphography: Classify the type of tumor of a patient with metastases.

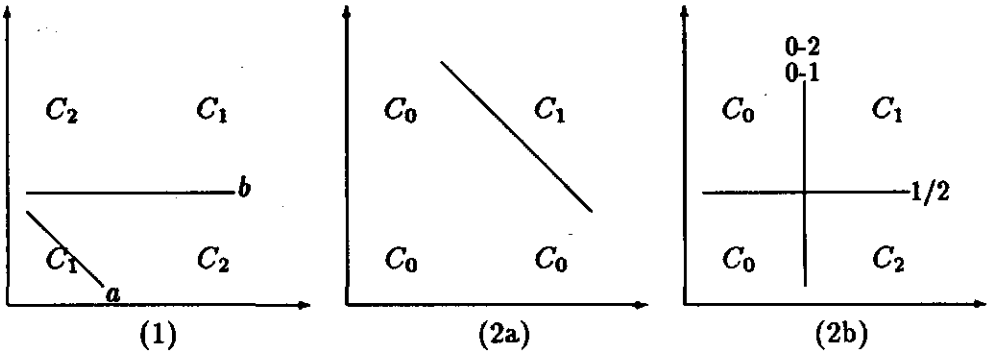


Figure 2
(1) Original XOR problem
(2a) New problem obtained from discrimination function a
(2b) New problem obtained from discrimination function b

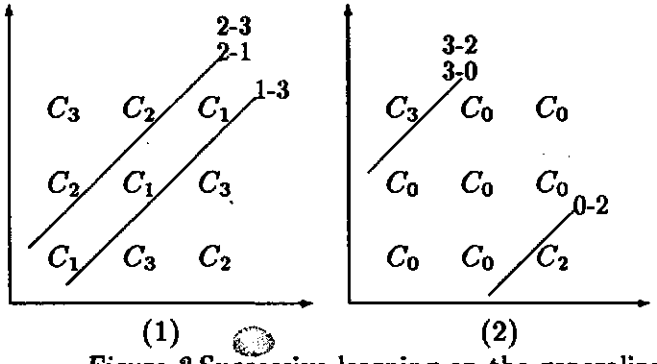


Figure 3 Successive learning on the generalized parity problem

domain	#class	#atts.	#val/att.	# instances
primary tumor	22	17	2.2	339
breast cancer	2	10	2.7	288
lymphography	4	18	3.3	148
rheumatology	6	32	9.1	355
criminology	4	11	4.5	723
chess 1	2	6	8.0	1000
chess 2	2	18	2.0	1000
mesh 1	13	3	7.0	278
mesh 2	13	15	7.3	278
artificial	2	12	2.0	200

Table 3: Basic description of data sets

Rheumatology: Determine the type of rheumatologic disease.

Criminology: Determine the education of the violator.

Chess 1: Detect illegal positions in King-Rook-King chess endgame given only the coordinates of pieces.

Chess 2: Detect illegal positions in King-Rook-King chess endgame given the relations such as "same rank", "neighbour file", etc.

Mesh 1: Determine the number of elements of an edge in a finite element mesh design, given the three basic attributes but no geometric relations.

Mesh 2: Determine the number of elements of an edge in a finite element mesh design, given the three basic attributes and additional attributes such as the number and the type of neighbour edges.

Artificial: A data set was generated with two attributes defining parity relation with class, 5 additional random binary attributes and 5 additional independent and slightly informative binary attributes. In addition, class labels were corrupted with 5% noise (5% of cases had wrong class labels).

Except in the "mesh" problems, the experiments were performed with 10 random splits on 70% of the instances for training and 30 % for testing. The results were averaged. In "mesh" problems, experiments were done in the same way as with ILP systems (see Džeroski, 1991, for details). The measured parameters were:

- accuracy: the percentage of correctly classified instances
- average information score (Kononenko & Bratko, 1991): a measure that eliminates the influence of prior probabilities. It is defined as follows:

$$Inf = \frac{\sum_{i=1}^{\#testing\ instances} Inf_i}{\#testing\ instances} \qquad (9)$$

where information score of classification of the i-th testing instance is defined by (10):

problem	naive Bayes		successive Bayes	
	%	bit	%	bit
primary tumor	51.0	1.57	51.7	1.61
breast cancer	79.2	0.18	78.4	0.16
lymphography	84.2	0.83	83.9	0.82
rheumatology	67.2	0.51	68.3	0.53
criminology	61.2	0.27	61.5	0.27
chess 1	66.2	0.18	66.5	0.18
chess 2	91.7	0.73	92.3	0.75
mesh 1	33.5	0.61	32.4	0.60
mesh 2	34.5	0.62	36.0	0.66
artificial	61.8	0.24	78.3	0.57

Table 4: Results of naive and successive naive learning.

where Cl_i is the class of i-th testing instance, $P(Cl)$ is the prior probability of class Cl and $P'(Cl)$ the probability returned by a classifier.

Results are summarized in table 4.

The results indicate that the performance of successive learning is the same as that of naive Bayes in most real-world domains. The only significant difference according to accuracy and information score appears in "primary tumor" and "mesh 2" problems. Both problems are very difficult (see tables 1 and 2), involving many classes. The result with the artificial data set indicates that successive learning may be much better in domains with strong dependencies among attributes.

5 Discussion

The results of experiments suggest that successive naive Bayesian learning may improve the performance of naive Bayes while preserving its advantages: simplicity, efficiency and transparency. The successive learning approach keeps all training instances together in all learning iterations and thus avoids the overfitting problem. However, the algorithm may reach a non-solvable learning problem. Covering algorithms (e.g. AQ, Assistant, CN2 and FOIL) discard correctly covered training instances in each iteration and are able to discriminate cases in any learning problem, but with great danger of overfitting the training data.

$$Inf_i = \begin{cases} -\log_2 P(Cl_i) + \log_2 P'(Cl_i), & P'(Cl_i) \geq P(Cl_i) \\ -(-\log_2(1 - P(Cl_i)) + \log_2(1 - P'(Cl_i))), & P'(Cl_i) < P(Cl_i) \end{cases} \quad (10)$$

The principle of successive learning can be used with any learning algorithm and, probably more efficiently, different learning algorithms may be successively applied. Further investigations should empirically verify this hypothesis, as well as the idea of combining the successive and covering approaches. More theoretical work is needed to determine the limitations of successive learning and answer questions such as: which problems cannot be solved with successive learning and which problems lead the approach into infinite cycling.

6 Acknowledgements

Medical data sets were obtained from the University Medical Center in Ljubljana, and were provided by Dr. Matjaž Zwitter (primary tumor and breast cancer), Dr. Milan Soklič (lymphography), and Dr. Vladimir Pirnat (rheumatology). The KRK endgame data set is the first of five KRK data sets used by Sašo Džeroski for testing rules generated by mFOIL (Džeroski, 1991); the data sets were originally provided by Michael Bain. The Mesh data were provided by Bojan Dolšak. Boštjan Vovk performed experiments with delta learning rule described in section 2. This research was supported by the Slovenian Ministry of Science and Technology.

References

[1] Cestnik, B. (1990) Estimating probabilities: A crucial task in machine learning, *Proc. European Conference on Artificial Intelligence*, Stockholm, August 1990, pp.147-149.

[2] Cestnik, B. & Bratko, I. (1988) Learning redundant rules in noisy domains, *Proc. European Conf. on Artificial Intelligence ECAI-88*, München, August 1988, pp.348-350.

[3] Cestnik, B., Kononenko, I., Bratko, I. (1987) ASSISTANT 86 : A knowledge elicitation tool for sophisticated users. In: I. Bratko, N. Lavrač (eds.), *Progress in Machine learning.* Wilmslow, England: Sigma Press.

[4] Clark P. & Boswell R. (1991) Rule induction with CN2: Recent improvements, *Proc. EWSL 91*, Porto, March 1991, pp.151-163.

[5] Dolšak, B. & Muggleton, S. (1991) The application of inductive logic programming to finite element mesh design. *Proc. 8th Inter. Workshop on Machine Learning*, Evanstone: Morgan Kaufmann.

[6] Džeroski S. (1991) Handling noise in inductive logic programming, M.Sc. Thesis, University of Ljubljana, Faculty of Electrical & Computer Engineering, Ljubljana.

[7] Good I.J. (1950) *Probability and the weighing of evidence.* London: Charles Griffin.

[8] Kononenko, I. (1990) Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition. In: B. Wielinga et al. (eds.) *Current Trends in Knowledge Acquisition*, Amsterdam: IOS Press.

[9] Kononenko, I. (1991) Feedforward Bayesian neural networks and continuous attributes, *Proc. Int. Joint Conf. on Neural Networks IJCNN-91*, Singapore, Nov. 1991, pp. 146-151.

[10] Kononenko, I. & Bratko, I. (1991) Information based evaluation criterion for classifier's performance. *Machine Learning*, Vol.6, pp.67-80.

[11] Kononenko, I., Bratko, I., Roškar, E. (1984) Experiments in automatic learning of medical diagnostic rules, Technical report, Faculty of Electrical & Computer Eng., Ljubljana, Slovenia (presented at ISSEK Workshop, Bled, August, 1984).

[12] Lavrač, N. & Džeroski, S. (1991) Inductive learning of relations from noisy examples, In: Muggleton, S. (ed.) *Inductive logic programming*, Glasgow: Academic Press.

[13] Michalski, R.S., Mozetič, I., Hong, J., Lavrač, N. (1986) The multi purpose incremental learning system AQ15 and its testing

application to three medical domains. *Proc. of the National Conf. on Artificial Intelligence AAAI 86.* Philadelphia, August, 1986, pp.1041-1047.

[14] Minsky, M. & Papert, S. (1969) *Perceptrons*, Cambridge, MA : MIT Press.

[15] Quinlan (1990) Learning logical definitions from relations, *Machine Learning*, Vol. 5, No. 3, pp.239-266.

[16] Smyth P. & Goodman R.M. (1990) Rule induction using information theory. In: G.Piarersky & W.Frawley (eds.) *Knowledge Discovery in Databases*, MIT Press.

MORAL HAZARD PROBLEM SOLVING BY MEANS OF PREFERENCE RANKING METHODS

Ines Saražin Lovrečič,
Health Care Institution of Slovenia,
Miklošičeva 24, 61000 Ljubljana, Slovenia
AND
Janez Grad,
Department of Economics, University of Ljubljana,
Kardeljeva pl. 17, 61000 Ljubljana, Slovenia

Keywords: moral hazard, preference ranking, pseudo-model

Edited by: Matjaž Gams
Received: February 17, 1993 **Revised:** May 18, 1993 **Accepted:** July 21, 1993

Moral hazard problems in the field of humanitarian health aid delivery can be difficult to solve, especially in outstanding circumstances caused by human or natural factors. In this paper, we present a solution to this problem by means of preference-ranking methods. The idea of a pseudo-model is also included, where standard input is considered as well as subjective elements.

1 Presentation of the problem

The treatment of refugees from Bosnia-Herzegovina and Croatia in 1992 presents a problem which the Slovenian health care system has to solve on the macroeconomic level. The problems which occur are as follows:

- shortage of financial resources,
- shortage of sanitary and pharmaceutical material,
- daily variation of data which depends both on the domestic and foreign political environment.

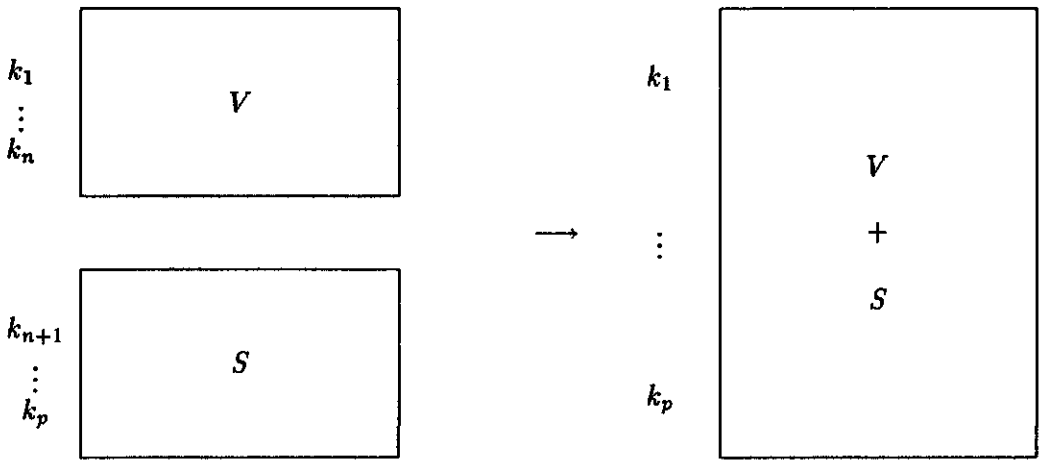
Since the media inform us daily about the lack of financial resources, we will not follow this topic any further. Let us address the issue of how much demand can be covered by the available state budget and how much help we can expect from various humanitarian organisations (domestic and foreign). Simultaneously, we raise the question, which risk group has priority at delivery. Therefore, our task is *moral hazard* problem solving.

With regard to available facilities of the Slovenian health care system (supply) and requests (demand), we defined criteria which can be considered in various optimization models, such as: rationalisation of sanitary material, medicines, maximisation of preventive medicine etc. This can be formalised as a vector of criteria $[k_1, \ldots, k_n]^T$. Along with standard criteria $k_1, \ldots, k_n$ they are the so-called subjective criteria, representing the impact on the final decision of subjective reasoning (see Figure 1) based on the estimated help from unreliable sources. The result of such a model is a set of optimal solutions of the preference functions under given conditions such that the space of optimal development of health aid is bounded by this optimal set.

Example. Suppose that we have two vaccination programmes for war refugees. The first one makes use of only reliable domestic resources, while the second one anticipates only financial and material support from abroad and charitable organisations. In the current situation, we can hardly judge which of the two programmes is more realistic.

2 Modern preference-ranking methods

The multicriteria nature of moral-hazard problems requires a suitable solving method. On the



V ...results from the optimisation and simulation model
S ...suitably formalised subjective elements

Figure 1: Combination of matrices V and S

basis of already-known advantages [3, 1] of up-to-date methods of multicriteria decision making, we decided to use the preference-ranking method as a tool for problem solving.

PROMETHEE (**Pre**ference **R**anking **O**rgani-zation **Meth**od for **E**nrichment **E**valuations) is a group of general-purpose methods, developed in Europe and also used elsewhere in the world. Their purpose is to help the decision maker in alternative evaluations using preference functions. For detailed discussion of the methods, see [1], and [3] for a specialised version for health care system. Here we only devise the necessary theoretical basis for PROMETHEE.

Let A be the the set of feasible decisions (actions). Suppose that criteria $c_1, \ldots, c_m$ are applied by the decision maker to evaluate individual actions; in short, c_j are numeric functions defined on A. The decision maker defines a generalised criterion $Q_j(a,b)$, also called the *preference function* (PF) for every c_j. Actually, it is a function of the difference $c_j(a) - c_j(b)$, where $a, b \in A$. There are six standard types of PF [1] and three types specialised for health-care system problem-solving [3]. In addition, most types have some parameters to determine. The choice of type of PF will be shown later by an example.

Define *preference index* Π as the average of all generalised criteria:

$$\Pi(a,b) = \sum_{j=1}^{m} w_j Q_j(a,b),$$

where w_j are weights ($w_j \geq 0$, for all j and $\sum_{j=1}^{m} w_j = 1$), a and b are arbitrary actions. The basis for action ranking is given by the so-called *flows* (leaving, entering, and net flow):

$$\Phi^+(a) = \sum_{b \in A} \Pi(a,b),$$

$$\Phi^-(a) = \sum_{b \in A} \Pi(b,a),$$

$$\Phi(a) = \Phi^+(a) - \Phi^-(a).$$

Since the argument of PF is the difference $c_j(a) - c_j(b)$, the choice of parameters depends greatly on the distribution of differences for all $a, b \in A$. The use of PF is sensible only if the ranking can be influenced by their parameters. The accurate determination is left to the decision maker for the concrete problem. But the interval from the smallest to the biggest difference is recommended.

3 Formalisation of the pseudo-model

Given a situation where both standard and subjective elements are to be considered, we combine both matrices V and S into one matrix denoted by T (Figure 1). The entries of T represent the input into the PROMETHEE model. The procedure where the subjective elements are taken into account is called *pseudo-modelling*. In our case, by delivering health aid, the risk groups are ranked according to the results of pseudo-modelling.

Table 1: The model standard input

criteria	min/max	A_1 children	A_2 women	A_3 elder	A_4 rest	type p.f.	parameters
C_1	max	19.81	2.62	2.10	0.26	I	–
C_2	max	6.93	1.98	0.80	0.20	I	–
C_3	min	1.15	0.16	27.50	1.28	III	$p = 25$
C_4	min	96.25	27.50	11.00	2.75	III	$p = 65$
C_5	min	23.45	6.70	2.68	0.67	V	$p = 18, q = 0.60$

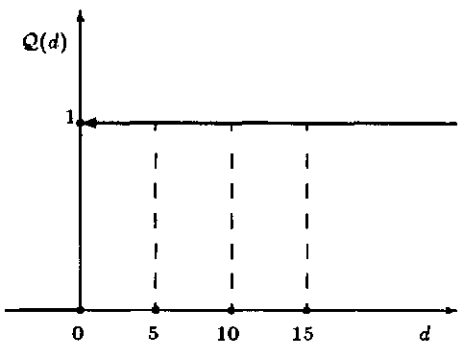


Figure 2: PF for criterion C_1

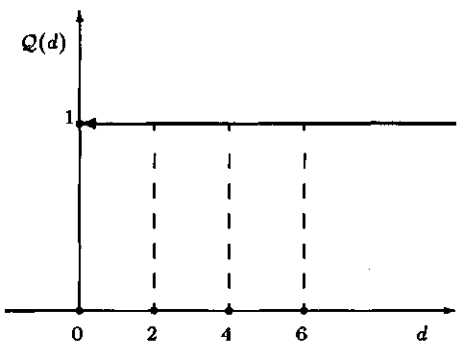


Figure 3: PF for criterion C_2

4 Numerical example

For preventive action, we take four health programmes (HP). The first one makes use only of reliable domestic resources, while the others foresee financial and material support from abroad and charitable organisations. Programmes differ in the costs which have to be covered for the same target i.e. the most suitable scheduling of risk groups versus different preventive programmes.

4.1 Under the first programme, all costs are covered by domestic resources (100%).

In table 1 only standard input is taken into account in the PROMETHEE model.

From table 1 it is clear that there are five criteria altogether which refer to the material costs of preventive vaccination. Criterion C_1 measures preventive examination costs, C_2 vaccination costs (labour, vaccine), C_3 sanitary material costs, C_4 laboratory material costs and C_5 medical costs. The first two are maximised on the 'better to prevent than to cure' principle, the other three are minimised.

The actions are represented as risk groups: children (A_1), women (A_2), elder persons (A_3) and others (A_4).

In table 1 the average values for each criterion and action are also shown.

The types of PF with adequate parameters are determined according to the rules in [1].

For the first criterion, we stick to the usual argument that high-quality preventive examination is particularly important, regardless of the costs. Accordingly, we choose the type of PF which treats every minimal difference d as strict preference. The type I suits these requirements and it has no parameters to determine (see Figure 2).

For the second criterion, we still do not rationalise the imunisation and vaccine costs. Both are necessary for preventing infections and deseases. Again, the most suitable choice is PF of type I. The difference between the costs of various immunisation programmes are illustrated in Figure 3.

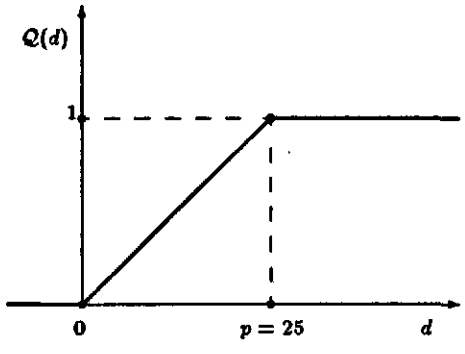


Figure 4: PF for criterion C_3

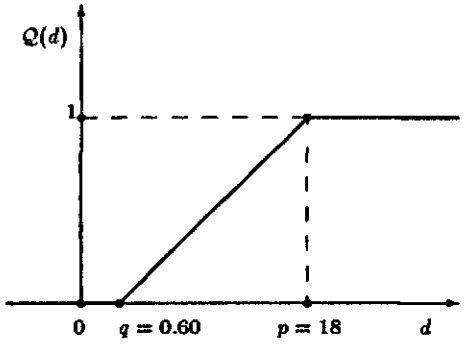


Figure 5: PF for criterion C_5

Criterion C_3 represents the costs of sanitary material, which are linearly dependent on its prices. The same is true for stored quantities. Here we choose PF of type III, i.e. PF with linear preferences. It is shown in Figure 4.

Type III of PF is also chosen for the fourth criterion and is justified by the same argument as for C_3.

For criterion C_5 the principle of rationalisation is used again. However, in contrast to the last two criteria, we introduce the so-called indifference threshold q. It stands for nonsensitivity to differences between costs of medicines to a certain extent. We pay attention to them only when the differences exceed the threshold. Such a situation can be dealt with using PF of type V with parameters q and p (Figure 5). The first parameter is the indifference threshold and the second denotes the strict preference threshold.

The results of the computer-solved problem are presented in Table 2.

The preference outranking list is defined by net flows. We see that the highest priority for delivering humanitarian aid has the risk group A_2 (women), followed by A_1 (children), A_4 (others) and A_3 (elder persons).

4.2 The second programme includes an additional two criteria O_1 and O_2, which determine implementability of C_1 and C_2 respectively, in the range between 91-100%. This means that domestic resources cover at least 90%, while the 1-9% gap will be covered in some other way. The second HP is considered to be optimistic because of the high rate of implementability.

Let us now combine the standard input data with the optimistic estimated implementability of criteria C_1 and C_2. Input data for this pseudo-model are shown in Table 3.

The results of analysis are presented in Table 4.

4.3 In the third HP, we are able to cover at most 75% of the costs, which determine the implementability of C_1 and C_2. In the model, two criteria of implementability are denoted by P_1 and P_2. This programme is considered pesimistic, in contrast with the previous one.

The data for standard input and the pessimistic HP are collected in Table 5.

The results of pseudo-modelling for the pessimistic cost coverage are presented in Table 6.

4.4 The last HP is a compromise between the previous two, because it is planned that 76-90% of the costs are covered by domestic resources. Here the criteria of implementability C_1 and C_2 are denoted by K_1 and K_2.

Table 7 containts the data which refer to the HP of compromise.

The results obtained are displayed in Table 8.

5 Comparison of the results

Since we considered

- the same standard input for all cases,
- the same types of PF for all cases,
- the same parameters for PF and
- the same weights for all criteria,

Table 2: Results of analysis at the standard input

action	leaving flow	enter. flow	net flow	outranking list
A_1	1.4010	1.1936	0.2075	2
A_2	1.4025	0.6286	0.7739	1
A_3	0.8901	1.4416	-0.5515	4
A_4	0.7802	1.2100	-0.4298	3

Table 3: Input data for the optimistic HP

criteria	min/max	A_1 children	A_2 women	A_3 elder	A_4 rest	tip p.f.	parameters
C_1	max	19.81	2.62	2.10	0.26	I	–
C_2	max	6.93	1.98	0.80	0.20	I	–
C_3	min	1.15	0.16	27.50	1.28	III	$p = 25$
C_4	min	96.25	27.50	11.00	2.75	III	$p = 65$
C_5	min	23.45	6.70	2.68	0.67	V	$p = 18, q = 0.60$
O_1	max	0.91	0.92	0.95	0.93	I	–
O_2	max	0.99	1.00	0.91	0.91	I	–

Table 4: Results of analysis of the optimistic HP

action	leaving flow	enter.flow	net flow	outranking list
A_1	1.2865	1.4240	-0.1375	2
A_2	1.5732	0.7347	0.8385	1
A_3	1.0643	1.3154	-0.2511	3
A_4	0.8430	1.2929	-0.4499	4

Table 5: Input data for the pesimistic HP

criteria	min/max	A_1 children	A_2 women	A_3 elder	A_4 rest	tip p.f.	parameters
C_1	max	19.81	2.62	2.10	0.26	I	–
C_2	max	6.93	1.98	0.80	0.20	I	–
C_3	min	1.15	0.16	27.50	1.28	III	$p = 25$
C_4	min	96.25	27.50	11.00	2.75	III	$p = 65$
C_5	min	23.45	6.70	2.68	0.67	V	$p = 18, q = 0.6$
P_1	max	0.60	0.70	0.65	0.67	I	–
P_2	max	0.62	0.63	0.68	0.74	I	–

Table 6: Results of analysis of the pesimistic HP

action	leaving flow	enter.flow	net flow	outranking list
A_1	1.0007	1.7097	-0.7089	4
A_2	1.5732	0.7347	0.8385	1
A_3	1.0643	1.4583	-0.3939	3
A_4	1.2715	1.0071	0.2644	2

Table 7: Input data for the HP of compromise

criteria	min/max	A_1 children	A_2 women	A_3 eldest	A_4 rest	tip p.f.	parameters
C_1	max	19.81	2.62	2.10	0.26	I	-
C_2	max	6.93	1.98	0.80	0.20	I	-
C_3	min	1.15	0.16	27.50	1.28	III	$p = 25$
C_4	min	96.25	27.50	11.00	2.75	III	$p = 65$
C_5	min	23.45	6.70	2.68	0.67	V	$p = 18, q = 0.6$
K_1	max	0.75	0.80	0.85	0.90	I	-
K_2	max	0.80	0.79	0.81	0.78	I	-

Table 8: Results of analysis for the HP of compromise

action	leaving flow	enter.flow	net flow	outranking list
A_1	1.2865	1.4240	-0.1375	3
A_2	1.2875	1.0205	0.2670	1
A_3	1.3501	1.1726	0.1775	2
A_4	0.9858	1.2929	-0.3070	4

Table 9: Net flow value analysis of the standard and optimistic HP

action	Φ_V	Φ_{V+S}	D	$\frac{D}{\lvert\Phi_V\rvert}(\%)$
A_1	0.2075	-0.1375	-0.3450	-166.27
A_2	0.7739	0.8385	0.0646	8.35
A_3	-0.5515	-0.2511	0.3004	54.47
A_4	-0.4298	-0.4499	-0.0201	-4.68

Table 10: Net flow value analysis of the standard and pessimistic HP

action	Φ_V	Φ_{V+S}	D	$\frac{D}{\lvert\Phi_V\rvert}(\%)$
A_1	0.2075	-0.7089	-0.9164	-441.64
A_2	0.7739	0.8385	0.0646	8.35
A_3	-0.5515	-0.3939	0.1576	28.58
A_4	-0.4298	0.2644	0.6942	161.52

the essential ascertaining is as follows. The addition of subjective elements to the standard input is the cause of change in the preference structure, i.e. the rankings of alternative risk groups. It can be deduced from the comparison of results that the smallest discrepancy is found between the standard and optimistic HP. The cause of this phenomenon lies in the high percentage of realisability of criteria C_1 and C_2. In the case where we decide to apply pseudo-modelling, moral-hazard problem solving depends on the input data of the subjective characters.

In the follow-up, we have to examine the changes of net flows which are due to the addition of subjective elements. Table 9 shows the values of net flows of the standard input Φ_V, as well as the optimistic programme Φ_{V+S}, the differences between net flows of the standard input $|\Phi_{V+S} - \Phi_V| = D$ for all actions, and changes relative to the net flows of the standard input ($\frac{D}{|\Phi_V|}$). The comparison of results between the standard and pessimistic HP is found in Table 10, while Table 11 refers to the standard programme and the programme of compromise.

The relative changes for particular actions are again minimal when comparing the standard and optimistic HP. Surely this is a consequence of the smallest discrepancy between the optimistic and standard HP in view of their inputs. In practical terms, with the optimistic HP the health-care system is able to cover almost all costs of health aid. In other words, with at most 9% reduction in certainty of the cost coverage, only two (already adjacent) actions swapped their places in the preference structure. Net flow analysis shows that their absolute values change with the addition of subjective elements and they do not change uniformly for each action. Therefore, the preference structure changes if:

Table 11: Net flow value analysis of the standard and HP of compromise

action	Φ_V	Φ_{V+S}	D	$\frac{D}{\lvert\Phi_V\rvert}(\%)$
A_1	0.2075	-0.1375	-0.3450	-166.27
A_2	0.7739	0.2670	-0.5069	-65.50
A_3	-0.5515	1.1770	1.7285	313.42
A_4	-0.4298	-0.3070	0.1228	28.57

- we add subjective elements and
- we change their values.

From this point the analysis can be continued, for instance with varying implementability intervals of criteria, and studying stability of preference structure. We can also consider more criteria of implementability. Finally, we can observe the behaviour of particular actions according to the varying implementability intervals of criteria or the addition of new criteria.

6 Summary

In this paper, we have exposed the moral-hazard problem in the field of humanitarian health aid delivery in outstanding circumstances. In the practical example, we have dealt with four various preventive health programmes. For the case when both objective and subjective elements are included, we constructed a pseudo-model. The PROMETHEE method is the basic tool for risk-group ranking. Both subjective and objective elements are treated equally, so we can avoid over and under estimation of either group of factors.

References

[1] Brans J.P., Mareschal B. (1986): How to select and how to rank projects. North Holland, European Journal of Operational Research (24), pages 228-238.

[2] Drummond Michael F., Stoddard Greg. L., Torrance George W. (1990): Methods for the Economic Evaluation of Health Care Programmes. Oxford, University Press, 182 pages.

[3] Saražin L. Ines, Grad Janez (1992): Multicriteria decision making in health care system (in slovene language). Ljubljana, Slovene economic review, (43/5), pages 334-347.

FIFTH GENERATION COMPUTER SYSTEMS (FGCS) PROJECT IN JAPAN

Koichi Furukawa
Faculty of Environmental Information, Keio University
5322 Endo, Fujisawa-shi, Kanagawa, 252 Japan
furukawa@icot.or.jp

Keywords: concurrent and constraint logic programming, fifth generation computer, follow-on project, forecasts, overview, parallel inference system, personal sequential inference machine

Edited by: Anton P. Železnikar
Received: July 10, 1993 **Revised:** August 8, 1993 **Accepted:** August 16, 1993

In this article, we give a short overview of the FGCS project and describe the research and development of the sequential inference machine PSI. Then, we present our research results on constraint logic programming. Finally, we discuss our research activities in the field of parallel inference from both hardware and software aspects.

1 Overview of the FGCS Project

1.1 Preliminary Study Stage for the FGCS Project

The circumstances prevailing during the preliminary stage of the FGCS Project, from 1979 to 1981, can be summarized as follows[1]:

Japanese computer technologies had reached the level where they are now among the most up-to-date overseas computer technologies.

A change of the role of the Japanese national project for computer technologies was being discussed whereby there would be a move away from improvement of industrial competitiveness by catching up with the latest European computer technologies and toward world-wide scientific contributions through the development of leading computer technologies with all its inherent risks.

Regarding this situation, the Japanese Ministry of International Trade and Industry (MITI) began study on a new project—the Fifth Generation Computer Project. This term expressed MITI's commitment to developing leading technologies that would progress beyond the fourth generation computers due to appear in the near future and which would anticipate upcoming trends.

The Fifth Generation Computer Research Committee and its subcommittee were established in 1979. It took until the end of 1981 to decide on target technologies and a framework for the project. Well over one hundred meetings were held with a similar number of committee members participating. The following important near-future computer technologies were discussed:

- Inference computer technologies for knowledge processing
- Computer technologies to process large-scale data bases and knowledge bases
- High performance workstation technologies
- Distributed functional computer technologies
- Super-computer technologies for scientific calculation

These computer technologies were investigated and discussed from the standpoints of international contribution through the development of original Japanese technologies, the important technologies of the future, social needs and conformance with Japanese government policy for the national project.

Through these studies and discussions, the committee decided on the objectives of the project

[1]Similar paper with the same title was published in the Japan Computer Quarterly, No. 93, 1993. Permition for reprint given by the Japan Information Processing Development Center.

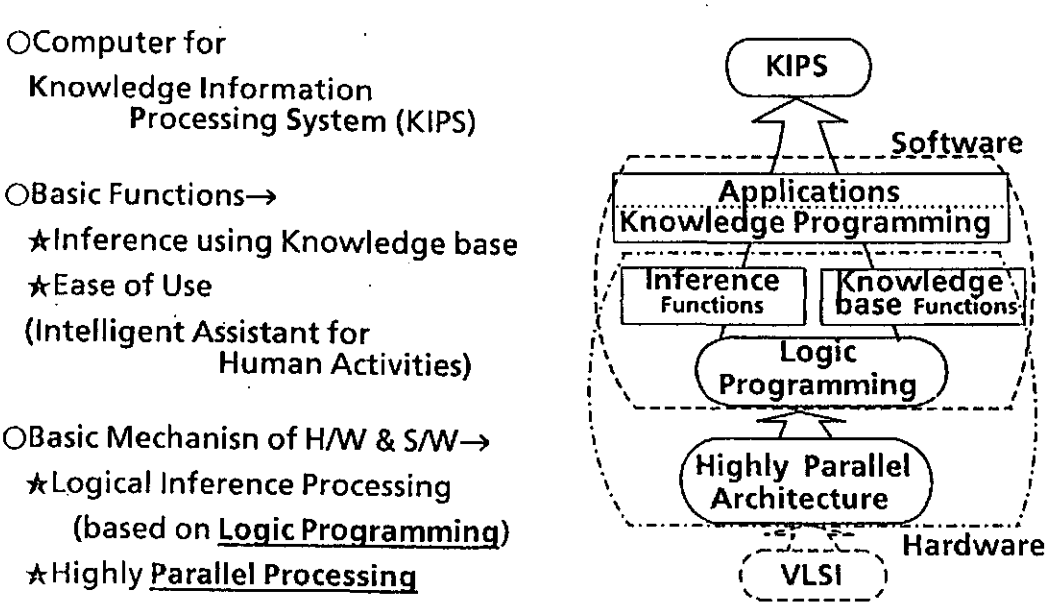


Figure 1: Concept of the Fifth Generation Computer

by the end of 1980, and continued future studies of technical matters, social impact, and project schemes.

The committee's proposals for the FGCS Project are summarized as follows:

(1) The concept of the Fifth Generation Computer: to have parallels (non–Von Neumann) processing and inference processing using knowledge bases as basic mechanisms. In order to possess these mechanisms, the hardware and software interface is to be a logic program language (see Figure 1).

(2) The objectives of the FGCS project: to develop these innovative computers which are capable of knowledge information processing and to overcome the technical restrictions of conventional computers.

(3) The goals of the FGCS project: to research and develop a set of hardware and software technologies for FGCS, and to develop an FGCS prototype system consisting of a thousand element processors with inference execution speeds of between 100M LIPS and 1G LIPS (Logical Inferences Per Second).

(4) R&D period for the project: estimated to be ten years, divided into three stages.

- 3-year initial stage for R&D of basic technologies
- 4-year intermediate stage for R&D of subsystems
- 3-year final stage for R&D of total prototype system

MITI decided to launch the Fifth Generation Computer System (FGCS) project as a national project for new information processing, and made efforts to acquire a budget for the project. At the same time, the international conference on FGCS' 81 was prepared and held in October 1981 to announce these results and to hold discussions on the topic with foreign researchers.

1.2 Stages and Budgeting in the FGCS Project

The FGCS project was designed to investigate a large number of unknown technologies that were yet to be developed. Since this involved a number of risky goals, the project was scheduled over a relatively long period of ten years. This ten-year period was divided into three stages.

- In the initial stage (fiscal 1982–1984), the purpose of R&D was to develop the basic computer technologies needed to achieve the goal.

- In the intermediate stage (fiscal 1984–1988), the purpose of R&D was to develop small to medium subsystems.

- In the final stage (fiscal 1989–1992), the purpose of R&D was to develop a total prototype system. The final stage was initially planned to be three years. After re–examination halfway through the final stage, this stage was extended to four years to allow evaluation and improvement of the total system in fiscal year 1992. Consequently, the total length of this project has been extended to 11 years.

Each year the budget for the following years R&D activities is decided. MITI made strenuous efforts in negotiating each year's budget with the Ministry of Finance. The budgets for each year, which are all covered by MITI, are shown in Figure 2. The total budget for the 3–year initial stage was about 8 billion yen. For the 4–year intermediate stage, it was approximately 22 billion yen. The total budget for 1989 to 1991 was around 21 billion yen. The budget for 1992 is estimated to be 3.6 billion yen. Consequently, the total budget for the 11–year period of the project will be about 54 billion yen.

1.3 Summary of the Project Research Results

In the Fifth Generation Computer Project, two main research targets were pursued: knowledge information processing and parallel processing. Logic programming was adopted as a key technology for achieving both targets simultaneously. At the beginning of the project, we adopted Prolog as our vehicle to promote the entire research of the project. Since there were no systematic research attempts based on Prolog before our project, there were many things to do, including the development of a suitable workstation for the research, experimental studies for developing a knowledge–based system in Prolog and investigation into possible parallel architecture for the language. We rapidly succeeded in promoting research in many directions.

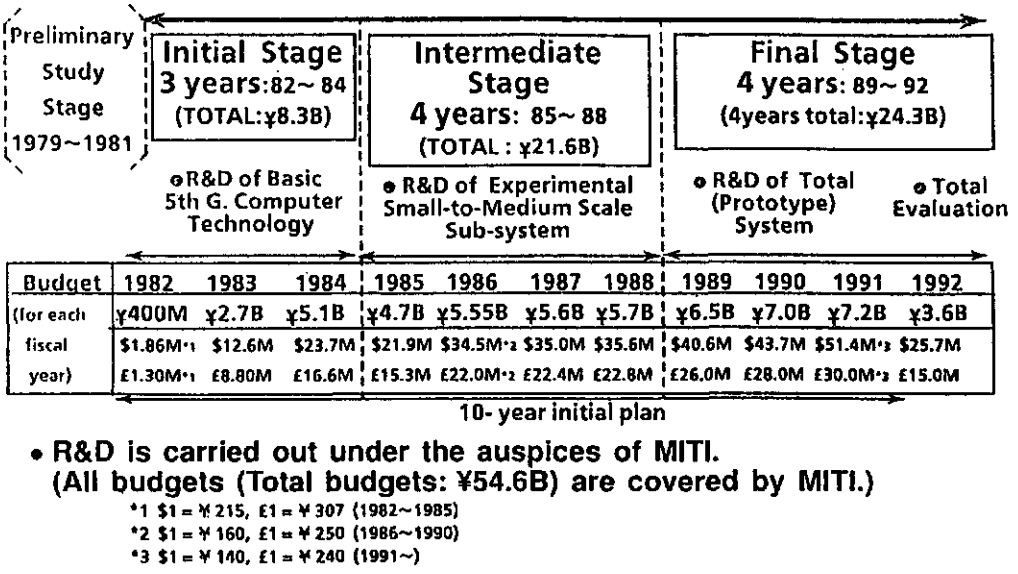
From this research, three achievements are worth noting. The first is the development of our own workstation dedicated to ESP: Extended Self-contained Prolog. We developed an operating system for the workstation completely in ESP [1].

The second is the application of partial evaluation to meta programming. This enabled us to develop a compiler for a new programming language and then partially evaluating it. We applied this technique to derive a bottom–up parser for context–free grammar given a bottom up interpreter for them. In other words, partial evaluation made meta programming useful in real applications.

The third achievement was the development of constraint logic programming languages. We developed two constraint logic programming languages: CIL and CAL. CIL is for natural language processing and is based on the incomplete data structure for representing "Complex Indeterminates" in situation theory. It has the capability to represent structured data like Minsky's frame and any relationship between slots' values can be expressed using constraints. CIL was used to develop a natural language understanding system called DUALS. Another constraint logic programming language, CAL, is for non–linear equations. Its inference is done using the Buchberger algorithm for computing the Grobner Basis which is a variant of the Knuth–Bendix completion algorithm for a term rewriting system.

We encountered one serious problem inherent to Prolog: that was the lack of concurrency in the fundamental framework of Prolog. We recognized the importance of concurrency in developing parallel processing technologies, and we began searching for alternative logic programming languages with the notion of concurrency.

We noticed the work by Keith Clark and Steve Gregory on Relational Language [2] and Ehud Shapiro on Concurrent Prolog [3]. These languages have a common feature of committed choice nondeterminism to introduce concurrency. We devoted our efforts to investigating these languages carefully and Ueda finally designed a new committed choice logic programming language called GHC [4, 5], which was simpler syntax than the above two languages but still has similar expressiveness. We recognized the importance of GHC and adopted it as the core of our kernel language, KL1, in this project. The introduction of KL1 made it possible to divide the entire research



Budget (for each fiscal year)	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992
	¥400M	¥2.7B	¥5.1B	¥4.7B	¥5.55B	¥5.6B	¥5.7B	¥6.5B	¥7.0B	¥7.2B	¥3.6B
	$1.86M*1	$12.6M	$23.7M	$21.9M	$34.5M*2	$35.0M	$35.6M	$40.6M	$43.7M	$51.4M*3	$25.7M
	£1.30M*1	£8.80M	£16.6M	£15.3M	£22.0M*2	£22.4M	£22.8M	£26.0M	£28.0M	£30.0M*3	£15.0M

Figure 2: Budgets of the FGCS project

project into two parts: the development of parallel hardware dedicated to KL1 and the development of software technology for the language. In this respect, the invention of GHC is the most important achievement for the success of the Fifth Generation Computer System project.

Besides this language oriented research, we performed extensive fundamental research in the field of artificial intelligence and software engineering based on logic and logic programming. This includes research on non–monotonic reasoning, hypothetical reasoning, abduction, induction, knowledge representation, theorem proving, partial evaluation and program transformation. We expected that this research would become important application fields for our parallel machines by the affinity of these problems with logic programming and logic–based parallel processing. This in now happening.

In this article, we first describe the research and development of the sequential inference machine PSI. Then, we present our research results on constraint logic programming. Finally, we discuss our research activities in the field of parallel inference from both hardware and software aspects.

2 FGCS Project Research Results

2.1 Personal Sequential Inference Machine (PSI-I)

To actually build the parallel inference system, especially a productive parallel programming environment which is now provided by PIMOS, we needed to develop various element technologies step by step to obtain hardware and software components. On the way toward this development, the most promising methods and technologies had to be selected from among many alternatives, followed by appropriate evaluation processes. To make this selection reliable and successful, we tried to build experimental systems which were as practical as possible.

In the initial stage, to evaluate the descriptive power and execution speed of logic languages, a personal sequential machine, PSI, was developed. This was a logic programming workstation. This development was also aimed at obtaining a common research tool for software development. The PSI was intended to attain an execution speed similar to DEC 10 Prolog running on a DEC20 system, which was the fastest logic programming system in the world.

To begin with, a PSI machine language, KL0,

was designed based on Prolog. Then a hardware system was designed for the KL0. We employed tag architecture for the hardware system. Then we designed a system description language, ESP, which is a logic language having a class and inheritance mechanisms to make program modules efficiently [6]. ESP was used not only to write the operating system for PSI, which is named SIMPOS, but also to write many experimental software systems for knowledge processing research.

The development of the PSI machine and SIMPOS was successful. We were impressed by the very high software productivity of the logic language. The execution speed of the PSI was about 35K LIPS and exceeded its target. However, we realized that we could improve its architecture by using the optimization capability of a compiler more effectively. We produced about 100 PSI machines to distribute as a common research tool. This version of the PSI is called PSI-I.

The implementation of the PSI-I hardware required 11 printed circuit boards. As the amount of hardware became clear, we established that we could obtain an element processor for a parallel machine if we used VLSI chips for implementation.

For the KL0 language processor which was implemented in the firmware, we estimated that better optimization of object code made by the compiler would greatly improve execution speed. (Later, this optimization was made by the introduction of the "WAM" code [7].

The PSI-II used VLSI gate array chips for its CPU. The size of the cabinet was about one sixth that of PSI-I. Its execution speed was 330K LIPS, about 10 times faster than that of PSI-I. This improvement was attained mainly through employment of the better compiler optimization technique and improvement of its machine architecture. The main memory size was also expanded to 320 MB so that prototyping of large applications could be done quickly.

2.2 Constraint Logic Programming

Constraint logic programming is one of the most promising areas in the field of logic programming. The domain of Prolog is extended to cover most AI problems. The objective is to combine constraint satisfaction and logic programming. The reasons for its success are 1) it is a straightforward extension of Prolog by extending the notion of unification to deal with constraint satisfaction, and 2) it extends the scope of declarative programming to a wider class of problems such as linear equations and inequations by dealing with them in a way uniform with unification between terms. From the constraint satisfaction viewpoint, it provides programming capability to the description of the problem based on constraint satisfaction. Jaffar and Lassez [11] gave criteria for a constraint logic programming system to inherit important aspects of logic programming such as soundness, completeness and fixpoint semantics. It is worth noting that constraint logic programming is derived by extending unification. We will discuss later how concurrent logic programming is derived by restricting unification.

We began our constraint logic programming research almost at the beginning of our project, in relation to the research on natural language processing. Mukai [8] developed a language called CIL (Complex Indeterminates Language) for the purpose of developing a computational model of situation semantics. A complex indeterminate is a data structure allowing partially specified terms with indefiniteness. During the design phase of the language, he encountered the idea of freeze in Prolog II by Colmerauer [9]. He adopted freeze as a proper control structure for our CIL language.

From the viewpoint of constraint satisfaction, CIL only has a passive way of solving constraint, which means that there is no active computation for solving constraints such as constraint propagation or solving simultaneous equations. Later, we began our research on constraint logic programming involving active constraint solving. The language we developed is called CAL. It deals with non–linear equations as expressions to specify constrains. Three events triggered the research: one was our preceding efforts on developing a term rewriting system called METIS for a theorem prover of linear algebra [10]. Another event was our encounter with Buchberger's algorithm for computing the Grobner Basis for solving non–linear equations. Since the algorithm is a variant of the Knuth–Bendix completion algorithm for a term rewriting system, we were able to develop the system easily from our experience of developing METIS. The third event was the development of the CLP(X) theory by Jaffar and

Lassez which provides a framework for constraint logic programming languages [11].

There is further remarkable research on constraint logic programming in the field of general symbol processing [12]. Tsuda developed a language called cu-Prolog. In cu-Prolog constraints are solved by means of program transformation techniques called unfold/fold transformation (these will be discussed in more detail later in this paper, as an optimization technique in relation to software engineering). The unfold/fold program transformation is used here as a basic operation for solving combinatorial constraints among terms. Each time the transformation is performed, the program is modified to a syntactically less constrained program. Note that this basic operation is similar to term rewriting, a basic operation in CAL. Both of these operations try to rewrite programs to obtain certain canonical forms. The idea of cu-Prolog was introduced by Hasida during his work on dependency propagation and dynamic programming [13]. They succeeded in showing that context-free parsing, which is as efficient as chart parsing, emerges as a result of dependency propagation during the execution of a program given as a set of grammar rules in cu-Prolog. Actually, there is no need to construct a parser. cu-Prolog itself works as an efficient parser.

Hasida [13] has been working on a fundamental issue of artificial intelligence and cognitive science from the aspect of a computational model. In this computation model of dynamic programming, computation is controlled by various kinds of potential energies associated with each atomic constraint, clause, and unification. Potential energy reflects the degree of constraint violation and, therefore, the reduction of energy contributes to constraint resolution.

Constraint logic programming greatly enriched the expressiveness of Prolog and is now providing a very promising programming environment for applications by extending the domain of Prolog to cover most AI problems.

2.3 Parallel Inference System

2.3.1 FGHC

The most important feature of FGHC is that there is only one syntactic extension to Prolog called the commitment operator and is represented by a vertical bar "|". A commitment operator divides an entire clause into two parts called the guard part (the left-hand side of the bar) and the body part (the right-hand side).

FGHC is a subset of GHC and allows only unification and test predicates to be written in its guard part to prevent the guard from being nested during execution. FGHC is more amenable to the process interpretation of programs than GHC and is expressive enough.

The guard of a clause has two important roles: one is to specify a condition for the clause to be selected for the succeeding computation, and the other is to specify a condition for the clause to be selected for the succeeding computation, and the other is to specify the synchronization condition. The general rule of synchronization in FGHC is expressed as dataflow synchronization. This means that computation is suspended until sufficient data for the computation arrives. In the case of FGHC, guard computation is suspended until the caller is sufficiently instantiated to judge the guard condition. For example, consider how a ticket vending machine works. After receiving money, it has to wait until the user pushes a button for the destination. This waiting is described as a clause such that "if the user pushed the 160-yen button, then issue a 160-yen ticket".

The important thing is that dataflow synchronization can be realized by a simple rule governing head unification which occurs when a goal is executed and a corresponding FGHC clause is called. This rule is stated as follows: the information flow of head unification must be one way, from the caller to the callee.

The caller corresponds to a job scheduler and the callee corresponds to workers in the office. The dataflow corresponds to the job orders from the job scheduler to workers. Each worker has a speciality and conditions about the jobs that can be done. In this case, each worker has to wait for the arrival of a detailed job order before starting work in order to check the conditions. Note that the information flow of job orders is one way—from the job scheduler to workers.

This principle is very important in two aspects: one in that the language provides a natural tool for expressing concurrency, and the other in that the synchronization mechanism is simple enough

Item	PIM/p	PIM/c	PIM/m	PIM/i	PIM/k
Machine instructions	RISC-type + macro instructions	Horizontal microinstructions	Horizontal microinstructions	RISC-type	RISC-type
Target cycle time	60 nsec	65 nsec	50 nsec	100 nsec	100 nsec
LSI devices	Standard cell	Gate array	Cell base	Standard cell	Custom
Process Technology (line width)	0.96 μm	0.8 μm	0.8 μm	1.2 μm	1.2 μm
Machine configuration	Multicluster connections (8 PEs linked to a shared memory) in a hypercube network	Multicluster connections (8 PEs + CC linked to a shared memory) in a crossbar network	Two-dimensional mesh network connections	Shared memory connections through a parallel cache	Two-level parallel cache connections
Number of PEs connected	512 PEs	256 PEs	256 PEs	16 PEs	16 PEs

Table 1: Budgets of the FGCS project

to realize very efficient parallel implementation.

2.3.2 KLI

KL1 (Kernel Language version 1) is an extension to FGHC and consists of two sublanguages, KL1c (KL1 core), and KL1p (KL1 pragma).

KL1 enables a process to observe and control the execution of other processes at the metalevel, a feature needed in developing our operating system PIMOS for Multi-PSI and PIM. KL1p is a sublanguage for annotating map information; allocation of processes to processors and priorities among processes.

While FGHC programs represent concurrency independently from the machine architecture on which they are executed, KL1p maps them onto parallel processors. Even if the FGHC programs express high potential parallelism, it does not necessarily mean that they run fast on existing parallel hardware. The important issues to be considered in achieving good performance on parallel hardware include: load balancing, locality of communication and priority control. Note that these issues affect only efficiency, not correctness. Although it is desirable to automate mapping, there is no universal way to do this appropriately. We succeeded in developing automatic load balancing.

Communication costs are very high, especially in distributed memory architecture. Primitive operations like unification are around 10 to 100 times slower when they are performed across the interconnection network. We need to take communication overheads into account when we distribute loads to processors. The locality of computation is important, and if granularity and increase locality by appropriately specifying load distribution using the annotation facilities in KL1p.

2.3.3 PIM Hardware and KL1 Language Processor

To find an optimal architectural design, we selected several important features, such as element processor architecture and network structure, and decided to build five PIM modules having different design choice. The main features of these five modules are listed in Table 1. The number of element processors required for each module was determined depending on the main purpose of the module. Large modules have 256 to 512 element processors, and were intended to be used for software experiments. Small modules have 16 or 20 element processors and were built for architectural experiments and evaluation.

All of these modules were designed to support KL1 and PIMOS, so that software researchers could run one program on the different modules and compare and analyze the behaviors of parallel program execution.

A PIM/m module employed architecture similar to the multi-PSI system. Thus, its KL1 language processor could be developed by simply modifying and extending that of the multi-

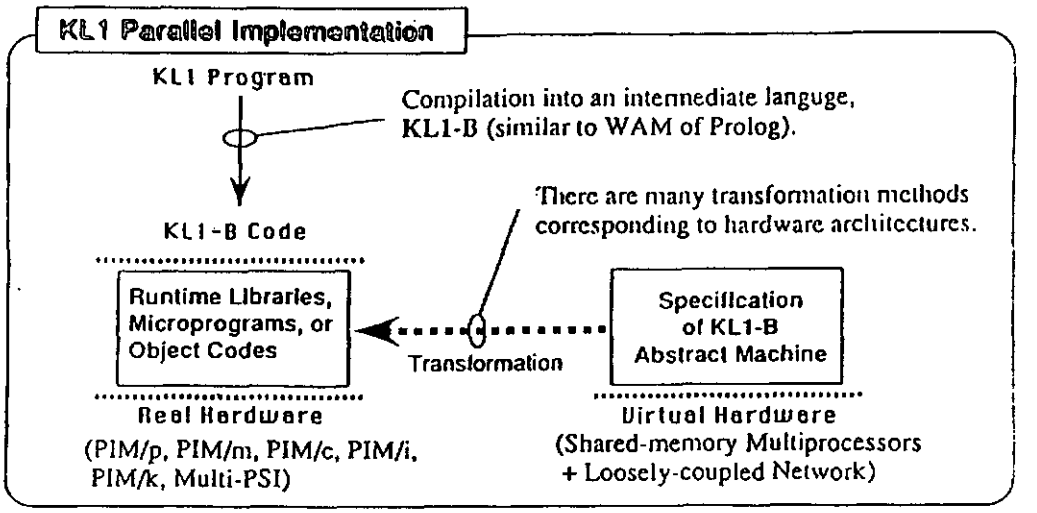


Figure 3: KL1 Language Processor and VPIM

PSI system. For other modules, namely PIM/p, PIM/c, PIM/k, and PIM/i, the KL1 language processor had to be newly developed because all of these modules have a cluster structure. In a cluster, four to eight element processors were tightly coupled by a shared memory and a common bus with coherent caches. While communication between element processors is done through the common bus and shared memory, communication between clusters is done via a packet switching network. These four PIM modules have different machine instruction sets.

We intended to avoid the duplication of development work for the KL1 language processor. We used the KL1-C language to write PIMOS and the usual application programs. A KL1-C program is compiled into the KL1-B language, which is similar to the "WAM" as shown in Figure 3. We defined an additional layer between the KL1-B language and the real machine instruction. This layer is called the virtual machine instruction at called "PSL". The specification of the KL1-B interpreter or runtime routines dedicated to each PIM modules. The specification in PSL is called a virtual PIM processor (the VPIM processor for short) and is common to four PIM modules.

PIM/p, PIM/m and PIM/c are intended to be used for large software experiments; the other modules were intended for architectural evaluations. We plan to produce a PIM/p with 512 element processors, and a PIM/m with 384 element processors. Now, at the beginning of March 1992, a PIM/m of 256 processors has just started to run a couple of benchmarking programs.

We aimed at a processing speed of more than 100 MLIPS for the PIM modules. The PIM/m with 256 processors will attain more than 100 MLIPS as its peak performance. However, for a practical application program, this speed may be much reduced, depending on the characteristics of the application program and the programming technique. To obtain better performance, we must attempt to augment the effect of compiler optimization and to implement a better load balancing scheme. We plan to run various benchmarking programs to evaluate the gain and loss of implemented hardware and software functions.

2.3.4 Development of PIMOS

PIMOS was intended to be a standard parallel operating system for large- scale parallel machines used in symbol and knowledge processing. It was designed as an independent, self-contained operating system with a programming environment suitable for KL1. Its functions for resource management and execution control of user programs were designed as independent from the architectural details of the PIM hardware. They were implemented based on an almost completely non-centralized management scheme so that the design could be applied to a parallel machine with one million element processors [15].

PIMOS is completely written in KL1. Its

management and control mechanisms are implemented using "meta-call" primitive of KL1. The KL1 language processor has an embedded automatic memory management mechanism and a dataflow synchronization mechanism. The management and control mechanisms are then implemented over these two mechanisms.

The resource management function is used to manage the memory resources and processor resources allocated to user processes and input and output devices. The program executing control function is used to start and stop user processes, control the order of execution following priorities given to them, and protect system programs from user program bugs like the usual sequential operating systems.

PIMOS supports multiple users, accesses via network and so on. It also has an efficient KL1 programming environment. This environment has some new tools for debugging parallel programs such as visualization programs which show a programmer the status of load balancing in graphical forms, and other monitoring and measurement programs.

2.3.5 Knowledge Base Management System

The knowledge base management system consists of two layers. The lower layer is a parallel database management system, Kappa-P. Kappa-P is a database management system based on a nested relational model. It is more flexible than the usual relational database management system in processing data of irregular sizes and structures, such as natural language dictionaries and biological databases.

The upper layer is a knowledge base management system based on a deductive object-oriented database [16]. This provides us with a knowledge representation language, Quixote [17]. These upper and lower layer are written in KL1 and are now operational on PIMOS.

The development of the database layer, Kappa, was started at the beginning of the intermediate stage. Kappa aimed to manage the "natural databases" accumulated in society, such as natural language dictionaries. It employed a nested relational model so that it could easily handle data sets with irregular record sizes and nested structures. Kappa is suitable not only for natural language dictionaries but also for DNA databases, rule databases such as legal data, contract conditions, and other "natural databases" produced in our social systems.

The first and second versions of Kappa were developed on a PSI machine using the ESP language. The second version was completed at the end of the intermediate stage, and was called Kappa-II [18].

In the final stage, a parallel and distributed implementation of Kappa was begun. It is written in KL1 and is called Kappa-P. Kappa-P is intended for the use of large PIM main memories for implementing the main memory database scheme, and to obtain a very high throughput rate for disk input and output by using many disks connected in parallel to element processors.

In conjunction with the development of Kappa-II and Kappa-P, research on a knowledge representation language and a knowledge base management system was conducted. After repeated experiments in design and implementation, a deductive object-oriented database was employed in this research.

At this point the design of the knowledge representation language, Quixote, was completed. Its language processor, which is the knowledge base management system, is under development. This language processor is being built over Kappa-P. Using Quixote, construction of a knowledge base can then be made continuously from a simple database. This will start with the accumulation of passive fact data, then gradually add active rule data, and will finally become a complete knowledge base.

The Quixote and Kappa-P system is a new knowledge base management system which has a high-level knowledge representation language and the parallel and distributed database management system as the base of the language processor. The first versions of Kappa-P and Quixote are now almost complete. It will be interesting to see how this large system operates and how much of an overhead it will require.

2.4 Concurrent Logic Programming

In this subsection, we first present the methodology to realize search paradigm in FGHC/KL1. Then, we describe three application programs in FGHC/KL1: a routing problem in VLSI CAD,

a sequence alignment problem in genetic information processing, and a bottom–up theorem prover. These items are very different from each other in their application areas as well as in the programming techniques used in their development.

2.4.1 Search Paradigms in FGHC/KL1

There is one serious drawback to FGHC/KL1 because of the very nature of committed choice; that is, it no longer has an automatic search capability, which is one of the most important features of Prolog. Prolog achieves its search capability by means of automatic backtracking. However, since committed choice uniquely determines a clause for succeeding computation of a goal, there is no way of searching for alternative branches other than the branch selected. The search capability is related to the notion of completeness of the logic programming computation procedure and the leak of this capability is very serious in that respect.

One could imagine a seemingly trivial way of realizing search capability by means of OR–parallel search: that is, to copy the current computational environment which provides the binding information of all variables that have appeared so far and to continue computations for each alternative case in parallel. But this does not work because copying non–ground terms is impossible in FGHC/KL1. The reason why it is impossible is that FGHC/KL1 cannot guarantee when actual binding will occur and there may be a moment when a variable observed at some processor remains unchanged even after some goal has instantiated it at a different processor.

One might ask why we did not adopt a Prolog–like language as our kernel language for parallel computation. There are mainly two reasons. One is that, as stated above, Prolog does not have enough expressiveness for concurrency, which we see as a key feature not only for expressing concurrent algorithms but also for providing framework for the control of physical parallelism. The other reason is that the execution mechanism of Prolog–like languages with a search capability seemed too complicated to develop efficient parallel implementations.

We tried to recover the search capability by devising programming techniques while keeping the programming language as simple as possible. We succeeded in inventing several programming methods for computing all solutions to a problem which effectively achieve the completeness of logic programming. Three of these methods are listed as follows:

(1) Continuation-based method [19]

(2) Layered stream method [20]

(3) Query compilation method [21]

In this paper we pick up (1) and (3), which are complementary to each other. The continuation–based method is suitable for the efficient processing of rather–algorithmic problems. An example is to compute all ways of partitioning a given list into two sublists by using append. This method mimics the computation of OR–parallel Prolog using AND–parallelism of FGHC. AND–serial computation in Prolog is translated to continuation processing which remembers continuation points in a stack. The intermediate results of computation are passed from the preceding goals to the next goals through the continuation stack kept as one of the arguments of the FGHC goals. This method requires input/output mode analysis before translating a Prolog program into FGHC. This requirement makes the method impractical for databases applications because there are too many possible input-output modes for each predicate.

The query compilation method solves this problem. This method was first introduced by Fuchi [22] when he developed a bottom–up theorem prover in KL1. In this coding technique, the multiple binding problem is avoided by reversing the role of the caller and the callee in straightforward implementation of database query evaluation. Instead of trying to find a record (represented by a clause) which matches a given query pattern represented by a goal, his method represents each query component with a compiled clause, represents a database with a data structure passed around by goals, and tries to find a query component clause which matches a goal representing a record and recourses the process for all potentially applicable records in the database[2]. Since every record is a ground term, there is no variable

[2] We need an auxiliary query clause which matches every record after failing to match the record to all the real query clauses.

in the caller. Variable instantiation occurs when query component clauses are searched and an appropriate clause representing a query component is found to match a currently processed record. Note that, as a result of reversing the representation of queries and databases from straightforward representation, the information flow is now from the caller (database) to the callee (a query component). This inversion of information flow avoids deadlock in query processing. Another important trick is that each time a query clause is called, a fresh variable is created for each variable in the query component. This mechanism is used for making a new environment for each OR-parallel computation branch. These tricks make it possible to use KL1 variables to represent object level variables in database queries and, therefore, we can avoid different compilations of the entire database and queries for each input/output mode of queries.

The new coding method stated above is very general and there are many applications which can be programmed in this way. The only limitation to this approach is that the database must be more instantiated than the queries. In bottom-up theorem proving, this requirement is referred to as the range-restrictedness of each axiom. Range-restrictedness means that, after successfully finding ground model elements satisfying the antecedent of an axiom, the new model element appearing as the consequent of the axiom must be ground.

This restriction seems very strong. Indeed, there are problems in the theorem proving area which do not satisfy the condition. We need a top-down theorem prover for such problems. However, many real life problems satisfy the range-restrictedness because they almost always have finite concrete models. Such problems include VLSI-CAD, circuit diagnosis, planning, and scheduling.

2.4.2 A Routing Problem in VLSI CAD

The aim the routing problem is to determine connection paths between terminals of circuit blocks on a VLSI surface. Well-known routing methods include maze routing, line searching and channel routing. We adopted an extended line searching algorithm called look-ahead line searching [23].

Before introducing the details of the algorithm and its implementation in KL1, let us explain the problem more precisely as illustrated in Figure 4. We assume that there are two layers for connection in VLSI chips: the first surface and the second surface. Let us assume that the first surface is for vertical lines and the second for horizontal lines. There are two kinds of constraints to connection: the first are surface obstacles like Block 1, Block 2 and Block 3 in Figure 4, inside which wiring is prohibited, the other are via-hole constraints which prevent use of via-hole constraints to change direction (go from one surface to the other). A routing problem is given by a set of nets, which are sets of terminals to be connected (a net is shown by a set of terminals with the same number as in Figure 4).

The look-ahead line searching method tries to find a route for a given pair incrementally by finding the best position to turn based on a local estimate. Estimation is done by computing the next nearest points to the target point for each possible turn and selecting the closest as the best. Since the method tries to compute all possible turns to determinate the next turn, it is called look-ahead line searching.

To implement the look-ahead line searching method in KL1, we adopted the object-oriented programming paradigm. Each line segment is represented by an object. To determinate the turning point of the current line segment, a message to compute the distance between the nearest attainable point and the target point is sent to all candidate lines intersecting the current line. After receiving all answers from all candidates, the current line object determines the point having the shortest distance to the target as the next turning point. Then, the selected line segment becomes the new current line segment and the same process is repeated.

A line object is realized in KL1 by a recursive program like filter in Erastothenes' sieve algorithm. Note that a current line segment is dynamically divided into a fixed route part and unused (free) parts. Preliminary evaluation showed 16-fold speed-up with 64 processors (compared to a single processor), and comparable execution time with a high-end general purpose computer. The speed will be 20 times faster on PIM/p.

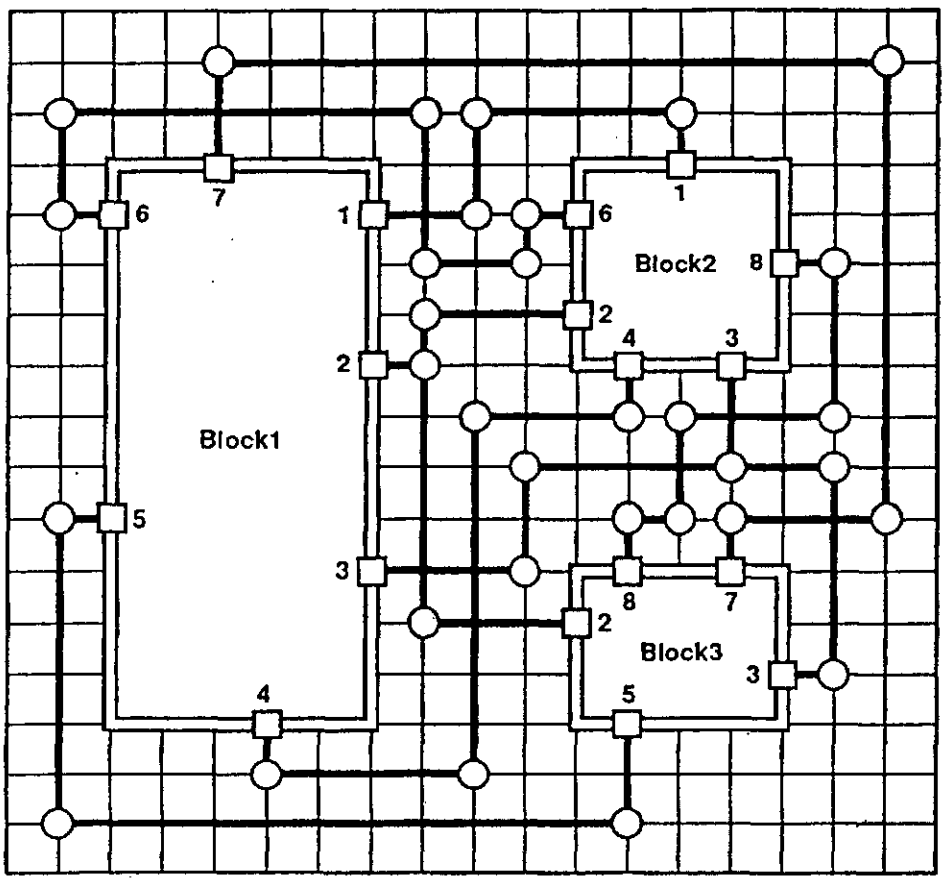


Figure 4: A Routing Problem and Its Solution

2.4.3 Sequence Alignment in Genome Analysis

Sequence alignment is a very important yet time consuming task in genetic information processing. It is difficult to find the best alignment of amino-acid sequences for more than one protein. There are two ways of aligning different sequences: one is to match two different amino-acids with a cost associated to each pair of amino-acids reflecting the similarity between them, the other is to match an amino-acid to a gap. Generally, the cost of matching two different amino-acids is less than that of matching an amino-acid to a gap. A well-known algorithm for aligning two amino sequences is two dimensional dynamic programming (DP) matching. Let us explain the algorithm for a simple case of aligning two four-letter sequences: ADHE and AHHE. The problem is formulated as the problem of finding the shortest path from the top-left corner to the bottom-right corner in the graph in Figure 5.

The DP matching algorithm proceeds from the top-left corner to the bottom-right corner. For each (ij) node, it computes the distance, D_{ij}, of the two sequences from the top-left corner to the point using the following formula:

$$D_{ij} = \min \begin{pmatrix} D_{i,j-1} + \text{gapcost}, \\ D_{i-1,j} + \text{gapcost}, \\ D_{i-1,j-1} + \\ \text{Dayhoff\_matrix}(X, Y) \end{pmatrix}$$

where X and Y are amino-acids associated with the arc from the node of $(i-1, j-1)$ to that of (i, j), and the Dayhoff-matrix is a cost matrix of pairs of amino-acids. Note that this formula represents a simple expression of dynamic programming which can compute each cost locally and prunes possibilities other than the minimum one.

Implementation of this algorithm in KL1 is straightforward: first, a set of processes representing each note is created, and then each D_{ij}

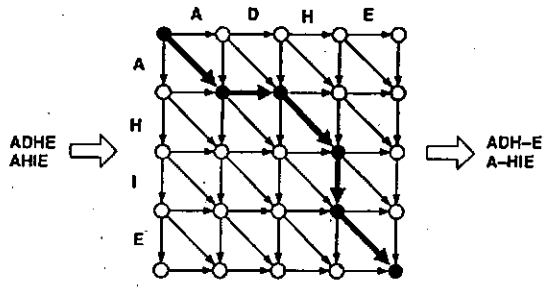


Figure 5: Two Dimensional DP Matching for Shortest Path Finding

is computed by this expression in a data driven style from top–left to bottom–right.

We developed a three–dimensional DP matching program in KL1 [24]. Ideally we need N dimensional DP matching where N is the number of sequences to be aligned. However, the computational complexity grows exponentially and, therefore, such an extension is not feasible. Thus, we tried to merge multiple three–dimensional alignments by aligning similar sequences of different gaps to maximize alignment scores. Furthermore, we applied a simulated annealing procedure to avoid the local optimum and attain a further increase in alignment scores. The improvement in alignment scores by simulated annealing was surprisingly good, resulting in a relatively short execution time.

2.4.4 Theorem Prover

Since head unification is limited to one–way in GHC/KL1, it is generally difficult to implement a first order theorem prover since these usually require the extensive use of full unification. There are two ways to solve this problem. One is to write a full unification program in GHC/KL1 and regard the language as a very low–level language like C. The result is not very attractive because it decreases efficiency by 10 to 100 times that with direct use of unification (in the case of one way unification).

The other is to limit the usage of unification to only one way. Manthey and Bry proposed a new bottom up theorem prover, called SATCHMO, based on model generation [25]. SATCHMO tries to generate all possible models incrementally by bottom–up evaluation of clauses. It tries to prove the antecedent conjunction of literals by searching for their instances in each model. If it succeeds in finding a model in which the proof succeeds, then it extends the model by adding the consequent disjunctive ground literals. If there is more than one literal in its disjunct, then SATCHMO splits the model into the number of literals in the disjunct, and adds each literal to each split model. Since every element in every model is ground, we need only one–way unification to search models during the proof of antecedent conjuncts. This enables us to implement an efficient theorem prover in GHC/KL1.

One significant problem was how to implement multiple bindings for finding all possible proofs by different instantiations of variables. In the case of Prolog, this function is achieved by utilizing backtracking. However, there is no backtracking mechanism in GHC/KL1. Fuchi [26] invented an elegant coding technique when implementing SATCHMO in FGHC. Fuchi utilized FGHC variables as object variables appearing in the theories to be proved. This was improved by Fujita and Hasegawa [27]. In their coding technique, the multiple binding problem is avoided by reversing the role of the caller and the callee in naive implementation of database query evaluation: instead of trying to find a model element (a database item) with a pattern appearing in a theorem (a query pattern), their method tries to find a theorem (a query component) with a given model element (a database item) as an instantiation of a literal of the theorem. Since every model element is a ground literal, there is no variable in the caller. The variable instantiations occur when a theorem database is searched and an appropriate clause representing a literal of some theorem is found to match a given model element.

We tried to apply this theorem prover to solve several hard problems on our PIM with 256 processor and, for some problems, we obtained approximately 200 times performance improvement compared to the run time on a single processor.

We have been applying the theorem prover to such divergent problems as mathematical theorem

proving, legal reasoning, design problems and syntax analysis. In particular, we have succeeded in solving several theorems of semigroup which were not solved earlier.

3 FGCS Follow–on Project & Forecasts

3.1 FGCS Follow–on Project

As described in the above, the FGCS Follow–on Project is a two project which runs from the beginning of fiscal 1993 until the end of fiscal 1994. A major role of the FGCS Follow–on Project is to promote a diffusion of parallel knowledge processing technologies that have been developed in the FGCS Project.

Much of the KL1 software which aims at the provision of the new infrastructure for advanced computer research has been developed for research on parallel knowledge processing technologies in the FGCS Project. Moreover, the major software has been released as IFS. However, a sequential inference machine, PSI, or parallel inference machine, Multi-PSI or PIM, is required to execute the software. Though a "Pseudo Multi-PSI", that is, a pseudo parallel system for KL1 software, has been released as IFS, a PSI-III is required to execute it. A "PDSS", that is a KL1 programming environment on UNIX machines, has also been released as IFS, there are some limits to its efficiency and functions for executing KL1 software on it. Thus, although the PDSS system is suitable for learning KL1 language, it cannot be used to execute large KL1 software released as IFS. Therefore, it is difficult to execute the KL1 software released as IFS at hand.

In the FGCS Follow–on Project a series of KL1 programming environments, including a KL1 language processor and a parallel operating system, PIMOS, shall be ported onto sequential and parallel UNIX machines so that they can be used easily at any site. These UNIX based KL1 programming environments shall be designed as machine–independent as possible. They are also planned to be released as IFS.

An experimental version of the UNIX based KL1 programming environment is currently under development for evaluating an implementation scheme. Although we plan to release it as IFS in April '93, this version is for language implementation experts and is not suitable for application users since it lacks some important features for application users, such as debugging aids.

The first version for application users is planned release in September '93. This version shall provide reasonable software development functions, including debugging and performance analyses. This system shall be ten times faster than the PDSS, although it is for single processor UNIX machines.

The release of KL1 programming environment for parallel UNIX machines is planned for the second quarter of '94. It shall be designed avoiding the use of machine–dependent functions. Various improvements are planned after these releases.

3.2 Forecasts for Some Aspects of 5G Machines

LSI technologies have advance in accordance with past trends. Roughly speaking, the memory capacity and the number of gates of a single chip quadruple every three years. The number of boards for the CPU of an inference machine was more than ten for PSI-I, but only three for PSI-II, and a single board for PIM.

The number of boards for 80M bytes memory was 16 for PSI-I, but only four for PSI-II, and a single board for PIM (m).

Figure 6 shows the anticipated trend for board numbers for one PE (processor element: CPU and memory) and the cost of one PE based on the actual value of inference machines developed by this project.

The trend reveals that by the year 2000 approximately ten PEs will fit on one board, around 100 PEs will fit in one desk side cabinet, and 500 to a 1,000 PEs will fit into a large cabinet. This trend also shows that the cost of one PE will have every three years.

Figure 7 shows the performance trends for 5G machines based on the actual performance of inference machines developed by this project.

The sequential inference processing performance for one PE quadrupled every three years. The improvement in parallel inference processing performance for one PE was not as large as it was for sequential processing, because PIM performance is estimated at around two and one half times that of multi–PSI. Furthermore, Figure

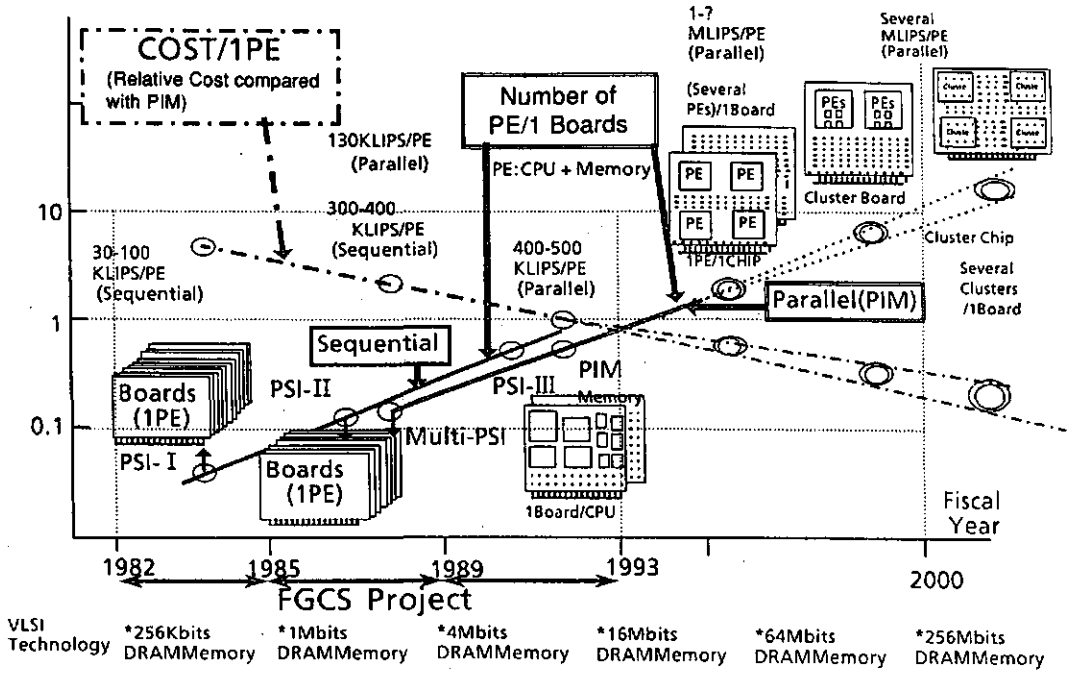


Figure 6: Size and Cost Trends of 5G Machines

7 shows the performance of one board for both sequential and parallel processing, and the performance of a conventional micro-processor with CISC and RISC technology. In this figure, future improvements in the performance of one PE are estimated to be rather lower than a linear extension of past values would indicate because of the uncertainty of whether future technology will be able to elicit such performance improvements. Performance for one board is estimated at about 20 MLIPS, which is 100 times faster than PIM. Thus, a parallel machine with a large cabinet size could have 1 GLIPS. These parallel systems will have the processing speeds needed for various knowledge processing applications in the near future.

Several parallel applications in this project, such as CAD, theorem provers, and genetic information processing, natural language processing, and legal reasoning were described previously. These applications are distributed in various fields and aim cultivating new parallel processing application fields.

We believe that parallel machine applications will be extended to various areas in industry and society, because parallel technology will become common for computers in the near future. Parallel application fields will expand gradually according to function expansion by the use of advanced parallel processing and knowledge processing technologies.

References

[1] T. Chikayama, Programming in ESP-Experiences with SIMPOS. In Programming of Future Generation Computers, Fuchi and Nivat (eds.), North-Holland, 1988.

[2] K.L. Clark and S.Gregory, A Relational Language for Parallel Programming. In Proc. ACM Conf. on Functional Programming Languages and Computer Architecture, ACM, 1981.

[3] E.Y. Shapiro, A Subset of Concurrent Prolog and Its Interpreter. Tech. Report TR-003, Institute for New Generation Computer Technology, Tokyo, 1983.

[4] K. Ueda, Guarded Horn Clauses. In Logic Programming '85, E. Wada (ed.), Lecture Notes in Computer Science, 221, Springer-Verlag,1986.

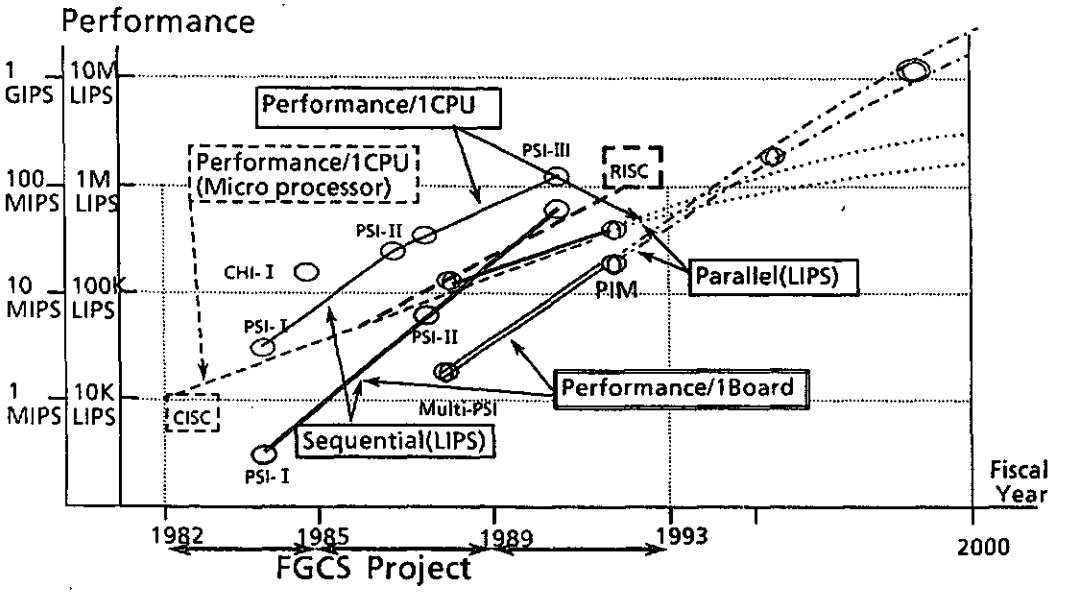


Figure 7: Performance Trends of 5G Machines

[5] K. Ueda and T. Chikayama, Design of the Kernel Language for the Parallel Inference Machine. The Computer Journal, Vol. 33, No. 6, pp.494–500, 1990.

[6] T. Chikayama, Unique Features of ESP. In Proc. Int. Conf. on Fifth Generation Computer Systems 1984, ICOT, 1984, pp.292–298.

[7] D.H.D. Warren, An Abstract Prolog Instruction Set. Technical Note 304, Artificial Intelligence Center, SRI, 1983.

[8] K. Mukai and H. Yasukawa, Complex Indeterminates in Prolog and its Application to Discourse Models. New Generation Computing, Vol.3, No.4, 1985.

[9] A. Colmerauer, Theoretical Model of Prolog II. In Logic Programming and Its Applications, M. Van Caneghem and D.H.D. Warren (eds.), Albex Publishing Corp., 1986.

[10] A. Ohsuga and K. Sakai, Metis: A Term Rewriting System Generator. In Software Science and Engineering, I. Nakata and M. Hagiya (eds.), World Scientific, 1991.

[11] J. Jaffar and J.-L. Lassez, Constraint Logic Programming. Technical Report, Department of Computer Science, Monash University, 1986.

[12] H. Tsuda, cu–Prolog for Constraint–based Grammar. In Proc. of the International Conf. on Fifth Generation Computer Systems 1992, Tokyo, 1992.

[13] K. Hasida, Dynamics of Symbol Systems—An Integrated Architecture of Cognition. In Proc. of the International Conf. on Fifth Generation Computer Systems 1992, Tokyo, 1992.

[14] M. Furuichi, K. Taki, and N. Ichiyoshi, A Multi–Level Load Balancing Scheme for OR–Parallel Exhaustive Search Programs on the Multi–PSI. In Proc. of the 2nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, 1990.

[15] T. Chikayama, Operating System PIMOS and Kernel Language KL1. Proc. Int. Conf. on Fifth Generation Computer Systems, Tokyo, Jul. 1–5, 1992.

[16] K. Yokota and S. Nishio, Towards Integration of Deductive Databases and Object–Oriented Databases—A Limited Survey. Proc. Advanced Database System Symposium, Kyoto, Dec., 1989.

[17] K. Yokota and H. Yasukawa, Towards an Integrated Knowledge-Base Management System. Proc. Int. Conf. on Fifth Generation Computer Systems, Tokyo, Jul. 1-5, 1992.

[18] K. Yokota, M. Kawamura, and A. Kanaegami, Overview of the Knowledge Base Management System (KAPPA). Proc. Int. Conf. on Fifth Generation Computer Systems, Tokyo, Nov. 28-Dec. 2, 1988.

[19] K. Ueda, Making Exhaustive Search Programs Deterministic. In Proc. of the Third Int. Conf. on Logic Programming, Springer-Verlag,1986.

[20] A. Okamura and Y. Matsumoto, Parallel Programming with Layered Streams. In Proc. 1987 International Symposium on Logic Programming, pp. 224-232, San Francisco, Sep. 1987.

[21] K.Furukawa, Logic Programming as the Integrator of the Fifth Generation Computer Systems Project. Comm. of the ACM, Vol. 35, No.3, 1992.

[22] K. Fuchi, An Impression of KL1 Programming—from my experience with writing parallel provers—. In Proc. of KL1 Programming Workshop '90, ICOT, 1990 (in Japanese).

[23] H. Date, Y. Ohtake, and K. Taki, A Parallel Router Based on a Concurrent Object Model. In Proc. of the Logic Programming Conference, ICOT, 1991 (in Japanese).

[24] M. Ishikawa, M. Hoshida, M. Hirosawa, T. Toya, K. Onizuka, K. Nitta, and M. Kanehisa, Protein Sequence Analysis by Parallel Inference Machine. Technical Report TR-681, ICOT, 1991 (in Japanese).

[25] R. Manthey and F. Bry, SATCHMO: A Theorem Prover Implemented in Prolog. In Proc. of CADE-88, Argonne, Ill., 1988.

[26] K. Fuchi, An Impression of KL1 Programming—from my experience with writing parallel provers—. In Proc. of KL1 Programming Workshop, ICOT, 1990 (in Japanese).

[27] J. Fujita and R. Hasegawa, A Model Generation Theorem Prover in KL1 Using a Ramified-Stack Algorithm. In Proc. of the Eight International Conference on Logic Programming, Paris,1991.

Reports and Announcements

Report: AAAI'93

11–15 July, Washington, DC, USA

Sašo Džeroski

The National Conference on Artificial Intelligence is the major yearly event in Artificial Intelligence in the USA (proceedings with AAAI Press/MIT Press). The Eleventh National Conference on Artificial Intelligence was held in the Washington Convention Center in the capital city of the USA, Washington, DC. Over 2000 persons attended the conference, which was preceeded by a wide range of workshops and tutorials. Among the workshops, let us mention the *AI in Business*, *Reasoning About Function* and the *Knowledge Discovery in Databases* workshops. Of the tutorials, let us mention the *Mobile Robots*, *Multistrategy learning* and *Genetic Algorithms and Genetics-Based Machine Learning*.

The *Knowledge Discovery in Databases* (KDD) workshop was attended by over 60 people. The 28 papers in the workshop proceedings (available as a technical report from AAAI Press) were organized in five parts: real-world applications, discovery of dependencies and models, integrated and interactive systems, database specific techniques and discovery in textual documents. Nine papers were presented in three sessions corresponding to the first three parts. In addition, an invited talk, titled *Inductive Logic Programming for Knowledge Discovery in Databases: An Overview* was given by Džeroski. The talk emphasized that inductive logic programming can be used for knowledge discovery of patterns that involve several relations in a database, as opposed to most current efforts that only consider patterns within a single relation. An overview of empirical ILP techniques was given, including the issue of handling imperfect (noisy) data. The talk also gave a summary of recent advances in ILP relevant to knowledge discovery in databases, such as multiple predicate learning and discovery of clausal integrity constraints. There was a lively discussion after the talk and especially strong interest in the potential of ILP for practical applications in KDD.

The applications papers described a system for cataloguing sky objects and the problems and prospects of image database exploration (Fayyad and Smyth), the selection among rules induced from a hurricane database (Major and Mangano), and the use of templates in knowledge discovery (Anand and Kahn). In addition, there was an application posters and system demonstration session. The most interesting poster by Ziarko et al. described an application of a discovery tool to identifying strong predictive rules in stock market data. Among the dependency and model discovery papers, Džeroski and Todorovski addressed the task of learning qualitative models from example behaviours, a task also addressed by several ILP systems. Savnik and Flach described an efficiency improvement to an earlier algorithm for induction of functional dependencies due to Flach. The workshop concluded with a panel, titled *Progress and Remaining Obstacles for KDD*, the main conclusions of which were that there has been good progress in applications of KDD, detecting dependencies and models and building efficient propositional induction methods, but much work remains to be done on integrating multiple types of discovery, building interactive discovery systems, as well as discovery in text, complex data and knowledge bases. A special issue of the *Journal of Intelligent Information Systems* (Kluwer Academic Publishers), based on papers presented at the workshops, will be edited by Piatetsky-Shapiro, the Workshop chair.

The tutorial on *Genetic Algorithms and Genetic-Based Machine Learning* by Goldberg and Koza gave a gentle introduction to genetic algorithms, followed by some practical genetic algorithms (GA) theory, an overview of more advanced GA techniques (such as genetic programming) and a parade of applications. Most relevant to ILP is the genetic programming paradigm by Koza, where LISP programs to perform specific tasks are evolved. Examples of input/output pairs are used to calculate the appropriateness (fitness) of all LISP programs in the evolving population. A similar approach to the evolution of logic programs (*Genetic ILP* or GILP) has been recently taken by Varšek.

The main part of the conference started on Monday (12 July) evening, with the invited talk by Feigenbaum, titled *Tiger in a Cage: The Applications of Knowledge-Based Systems*. As the conference was held in the USA capital city, the talk gave many examples of successful applications of AI in government- supported agencies. Of these, let us mention the applications deployed in NASA, which include GPSS (Ground Processing Scheduling System) that schedules the Kenedy Space Center operations between consequtive flights of the space shuttle and Principal Investigator In-a-Box, a system for planning experiments on the space shuttle. DARPA (Defense Advanced Research Project Agency) has stated that all of its investment in AI over the years has been paid back by the DART logistic system, used to schedule logistic operations during the Desert war. Examples of expert systems that embody knowledge about specific areas of legislature and give advice to citizens regarding their specific problems. According to Feigenbaum, there is an enormous potential for government applications of this kind. The second part of the talk summarized the problems encountered so far with the development and deployment

of knowledge-based systems: too little knowledge in products such as shells, ease of use of knowledge-based systems, maintenance problems, lack of standards, etc. All in all, this was a good political talk addressed mostly to the government officials present, with just enough criticism to be realistic and enough promise to attract government funding for the field.

The next invited talk at the conference was given by the Nobel laureate (Nobel prize for economy) who has been working in AI for decades, Herbert Simon. The talk, titled *Artificial Intelligence as an Experimental Science*, argued for more experimental work in AI, as a complement to recent trends (as exemplified by publications in the scientific journal *Artificial Intelligence*) of formal theoretical work in the field. Among the many other invited talks, let us mention the overviews of research in the fields of nonmonotonic reasoning (given by Reiter), a type of reinforcement learning called temporal-difference learning (given by Sutton) and qualitative physics (given by Forbus).

Paper presentations ran in at least three (sometimes five) parallel sessions. There were sessions on diagnostic reasoning, reasoning about physical systems, natural language, vision processing, machine learning, planning, knowledge acquisition, complexity in machine learning, heuristic search, and many others. Of 126 papers in the proceedings, about ten papers are on machine learning. Of these, five can be considered inductive logic programming papers. Three of the four papers in the session on complexity in machine learning were discussing learnability results for logic programs. Cohen first presented some negative results: he showed that determinate clauses of log depth are not learnable, recursive clauses of constant depth are not learnable and indeterminate clauses with k "free" variables are exactly as hard to learn as DNF. Based on the technique of relative least general generalization, he than proves that two-cause definitions consisting of a base clause and an ij-determinate linear recursive clause are learnable from examples and queries. One type of queries gives an upper bound on the depth of the proof for and example and the other type of queries tells whether an example can be proved in unit depth.

Frazier and Page proved that concepts expressed by at most two Horn clauses, each clause having unary predicates only, at most one literal in the body and at most unary function symbols, are learnable. If the predicate arity restriction is removed, the resulting concept class is not learnable. However, if predicate arity is bounded by a constant k, then the resulting class is PAC-predictable, i.e., PAC-learnable in terms of a slightly more general concept class. The question of whether it is PAC-learnable is left open.

FOCL is a learning system that combines ILP and EBL (explanation based learning). Pazzani and Brunk described a modified version of FOCL which does not entirely operationalize the induced concept, but rather uses an information theoretic metric to decide whether to operationalize or not parts of the concept description. Zelle and Mooney described an application of inductive logic programming to learning semantic grammars. Their system CHILL also invents new predicates that correspond to syntactic and semantic classes of words and phrases. Experiments on two reasonably large corpora of sentence/case-role pairs demonstrate that the system learns accurate parsers that generalize well to novel sentences and creates interesting and recognizable syntactic and semantic concepts. The paper by Neville and Weld on innovative design as systematic search is related to ILP insofar as innovative design has been recognized as a potential application area for ILP by Bratko.

Summarizing more of the over 120 papers would make this report too long. Let us now mention some of the activities that took part in parallel with the conference. On July 13th and 14th, The Fifth Conference on Innovative Applications of AI (Intelligent Systems at Work) was held in the same building. In addition, there was an exhibition of AI related books and software throughout the conference. There was also a robot competition and a robot building laboratory. Although the robot building fans probably had a very good time, they missed most of the conference, spending long hours to build their robots. At the end, there was a competition for the small robots built by the robot building laboratory participants, titled *Escape from The Office*. It was great fun. To wrap up, the conference was a very exciting event.

Report: ML'93

27 June - 1 July, Amherst, MA, USA

Sašo Džeroski

The International Conference on Machine Learning is the major yearly event in Machine Learning (proceedings with Morgan Kaufmann). The Tenth International Conference on Machine Learning was organized by Paul Utgoff at the University of Massachusetts at Amherst, a town in rural Western Massachussetts. Over 200 persons attended the conference, which comprised 4 invited talks and 44 papers, presented in 3 plenary sessions and 4 sessions of 3 parallel tracks. Most areas of machine learning were represented, including reinforcement learning, neural networks, genetic algorithms, explanation based learning, Bayesian networks, learnability, inductive learning, machine discovery, etc.

The invited talks were given by Leo Breimann, Micki Chi, David Sandler and Pat Langley. Leo Breimann talked about statistical methods and their use in machine learning, Micki Chi talked about conceptual change, David Sandler described an application of neural networks for adjusting telescope mirrors to account for atmospheric disturbances, and Pat Langley gave an overview of the developments in the field of machine learning in the last ten years, concluding that machine learning has in many ways become a mature discipline.

There were two ILP papers at the conference. Peter Idestam-Almquist (*Generalization under Implication by Recursive Anti-unification*) presented a technique, called recursive anti-unification, that computes least general generalizations under implication. This is important as generalization under θ-subsumption, performed by many ILP systems, is incomplete with respect to implication. Datta and Kibler (*Concept Sharing: A Means to Improve Multi-Concept Learning*) presented M-FOCL, a system that learns multiple predicates. Their thesis was that re-use of (parts of) already learned predicate definitions as background knowledge for learning new predicates from the same domain can greatly facilitate the learning process. Experimental evaluation confirms that both classification accuracy (in the presence of noise) and predicate definition size are improved if parts of previously learned definitions are used. These include single clauses or conjuncts that appear in more than one clause.

Several other papers were related to topics considered within the ILP Project. Schlimmer (*Efficiently Inducing Determinations: A Complete and Sytematic Search Algorithm that Uses Optimal Pruning*) repeated some of the work on discovering functional dependencies done within the INDEX system by Flach. Frazier and Pitt (*Learning From Entailment: An Application to Propositional Horn Sentences*) proved that a formula equivalent to a target theory in the form of a propositional Horn sentence can be learned in polynomial time using equivalence queries and membership queries (of entailed clauses). Tadepalli (*Learning from Queries and Example with Tree-structured Bias*) proved that functions consistent with a tree-structured bias can be PAC-learned from examples and queries, provided that nodes in the determination tree have a small branching factor. The two papers on learnability are loosely related to the work on PAC-learnability of logic programs by Džeroski, Muggleton and Russell. Džeroski and Todorovski (*Discovering Dynamics*) presented an approach to finding a set of algebraic and differential equations that are consistent with a given behaviour of a dynamic system. This approach can be integrated with the clausal discovery engine of De Raedt and Bruynooghe to allow for numerical constraints in the discovered clauses.

Of the remaining papers, several were devoted to the combining of different learning methods to achieve better performance. Quinlan (*Combining Instance-Based and Model-Based Learning*) presented a method that combines the 3-NN method with model trees (an extension of regression trees). The combined method performs better than either of the components, as well as better than neural networks and simple linear regression. Brodley (*Addressing the Selective Sueriority Problem: Automatic Algorithm/Model Class Selection*) combines decision trees, linear discriminant functions and instance-based classifiers, while Cardie (*Using Decision Trees to Improve Case-Based Learning*) uses decision trees to select attributes to be used for case-based learning.

The Conference was followed by three workshops: *Fielded Applications of Machine Learning*, *Reinforcement Learning* and *Knowledge Compilation and Speedup Learning*. By far the most attended was the workshop on reinforcement learning. I attended the applications workshop, where several interesting machine learning applications that are in daily use were presented.

The SKICAT system for cataloging sky objects was especially impressive: it was presented by Fayyad et al. at the Conference and specific implementation and application details were presented at the workshop. Using decision tree learning techniques, very accurate (94%) rules were induced, that can now perform the task of cataloging sky objects from a large sky survey, a task that would have been by far too huge for humans to do manually. Other applications were: process delay analysis in rotogravure printing (Evans), predicting pilot bid behaviour with genetic algorithms (Adriaans), support for help desks (Allen), text retrieval (Waltz), predicting activity in the automobile market (Nakhaeizadeh) and form filling (Schlimmer). The applications workshop had no proceedings, but a book with improved versions of the presented papers will be edited by Pat Langley and Yves Kodratoff.

MACHINE LEARNING AND KNOWLEDGE ACQUISITION IJCAI-93 Workshop

Machine learning and knowledge acquisition: common issues, contrasting methods, and integrated approaches.

29 August 1993, Chambery, France.

Machine learning and knowledge acquisition share the common goal of acquiring and organizing the knowledge of a knowledge-based system. However, each field has a different focus, and most research is still done in isolation from other fields. The focus of knowledge acquisition has been to improve and partially automate the acquisition of knowledge from human experts. In contrast, machine learning focuses on mostly autonomous algorithms for acquiring or improving the organization of knowledge, often in simple prototype domains. Also, in knowledge acquisition, the acquired knowledge is directly validated by the expert who expresses it, while in machine learning, the acquired knowledge needs an experimental validation on data sets independent of those on which learning took place. As machine learning moves to more 'real' domains, and knowledge acquisition attempts to automate more of the acquisition process, the two fields increasingly find themselves investigating common issues with complementary methods. However, a lack of common research methodologies, terminology, and underlying assumptions often hinder close collaboration. The purpose of this symposium is to bring together machine- learning and knowledge-acquisition researchers in order to facilitate cross-fertilization and collaboration, and to promote integrated approaches which could take advantage of the complementary nature of machine learning and knowledge acquisition.

Topics of interest include, but are not limited to, the following: Case Studies, Comparative Studies, Hard Problems, Knowledge Representation, Key Issues, Overviews, Position Papers.

It is recommended that papers make explicit the research methodology, the underlying assumptions, definitions of technical terms, important future issues, and potential points of collaboration. They should not exceed 15 pages. The organizers intend to publish a selection of the accepted papers as a book or in a special issue of a journal. They encourage the authors to take this into account when preparing their papers. The format of the workshop will be paper sessions with discussion at the end of each session, and a concluding panel on the integrated approaches, guidelines for successful collaboration, and concrete action items. The number of the participants in the workshop is limited to 40.

Each workshop attendee must also register for the IJCAI conference and must pay an additional 300FF (about $60) fee for the workshop. One student attending the workshop and in charge of taking notes will be exempted from the additional 300 FF fee. Volunteers are invited.

WORKSHOP CO-CHAIRS

Smadar Kedar, NASA Ames & Inst. for Learning Sciences, (kedar@ils.nwu.edu)

Yves Kodratoff, CNRS & Universite de Paris-Sud, (yk@lri.lri.fr)

Gheorghe Tecuci, George Mason Univ. & Romanian Academy, (tecuci@aic.gmu.edu)

PROGRAM COMMITTEE

Ray Bareiss, Institute for the Learning Sciences
Catherine Baudin, NASA Ames
Guy Boy, European Inst. of Cognitive Sc. and Eng.
Brian Gaines, University of Calgary
Matjaz Gams, Jozef Stefan Institute
Jean-Gabriel Ganascia, Univ. Pierre and Marie Curie
Nathalie Mathe, European Space Ag. and NASA Ames
Ryszard Michalski, George Mason University
Raymond Mooney, University of Texas at Austin
Katharina Morik, Dortmund University
Mark Musen, Stanford University
Michael Pazzani, Univ. of California at Irvine
Luc De Raedt, Catholic University of Leuven
Alan Schultz, Naval Research Laboratory
Mildred Shaw, University of Calgary
Maarten van Someren, University of Amsterdam
Walter Van de Velde, University of Brussels

ADDRESS FOR CORRESPONDENCE

Gheorghe Tecuci
Artificial Intelligence Center, Computer Science Department
George Mason University, 4400 University Dr., Fairfax, VA 22030
email: mlka93@aic.gmu.edu, fax: (703)993-3729

Those who would like to attend without a presentation should send a one to two-page description of relevant research interests and a list of selected publications.

(Information about events should be sent by e-mail to matjaz.gams@ijs.si, if possible in LaTeXor at least in ASCII)

ERK'93
Electrotechnical and Computer Conference
Elektrotehniška in računalniška konferenca
27.–29. September 1993

Conference Chairman
Baldomir Zajc
University of Ljubljana
Faculty of Electr. Eng. and Comp. Science
Tržaška 25, 61000 Ljubljana, Slovenia
Tel: (061) 265 161, Fax: (061) 264 990
E-mail: baldomir.zajc@fer.uni-lj.si

Conference Vice-chairman
Bogomir Horvat
University of Maribor
Technical Faculty,
Smetanova 17, 62000 Maribor, Slovenia
Tel: (062) 25 461, Fax: (062) 212 013
E-mail: horvat@uni-mb.ac.mail.yu

Program Committee Chairman
Saša Divjak
University of Ljubljana
Faculty of Electr. Eng. and Comp. Science
Tržaška 25, 61000 Ljubljana, Slovenia
Tel: (061) 265 161, Fax: (061) 264 990
E-mail: sasa.divjak@fer.uni-lj.si

Programe Committee
Tadej Bajd
Zmago Brezočnik
Janko Drnovšek
Matjaž Gams
Ferdo Gubina
Marko Jagodič
Jadran Lenarčič
Drago Matko
Miro Milanovič
Andrej Novak
Nikola Pavešić
Franjo Pernuš

Publications Chairman
Franc Solina
University of Ljubljana
Faculty of Electr. Eng. and Comp. Science
Tržaška 25, 61000 Ljubljana, Slovenia
Tel: (061) 265 161, Fax: (061) 264 990
E-mail: franc@fer.uni-lj.si

Advisory Board
Ludvik Gyergyek
Dali Djonlagić
Karel Jezernik
Peter Jereb
Alojz Kralj
Marjan Plaper
Jernej Virant
Lojze Vodovnik

Call for Papers

for the second **Electrotechnical and Computer Conference ERK'93**, to be held from 27–29 September 1993 in Portorož, Slovenia. All presentations during the first day of the conference (invited lectures and selected sessions) will be given in English.

The following areas will be represented at the conference:

- *electronics,*
- *telecommunications,*
- *measurement,*
- *automatic control and robotics,*
- *computer and information science,*
- *artificial intelligence and pattern recognition,*
- *biomedical engineering,*
- *power engineering.*

The conference is being organized by the **Slovenian Section of IEEE** and other Slovenian professional societies:

- Slovenian Society for Automatic Control,
- Slovenian Measurement Society (ISEMEC 93),
- SLOKO–CIGRE,
- Slovenian Society for Medical and Biological Engineering,
- Slovenian Society for Robotics,
- Slovenian Artificial Intelligence Society,
- Slovenian Pattern Recognition Society.

Authors who wish to present a paper at the conference should send three copies of their abstract (500 words) to the chairman of the Programe Committee, Prof. S. Divjak. The abstract should include:

1. the title of the paper,
2. author's address,
3. telephone, fax and e-mail of the contact author,
4. the paper's subject area.

Authors of accepted papers will have to prepare a four-page camera-ready copy of their paper for inclusion in the proceedings of the conference.

Time schedule:	Abstracts due	*1 June 1993*
	Notification of acceptance	*30 June 1993*
	Camera-ready paper	*1 September 1993.*

For all additional information, please contact the conference chairmen.

AI'93 WORKSHOP ON EVOLUTIONARY COMPUTATION

Call for Papers and Participation

Melbourne, Australia, 16 November 1993

Scope and Format

The AI'93 Workshop on Evolutionary Computation will be held as part of AI'93 (The Sixth Australian Joint Conference on Artificial Intelligence, Melbourne, Australia, 17-19 November 1993). People from all areas of evolutionary computation are encouraged to participate in and submit their papers to the Workshop.

The workshop consists of a limited number of formal presentations and many informal discussions. It will also provide a forum for the exchange of information on current research among workers in the field of evolutionary computation. The first part of the workshop is the formal presentation. The second part will focus on certain specific topics, suggested by the participants. All participants are invited to bring their ideas and views into the discussion.

Topics of this workshop include, but are not limited to: *Optimisation; Evolutionary Artificial Neural Networks; Classifier Systems and Other Evolutionary Learning Systems; Self-Organisation; Collective Behaviour; Complexity in Evolutionary Systems; Analyses and Comparisons of Different Algorithms; Parallel Implementation; Applications.*

Participation and Submission

All participants should submit a description of their research interests by **9 August 1993**. Late application is acceptable only if there are vacant places. Participants will be notified by **10 September 1993**.

People who wish to present a paper at the Workshop should submit **three** hard copies of a 500–1000 word extended abstract (one or two pages) by **9 August 1993** to the workshop Chair. Electronic submissions by email are also acceptable. Notification of acceptance will be send out by **10 September 1993**. The full papers should be submitted by **16 November 1993**, i.e., the day of the Workshop.

Send all submissions/correspondence to:

Dr X. Yao (Workshop Chair)
Dept of Computer Science, University College,
University of New South Wales
Australian Defence Force Academy,
Canberra, ACT 2600, Australia
Email: xin@csadfa.cs.adfa.oz.au.
Tel: +61 6 2688819.
Fax: +61 6 2688581

Publication

Preprints of all accepted extended abstracts will be made available at the Workshop. Authors of selected papers presented at the Workshop will be invited to resubmit their papers for publication in a special issue of the journal **Informatica**.

Registration

$30 workshop registration fee should be paid to the AI'93 Conference Secretariat (Mures Convention Management).

Organising Committee

D. Abramson (Griffith Univ), E. Lewis (Univ College, UNSW), B. Marksjö (CSIRO DBCE), H.B. Penfold (Univ of Newcastle), X. Yao (Univ College, UNSW).

Important Dates

9 August 1993	Extended abstracts and research interest descriptions
10 September 1993	Notification of acc./rejection
16 November 1993	Full papers and the Workshop (half a day).

Symposium: CYBERNETIC PRINCIPLES OF KNOWLEDGE DEVELOPMENT

as part of the

12th European Meeting on Cybernetics and Systems Research EMCSR'94, Vienna, April 5-8, 1994

About the Symposium:

A symposium in collaboration with the Principia Cybernetica Project (PCP) will be held at EMCSR'94. The joint chairmen are F. Heylighen (representing PCP) and S. Umpleby. The theme is a cybernetic perspective on the creation and evolution of knowledge, with special emphasis on methods of model construction in science. We wish to focus on both fundamental principles (what is knowledge, what is science, which criteria distinguish adequate knowledge, how does knowledge originate and develop, what is the role of induction, abduction, blind variation, selection, recombination, memetic spreading...) and practical applications (which methods and tools can help us to steer or improve the generation of knowledge). The latter is especially important for the Principia Cybernetica Project, as a collaborative computer-supported attempt to develop philosophical knowledge.

The EMCSR meetings are possibly the most important and best-organized large congresses in their domain. Though they are traditionally called "European", they in fact bring together researchers from all continents, albeit with a relatively large proportion of people from Central and Eastern Europe. Among the distinctive features are the high-quality, well-distributed Proceedings, which are available at the start of the Conference. This implies that papers should be submitted (to the Congress secretariat, not to the chairpersons!) well in advance of the start of the conference. The official CFP and preliminary programme of EMCSR'94 are appended below.

After the succesful organization of a symposium at the 8th World Congress of Systems and Cybernetics (New York, 1990), of the 1st Workshop of the Principia Cybernetica Project (Brussels, 1991), and of a Symposium at the 13th Int. Congress on Cybernetics (Namur, 1992), this will be the fourth official event of the Principia Cybernetica Project.

For more information about the Symposium (not for paper submissions), contact:

Dr. Francis Heylighen
PO-PESP, Free University of Brussels, Pleinlaan 2,
B-1050 Brussels, Belgium.
Fax: +32-2-641 24 89.
E-mail: fheyligh@vnet3.vub.ac.be.

Prof. Stuart Umpleby
School of Business and Public Management,
George Washington University, Washington DC 20052.
Fax: +1-202-994 4930.
E-mail: umpleby@gwuvm.bitnet.

Organizers:

Austrian Society for Cybernetic Studies
in co-operation with:
University of Vienna, Department of Medical Cybernetics and Artificial Intelligence, and:
International Federation for Systems Research,

Chairman: Robert Trappl, President of the Austrian Society for Cybernetic Studies

About the Congress:

The international support of the European Meetings on Cybernetics and Systems Research held in Austria in 1972, 1974, 1976, 1978, 1980, 1982, 1984, 1986, 1988, 1990 and 1992 (when 300 scientists from more than 30 countries met to present, hear and discuss 210 papers) encouraged the Council of the Austrian Society for Cybernetic Studies (OSGK) to organize a similar meeting in 1994 to keep pace with the continued rapid developments in related fields.

A number of Symposia will be arranged, and we are grateful to colleagues who have undertaken the task of organizing these events. As on the earlier occasions, eminent speakers of international repute will present the latest research results at daily plenary sessions.

Symposia:

A. General Systems Methodology
G.J.Klir, USA

B. Advances in Mathematical Systems Theory
M.Peschel, Germany & F.Pichler, Austria

C. Fuzzy Sets, Approximate Reasoning & Knowledge-Based Systems
C.Carlsson, Finland, K-P.Adlassnig, Austria & E.P.Klement, Austria

D. Designing and Systems, and Their Education
B.Banathy, USA, W.Gasparski, Poland & G.Goldschmidt, Israel

E. Humanity, Architecture and Conceptualization
G.Pask, UK, & G.de Zeeuw, Netherlands

F. Biocybernetics and Mathematical Biology
L.M.Ricciardi, Italy

G. Systems and Ecology
F.J.Radermacher, Germany & K.Freda, Austria

H. Cybernetics and Informatics in Medicine
G.Gell, Austria & G.Porenta, Austria

I. Cybernetics of Socio-Economic Systems
K.Balkus, USA & O.Ladanyi, Austria

J. Systems, Management and Organization
G.Broekstra, Netherlands & R.Hough, USA

K. Cybernetics of National Development
P.Ballonoff, USA, T.Koizumi, USA & S.A.Umpleby, USA

L. Communication and Computers
A M.Tjoa, Austria

M. Intelligent Autonomous Systems
J.W.Rozenblit, USA & H.Praehofer, Austria

N. Cybernetic Principles of Knowledge Development
F.Heylighen, Belgium & S.A.Umpleby, USA

O. Cybernetics, Systems & Psychotherapy
M.Okuyama, Japan & H.Koizumi, USA

P. Artificial Neural Networks and Adaptive Systems
S.Grossberg, USA & G.Dorffner, Austria

Q. Artificial Intelligence and Cognitive Science
V.Marik, the Czech Republic & R.Born, Austria

R. Artificial Intelligence & Systems Science for Peace Research
S.Unseld, Switzerland & R.Trappl, Austria

Submission of papers:

Acceptance of contributions will be determined on the basis of Draft Final Papers. These Papers must not exceed 7 single-spaced A4 pages (maximum 50 lines, final size will be 8.5 x 6 inch), in English. They must contain the final text to be submitted, including graphs and pictures. However, these need not be of reproducible quality.

The Draft Final Paper must carry the title, author(s) name(s), and affiliation in this order. Please specify the symposium in which you would like to present your paper. Each scientist shall submit only one paper. Please send three copies of the Draft Final Paper to the Conference Secretariat (not to symposia chairpersons!).

DEADLINE FOR SUBMISSION: October 8, 1993. In order to enable careful refereeing, Draft Final Papers received after the deadline cannot be considered.

FINAL PAPERS: Authors will be notified about acceptance no later than November 13, 1993. They will be provided at the same time by the conference secretariat with detailed instructions for the preparation of the final paper.

For further information about the Congress, contact:

EMCSR 94 – Secretariat:
Oesterreichische Studiengesellschaft fuer Kybernetik
A-1010 Wien 1, Schottengasse 3, Austria.
Phone: +43-1-53532810
Fax: +43-1-5320652
E-mail: sec@ai.univie.ac.at

KR'94 - FOURTH INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING

Gustav Stresemann Institut, Bonn, Germany
May 24-27, 1994

The KR conferences emphasize the theoretical principles of knowledge representation and reasoning, the relationships between these principles and their embodiment in working systems, and the relationships between these approaches to problems and corresponding approaches in other areas of AI and in other fields. In 1994, the conference will be held in Europe for the first time.

Submissions are encouraged in (but not limited to) topics concerning representational formalisms, reasoning methods and tasks, generic ontologies, and issues in implemented KR&R systems. Submission deadline: **November 8, 1993.**

Program co-chairs:

Jon Doyle
Email: doyle@lcs.mit.edu,
Phone: +1 (617) 253-3512
Laboratory for Computer Science
545 Technology Square, Cambridge,
MA 02139, USA

Piero Torasso
Email: torasso@di.unito.it,
Phone: +39 11 7712002
Universita' di Torino, Dipartimento di Informatica
Corso Svizzera 185, I-10149 Torino, ITALY

Info: KR94-cfp-request@medg.lcs.mit.edu

INFORMATION SYSTEMS DEVELOPMENT – ISD'94
First Announcement and Call for Papers

Bled, 20-22 September 1994
University of Maribor,
School of Organizational Sciences
&
University of Gdansk
Department of Information Systems

International Program Committee

Gary Allen (UK), David Avison (UK), Christopher Barry (Ireland), Niels Bjorn-Andersen (Denmark), Cornelia Boldyreff (UK), Saša Dekleva (USA), Oscar Diaz (Spain), E.N. El-Sayed (Egypt), Eckhard Falkenberg (The Netherlands), Victor Gladun (Ukraine), Edwin Gray (UK), Igor Hawryszkiewycz (Australia), Juhani Iivari (Finland), Janete Ilmete (Latvia), Jerzy Kisielnicki (Poland), Rumjana Kirkova (Bulgaria), Marjan Krisper (Slovenia), Winfried Lamersdorf (Germany), Roland Langerhorst (The Netherlands), Leszek Maciaszek (Australia), Ir. Rik Maes (The Netherlands), Heinrich Mayr (Austria), Anders Nilsson (Sweden), T.William Olle Ltd., Eugene Ovsyannikov (Russia), Ivan Rozman (Slovenia), Bernd Schiemenz (Germany), Stefano Spaccapietra (Switzerland), Roland Stamper (The Netherlands), Frank Stowell (UK), Bo Sundgren (Sweden), Veljko Topolovec (Croatia), Roland Traunmueller (Austria), Tadeusz Wierzbicki (Poland)

Co-Chairmen

Jože Zupančič, University of Maribor (Slovenia)
Stanislaw Wrycza, University of Gdansk (Poland)

Invitation

This Conference gives an opportunity for participants to express ideas on the current state of the art in ISDM, and to discuss and exchange views about new methods, tools and applications. An objective of the conference is not only to share scientific knowledge and interests but to establish strong ties among the participants. We seek your active participation by presenting a paper and/or by your involvement in a discussion session.

The program includes paper and discussion sessions. The day just before the conference includes tutorials given by recognised IS scientists. Social program and sightseeing tours will also be organised.

Conference Topics

The conference committee seeks original papers based on research and/or practice. Suggested topics include, but are not limited to: 1. Theoretical foundations, new paradigms and trends of IS development 2. Modelling IS development process: models and meta models, modelling techniques and tools 3. Methods, techniques and tools of system development, CASE tools 4. Object Orientation in IS development 5. User interfaces design 6. Computer supported co-operative work (CSCW), hypermedia in IS development 7. Empirical studies, case studies, evaluation of existing methods 8. IS project management, quality assurance, risk and quality evaluation 9. Information system strategies, information planning 10. Education and training of IS personnel and users 11. Human, social and organizational dimension of IS development 12. Reconciliation of human and technical factors of IS development 13. IS re-engineering, IS support and maintenance 14. Implementation issues of specific application domains (e.g.: DSS and EIS, expert systems, safety/life critical systems, strategic IS, distributed/federated systems)

Publications

All papers accepted by the Programme Committee will be published in full in the Conference Proceedings which will be distributed at the Conference. A limited number of selected papers will, with approval of the authors, be published (in English) in a selected professional journal. The language of the Conference is English.

Submission Procedure and Time Table

Authors are requested to indicate their intention to submit a paper by completing and mailing the attached form. You will receive a copy of instructions concerning the standard format required for the preparation of papers. The format is optional for the initial submission but will be required for papers accepted for inclusion in the Proceedings and presentation at the Conference. All submissions will be reviewed by at least two referees. Submitted papers should include a separate title page with each author's full name, complete address, and if available, telephone, fax number and e-mail address.

Due Dates:

Initial submission (4 copies):	March 20, 1994
Notification of acceptance	June 20, 1994
Camera ready papers	August 10, 1994

MAIL: all mail should be addressed to:
Jože Zupančič (ISD'94)
School for Organizational Sciences
University of Maribor
Prešernova 11, 64000 Kranj, Slovenia
Tel. +38 64 222 804, Fax +38 64 221 424
E-mail ISD@FOV.UNI-MB.SI

THE MINISTRY OF SCIENCE AND TECHNOLOGY OF THE REPUBLIC OF SLOVENIA

The Ministry of Science and Technology also includes the Standards and Metrology Institute of the Republic of Slovenia, and the Industrial Property Protection Office of the Republic of Slovenia.

Scientific Research and Development Potential

The statistical data for 1991 showed that there were 230 research and development institutions, organizations or organizational units in Slovenia, of which 73 were independent, 32 were at the universities, and 23 at medical institutions. The remainder were for the most part departments in industry. Altogether, they employed 13,000 people, of whom 5500 were researchers and 4900 expert or technical staff.

In the past 10 years, the number of researchers has almost doubled: the number of Ph.D. graduates increased from 1100 to 1484, while the number of M.Sc.'s rose from 650 to 1121. The 'Young Researchers' (i.e. postgraduate students) programme has greatly helped towards revitalizing research. The average age of researchers has been brought down to 40, with one-fifth of them being younger than 29.

The table below shows the distribution of researchers according to educational level and fields of research:

	Ph.D.	M.Sc.
Natural Sciences	315	217
Engineering-Technology	308	406
Medical Sciences	262	174
Agricultural Sciences	122	69
Social Sciences	278	187
Humanities	199	68
Total	1484	1121

Financing Research and Development

Statistical estimates indicate that US$ 260 million (1.7% of GNP) was spent on research and development in Slovenia in 1991. Half of this comes from public expenditure, mainly the state budget. In the last three years, R&D expenditure by business organizations has stagnated, a result of the current economic crisis. This crisis has led to the financial decline and increased insolvency of firms and companies. These cannot be replaced by the growing number of mainly small businesses. The shortfall was addressed by increased public-sector R&D spending: its share of GNP doubled from the mid-seventies to 0.86% in 1993.

Overall, public funds available for Research & Development are distributed in the following proportions: basic research (35%), applied research (20%), R&D infrastructure (facilities) (20%) and education (25%).

Research Planning

The Science and Technology Council of the Republic of Slovenia, considering initiatives and suggestions from researchers, research organizations, professional associations and government organizations, is preparing the draft of a national research program (NRP). This includes priority topics for the national research policy in basic and applied research, education of expert staff and equipping institutions with research facilities. The NRP also defines the mechanisms for accelerating scientific, technological and similar development in Slovenia. The government will harmonize the NRP with its general development policy, and submit it first to the parliamentary Committee for Science, Technology and Development and after that to parliament as a whole. Parliament approves the NRP each year, thus setting the basis for deciding the level of public support for R&D.

The Ministry of Science and Technology provides organizational support for the NRP, but it is mainly a government institution responsible for controlling expenditure of the R&D budget, in compliance with the NRP and the criteria provided by the Law on Research Activities: International quality standards of groups and projects, relevance to social development, economic efficiency and rationality of the project. The Ministry finances research or co-finances development projects through public bidding and partly finances infrastructure research institutions (national institutes), while it directly finances management and top-level science.

The focal points of R&D policy in Slovenia are:
- maintaining the high level and quality of research activities,
- stimulating cooperation between research and industrial institutions,
- (co)financing and tax assistance for companies engaged in technical development and other applied research projects,
- research training and professional development of leading experts,
- close involvement in international research and development projects,
- establishing and operating facilities for the transfer of technology and experience.

In evaluating the programs and projects, and in deciding on financing, the Ministry works closely with expert organizations and Slovene and foreign experts. In doing this, it takes into consideration mainly the opinions of the research leaders and of expert councils consisting of national research coordinators and recognized experts.

The Ministry of Science and Technology of the Republic of Slovenia. Address: Slovenska c. 50, 61000 Ljubljana. Tel. +38 61 111 107, Fax +38 61 124 140.

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 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 800 staff, has 500 researchers, about 250 of whom are postgraduates, over 200 of whom have doctorates (Ph.D.), and around 150 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of Slovenia (or S♡nia). 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.:+38 61 159 199, Fax.:+38 61 161 029
Tlx.:31 296 JOSTIN SI
E-mail: matjaz.gams@ijs.si
Contact person for the Park: Iztok Lesjak, M.Sc.
Public relations: Ines Černe

INFORMATION FOR CONTRIBUTORS

1 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 the original figures on separate sheets and the text on an IBM PC DOS floppy disk or by e-mail – both in ASCII and the Informatica LaTeX format. Style (attached) and examples of papers can be obtained by e-mail from the Contact Person.

2 News, letters, opinions

Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the Contact Person.

```
\documentstyle[twoside,informat]{article}

\begin{document}
  \title{TITLE OF THE ARTICLE}
   \author{First Author \\ Address \\ E-mail \\
      AND \\
      Second Author \\ Address \\ E-mail}
  \titleodd{TITLE IN HEADER}
  \authoreven{Author's name in header}
  \keywords{article keywords}
  \edited{editor in charge}
  \received{date}
  \revised{date}
  \accepted{date}
  \abstract{abstract -- around 200 words}
  \maketitle

  \section{Introduction}
    Text of Introduction.

  \section{Subject}
    Text of the Subject.

  \begin{figure}
    An example of a figure over 1 column.
    \caption{Caption of the figure 1}
  \end{figure*}

  \begin{figure*}
    An example of a figure over 2 columns.
    \caption{Caption of the figure 2}
  \end{figure*}

  \begin{thebibliography}{99}
    \bibitem{} First Author: {\sl Title},
      Magazine, Vol. 1, No. 1.
  \end{thebibliography}

\end{document}
```

```
\def\journal{Informatica {\bf 17} page xxx--yyy}
\immediate\write16{'Informatica' V1.2  by B."Z}
\newif\iftitle \titlefalse
\hoffset=11mm \voffset=8mm
\oddsidemargin=-21mm \evensidemargin=-14mm
\topmargin=-33mm
\headheight=17mm \headsep=10mm
\footheight=8.4mm \footskip=52.5mm
\textheight=242mm \textwidth=170mm
\columnsep=5mm \columnseprule=0pt
\twocolumn \sloppy \flushbottom
\parindent 1em
\leftmargini 2em \leftmargin\leftmargini
\leftmarginv .5em \leftmarginvi .5em
\def\labelitemi{\bf --} \def\labelitemii{--}
\def\labelitemiii{--}
\setcounter{secnumdepth}{3}
\def\maketitle{\twocolumn[%
  \vbox{\hsize=\textwidth\Large\bf\raggedright
  \uppercase{\@title}}\vss\bigskip\bigskip \vbox{
  \hsize=\textwidth \@author}\bigskip\smallskip
  \vbox{\hsize=\textwidth {\bf Keywords:}
    \@keywords}
  \bigskip \hbox{{\bf Edited by:} \@edited}
  \smallskip
  \hbox{{\bf Received:}
    \hbox to 10em{ \@received\hss}
    {\bf Revised:}\hbox to 10em{ \@revised\hss}
    {\bf Accepted:}\hbox to 10em{ \@accepted\hss}}
  \bigskip \vbox{\hsize=\textwidth
    \leftskip=3em \rightskip=3em \sl \@abstract}
  \bigskip\bigskip]\titletrue}
\def\maketitleauthor{\twocolumn[%
  \vbox{\hsize=\textwidth \Large\bf\raggedright
  \@title}\vss \bigskip\bigskip
  \vbox{\hsize=\textwidth \@author}
  \bigskip\bigskip] \gdef\@title{}\titletrue}
\def\makeonlytitle{\twocolumn[%
  \vbox{\hsize=\textwidth \Large\bf\raggedright
  \@title}\vss\bigskip\bigskip]
  \gdef\@title{}\titletrue}
\def\@title{} \def\@author{}
\def\@titleH{} \def\@authorH{}
\def\@keywords{} \def\@edited{} \def\@abstract{}
\def\@received{} \def\@revised{} \def\@accepted{}
\def\authoreven#1{\gdef\@authorH{#1}}
\def\titleodd#1{\gdef\@titleH{\uppercase{#1}}}
\def\keywords#1{\gdef\@keywords{#1}}
\def\edited#1{\gdef\@edited{#1}}
\def\received#1{\gdef\@received{#1}}
\def\revised#1{\gdef\@revised{#1}}
\def\accepted#1{\gdef\@accepted{#1}}
\long\def\abstract#1{\gdef\@abstract{#1}}
\def\section{\@startsection {section}
  {1}{\z@}{-3.5ex plus -1ex minus -.2ex}
  {2.3ex plus .2ex}{\Large\bf\raggedright}}
\def\subsection{\@startsection{subsection}
  {2}{\z@}{-3.25ex plus -1ex minus -.2ex}
  {1.5ex plus .2ex}{\large\bf\raggedright}}
\def\subsubsection{\@startsection{subsubsection}
  {3}{\z@}{-3.25ex plus -1ex minus -.2ex}
  {1.5ex plus .2ex}{\normalsize\bf\raggedright}}
\def\@evenhead{\hbox to 3em{\bf\thepage\hss}
  {\small\journal\hfil \iftitle\else
  \@authorH \fi}\global\titlefalse}
\def\@oddhead{{\small\iftitle\else \@titleH \fi
  \hfil\journal}\hbox to 3em{\hss\bf\thepage}
  \global\titlefalse}
\def\@evenfoot{\hfil} \def\@oddfoot{\hfil}
\endinput
```

REVIEW REPORT

Basic Instructions

Informatica publishes scientific papers accepted by at least two referees outside the author's country. Each author should 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. 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. The names of the referees should not be revealed to the authors under any circumstances. The names of referees will appear in the Refereeing Board. Each paper bears the name of the editor who appointed the referees.

It is highly recommended that each referee writes **as many remarks as possible directly on the manuscript**, ranging from typing errors to global philosophical disagreements. The chosen editor will send the author copies with remarks, and if accepted also to the Contact Person with the accompanying completed Review Reports. The Executive Board will inform the author that the paper is accepted, meaning that it will be published in less than one year after receiving original figures on separate sheets and the text on an IBM PC DOS floppy disk or through e-mail – both in ASCII and the Informatica LaTeX format. Style and examples of papers can be obtained through e-mail from the Contact Person.

Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the Contact Person.

Date Sent:

Date to be Returned:

Name and Country of Referee:

Name of Editor:

Title:

Authors:

Additional Remarks:

All boxes should be filled with numbers 1-10 with 10 as the highest rated.

The final mark (recommendation) consists of two orthogonal assessments: scientific quality and readability. The readability mark is based on the estimated perception of average reader with faculty education in computer science and informatics. It consists of four subfields, representing if the article is interesting for large audience (interesting), if its scope and approach is enough general (generality), and presentation and language. Therefore, very specific articles with high scientific quality should have approximately similar recommendation as general articles about scientific and educational viewpoints related to computer science and informatics.

☐ SCIENTIFIC QUALITY

☐ Originality

☐ Significance

☐ Relevance

☐ Soundness

☐ Presentation

☐ READABILITY

☐ Interesting

☐ Generality

☐ Presentation

☐ Language

☐ FINAL RECOMMENDATION

☐ Highly recommended

☐ Accept without changes

☐ Accept with minor changes

☐ Accept with major changes

☐ Author should prepare a major revision

☐ Reject

INFORMATICA

AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

INVITATION, COOPERATION

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 17th year, it is becoming 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 (institutional rate 50 DM, individual rate 25 DM, and for students 10 DM). Send a check for the appropriate amount to Slovene Society Informatica, Vožarski pot 12, Ljubljana, account no. 50101-678-51841.

Please, return this questionnaire.

QUESTIONNAIRE

☐ Send Informatica free of charge

☐ Yes, we subscribe

☐ We intend to cooperate (describe):

☐ Proposals for improvements (describe):

Informatica is published in one volume (4 issues) per year.

If possible, send one copy of Informatica to the local library.

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: .....

Title and Profession (optional): .....

.....

Home Address and Telephone (optional): .....

.....

Office Address and Telephone (optional): .....

.....

E-mail Address (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 Refereeing Board. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or Board of Referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board and Board of Referees 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.

Volume 17 Number 2 August 1993 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Contents:

Knowledge – The New Informational Paradigm	Editorial	107
Profiles: Jiří Šlechta		108
On a Quantum–Statistical Theory of the Pair Interaction Between the Memory Traces in the Brain	J. Šlechta	109
Integrative Domain Analysis Via Multiple Perceptions	W. Rossak T. Zemel	117
A Prolog-Based Representation for Integrating Knowledge and Data	X. Wu	135
Walking Viability and Gait Synthesis for a Novel Class of Dynamically–Simple Bipeds	J. Kieffer R. Bale	145
Modelling Biodegradation by an Example Based Learning System	D. Gamberger S. Sekušak A. Sabljić	157
Successive Naive Bayesian Classifier	I. Kononenko	167
Moral Hazard Problem Solving by Means of Preference Ranking Methods	I. Saražin Lovrečič J. Grad	175
Fifth Generation Computer Systems (FGCS) Project in Japan	K. Furukawa	183
Reports and Announcements		200