Developing Mindful Machines Requires More Than Information

Darren J.R. Whobrey
Department of Computer Science
City University, London
darren@blueberry.netkonect.co.uk

Abstract

There is a growing need, particularly in the business community, for computers (digital machines) that use knowledge intelligently, implying that system developers need to start thinking more about semantic content in addition to syntactically manipulating information. In this regard, much promise has been shown by theories of content based on the dispositional account of meaning stemming from Charles Peirce's work on semiotics and championed by James Fetzer. Consequently, this paper considers the theoretical foundations of information processing from a cognitive perspective that takes the most fundamental ingredients to be the dispositional, and therefore causal, structure of the system.

Keywords: artificial intelligence, causal systems, dispositions, interpretation, representation, minds, semiotics, signification.

1. Introduction

Fetzer [6] suggests minds can be defined as sign using systems in the sense of Peirce's semiotic (theory of signs). For Peirce, the semiotic was just one part of his philosophical architectonic in which the nature and role of mind was central. Herein, Peirce's ideas are treated in a broader sense as a basis for exploring the overall nomic structure of mind. Thus, while minds can use information, Peirce suggested they might operate in terms of semiotic processes in a dispositional manner that has a particular semiotic triadic form – see Umberto Eco [3]. Briefly, a sign (S) is an aspect of an occurrent process and acts as a representation of something (x) for somebody or something (z). This involves the setting up of a triadic relation within the system as simplified in Figure 1 – further details may be found in Whobrey [17].

This leads to at least two kinds of structures in use by the system when signifying something. Firstly, in terms of explicit representational relational structures that are interpreted, and secondly, in terms of implicit relational structures occurring between the interaction of processes in the system. The explicit form of representation leads to structures that can be mapped to an equivalent Ramsey sentence preserving the information content with respect to some interpreter, whereas this is not applicable to implicit structures. For example, in computer science a distinction is made between declarative and procedural knowledge and programming languages. Theories of content based solely on the explicit form have yet to overcome the interpreter regress problem – see Barbara von Eckardt [2]. However, for the implicit form, the prospects look more promising: modelling semiotic processes by way of implicit relational structures is not a computational process in that it is not about computing a result but rather the occurrent form of the causal interactions – see Fetzer [8]. At this level, there are no interpreters to operate on representations. These structures are characterised by networks of dispositions with a certain temporal profile.

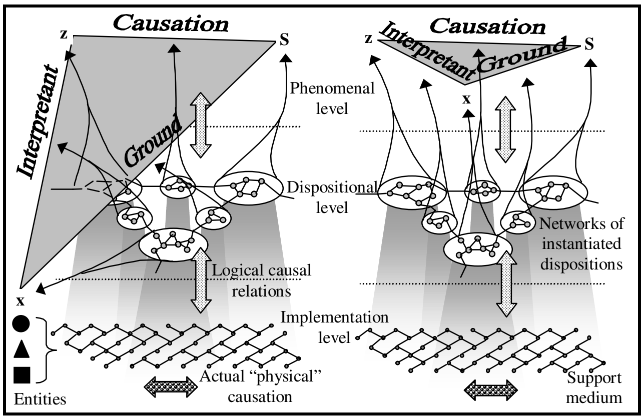

Figure 1. Ontological level interpretation of Fetzer's Human Being Relation. The dotted lines represent level boundaries. The left diagram has a physical entity as object, the right diagram has a sign (thought) as object. From Whobrey [17].

Consequently, to progress toward mindful digital machines that may one-day use knowledge intelligently (for example, such as the auditability approach proposed by Hart Will [18]), developers need to incorporate into their systems the dispositional approach to modelling semiotic processes taking into account the relative ontological levels of the system. This has been categorised under *mild* AI (see Whobrey [16]), according to which a digital machine may be capable of possessing a species of mentality, rather than the *strong* and *weak* categories originally distinguished by John Searle [15]. This paper explores how the dynamic causal structure, implied by the dispositional approach (see Fetzer [4]), might be represented as a basis for implementing semiotic processes in the goal of building mindful machines. Starting with an introductory taxonomy of types of dispositions, it proceeds to a formal presentation of the fundamental mechanism by which the form of a disposition may be coded. To put this in a computational perspective, the dispositional approach is related to functional programming. Finally, the basis for a dispositional programming language is discussed.

2. Taxonomy of Dispositions

Fetzer extended Peirce's analysis of the role played by habits in imparting meaning to signs by showing how this can be cast in terms of dispositions – see Fetzer [6, p. 78] and [7]. This stemmed from Fetzer's dispositional ontology for the physical world in which the concept of a disposition was formulated with respect to a descriptive language as:

> "A predicate is *dispositional* if and only if the property it designates (a) is a tendency (of universal or statistical strength) to bring about specific outcome responses when subject to appropriate singular tests, where that property (b) is an actual physical state of some individual object or of an arrangement of objects (should it happen to be instantiated by anything at all)." – see Fetzer [4, p. 401].

As a forerunner to instilling semiotic processes into a digital machine, the goal of this section is to lay a foundation for characterising the dynamics of the system in terms of the dispositional nature of the interaction of its emergent properties. Here, a conception of emergence is adopted in which an emergent property arises from an arrangement of conspecifics with respect to the systems ontological levels, i.e. a domain of emergent properties that are all instantiations of arrangements of properties of lesser complexity – see Whobrey [16].

2.1. Graded Dispositions

The mind is a system that is continuously evolving (when viewed at a certain resolution and from a certain perspective, cf. Fetzer [4, p. 415], also see van Gelder and Port [9]), which suggests the need for ways of describing the continuous operation of a causal system that allow for incremental and continuous changes. Graded dispositions refer to properties that may have a variable state, such as the disposition to laughter. For example, when someone hears a joke, their laughter grows and subsides, rather than being an all or none, or random event. Treating a disposition as a law, in examining graded dispositions interest lies with the laws of transition between dispositions of a similar type.

One way to describe graded dispositions is in terms of temporal sequences of universal dispositions, such that each member of the sequence is an incremental variation of its neighbours (cf. Fetzer [5, p. 51] – incremental changes in strength of tendency). These variations correspond to the possible grades of the dispositional property a thing may possess at any one time. A graded disposition is then defined as the set of all sequences, where set membership is determined by a reference class description. The variations between sequence members need not be discrete, they could be continuous in which case the sequence becomes a continuum, i.e. a continuous flow – see Norton [95]. Hence, it is possible to have discrete or continuous graded dispositions.

2.2. Distributive & Reversible Dispositions

Distributive dispositions are introduced by way of an example based on the artificial neural nets popularised by John Hopfield [10]. These nets are capable of memorising patterns, or prototypes, that can be later re-invoked when prompted by a similar pattern. Bart Kosko [11] considered pairs of Hopfield nets coupled such that the pattern on one net would invoke a certain pattern on the other – see also Dreyfus et al. [1]. Relations between patterns can be programmed into these nets. So, if patterns "A" and "B" are programmed into the respective nets, such that pattern A invokes pattern B, we can say A has a tendency to cause B.

Hopfield was interested in a nets ability to evolve toward one of its prototype memories when prompted by a similar pattern. In contrast, here interest lies in the causal interactions that have taken place and their pattern. Operationally, one way to examine this is to look upon the dynamical evolution of the state of these nets in causal terms. Consider the change in the state of a net s(t) at times t_1 and t_2:

$$\text{Overall state change: } \Delta s = s(t_2) - s(t_1) \approx \delta s \text{ for small } \delta t.$$

Just as there may be some δs that has a tendency to drive the nets toward one prototype state, there may be some other δs', or inverse, that has a tendency to drive the nets away from a prototype state: $\delta s' = -\delta s$. The possibility of an inverse emphasises that the dispositional tendency exhibited by the net may be cyclic, i.e. in some sense *reversible*. For example, a piece of clay can be moulded into a ball, then a slab, then a ball again, while for blocks of stone this would be an irreversible process. Consequently, the δs corresponds to a directional influence, or tendency toward a prototype state:

$$\text{Equivalently, new state: } s(t+\delta t) = s(t) + \delta s.$$

From this, the current state of a net can be thought of as consisting of 1) a position in state space, plus 2) a directional influence (the vector δs) toward some other prototype state. Thus, *a disposition when defined irrespective of the thing, would be the tendency to produce a change of a particular form under suitable conditions.*

At the start of this section a property was defined as emerging from an arrangement of its conspecifics. An instance of a property was effectively a function of a particular configuration of elements – defined formally in Whobrey [16], see definition D-SLP. Consequently, in the above neural net example, by abstracting from the operational medium, a certain category of *distributive* dispositions can be defined as a directional and potentially reversible tendency to redistribute the domain elements of some causally connected thing. A particular kind of distributive disposition would then be defined by a reference class description that specified over what elements the disposition exerted an influence, how this set may change, and the form of its influence. A further restriction may be applied such that the redistribution is always upon a particular subset of the domain elements.

2.3. Isogenetic & Dissociated Dispositions

In the previous subsection, a category of distributive dispositions was introduced as a tendency to redistribute the domain elements of some causally connected thing. Within this category, a class of *isogenetic* distributive dispositions can be singled out according to two further refinements. Firstly, the form of the distributive dispositional tendency is itself a function of the distribution of the elements upon which the disposition is instigated. In other words, the same type of process defines the tendency of the disposition as that upon which it acts. Secondly, by abstracting from the particular kind of underlying elements, i.e. the medium, only the causal power and form of the distributive influence becomes of importance in understanding the operation of the system. This allows the analysis to focus on those kind of arrangements of properties that manifest semiotic abilities as among their emergent properties. Here, "emergent" is meant in the sense that systems as instantiations of arrangements of properties of lesser complexity (or of different properties, etc.) do not manifest them.

Consequently, a homogeneous interactive network of graded isogenetic distributive instances of dispositions can be treated as an abstract level. Referring to these as *dissociated* instances of dispositions (DIDs), they will be associated with an ontological system level that supports the evolution and interaction of instances of these dispositions. It can be treated as an ontological system level because it will be characterised by laws specific to that level that determine the interaction and evolution of the instantiated dispositions. Of concern in what follows is the lawful nature of these dispositions, how their instances relate to one-another, how they relate to the other levels, and finally, how they might lead to semiotic processes. Hence, in what follows dissociated dispositions and their instances will be used to refer to dispositional processes in which details of the medium have been abstracted away along with any of its irrelevant dispositional properties.

3. Fundamental Nature of Dispositional Processes

An instance of a dissociated disposition is supported by a mechanism that can be treated as three integrated phases: *reduction*, which projects a high dimensional input pattern to a lower dimensional (group structured) space; *alteration*, a decisional stage wherein the pattern of the disposition's causal activity changes in response to the influence of the reduction phase and the current state; and *production*, which translates this evolving causal pattern into a higher dimensional output pattern – effectively extending the influence of the instantiated disposition. Remember that changes are incremental and continuous, so this is a type of second order continuous state machine – the next input is normally metrically similar to the last, which emphasises the structure and evolution of the permissible causal-flows in the state space – see Whobrey [16]. A causal-flow is a trajectory through the causal-space of the system. Each infinitesimal volume in this space corresponds to a particular pattern of causal interactions of a given strength and direction.

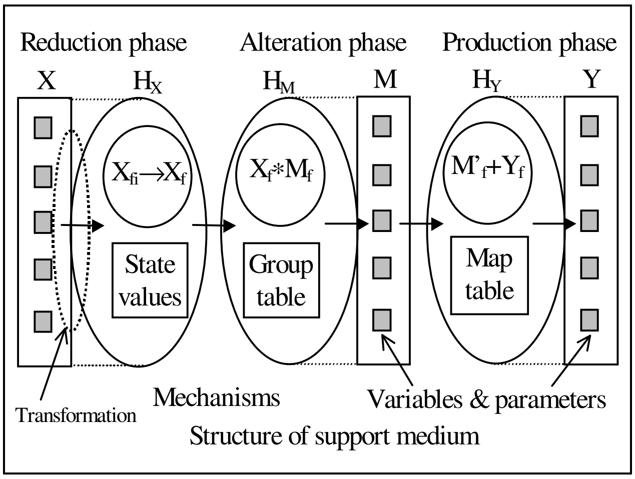

Figure 2. Stylised structure of a support medium for dissociated instances of dispositions.

At any one time the support mechanism may be supporting a number of instances of dispositions, in effect a fuzzy set (cf. quantum superposition) – this is a consequence of a continuity constraint, and a way of modelling more complex DIDs via a basis of simpler ones. Mathematically, the confluence of causal-flows that characterise a DID can be described by a Lie group element. In other words, a DID will typically be characterised by a single maximal causal-flow, called its primary mode, and a number of variations on this of decreasing metric similarity, i.e. more distant in the causal-space. Describing the DID's causal structure by a group element under a given interpretation, such as a Gaussian radial basis function, enables the interaction of DIDs to be modelled and characterised via group operations. This allows the interactions of DIDs to be dealt with in fairly high level, law like, terms: e.g. relations such as "Region H, when supporting a DID Y under the influence of DID X, will have a tendency to evolve toward being a DID Z." In simple cases, these relations can be represented in a "group table" (see below). Therefore, the causal-structure of a DID is specified by a combination of the group structure, the group element function, and the influence of the support mechanism. Consequently, the general evolution relation for a DID can be expressed as a fuzzy compound group operation:

$$X_f * M_f \xrightarrow{H_f} M'_f + Y_f \,. \qquad (1)$$

Here, X_f and Y_f represent the inputs and output influences, M_f the current state (i.e. superposition of DIDs) and H_f the influence of the support – for example, it may perform a normalising action on the DIDs.

For example, in a simple case, where the group element functions are basic logical primitives, such as AND, the causal-structure of the disposition is determined by the group table. For instance, if the input is X = AND, which is defined to mean all inputs are active, the state is M = NOT, which means it will invert inputs, and the state group table entry for these values is: AND ∗ NOT → OR, which means the state changes to OR. In addition, if there is the state group table entry: AND ∗ OR → NOT, this will then give rise to cyclic behaviour whenever the inputs are continously active. This behaviour could be identified with a disposition that *oscillates the ouput whenever all inputs are active*. In fact, the text in italics is the causal structure of the disposition and *it is* the disposition. However, notice that a table entry may act in other dispositions as well. For instance, as the input activity changes another entry becomes dominant.

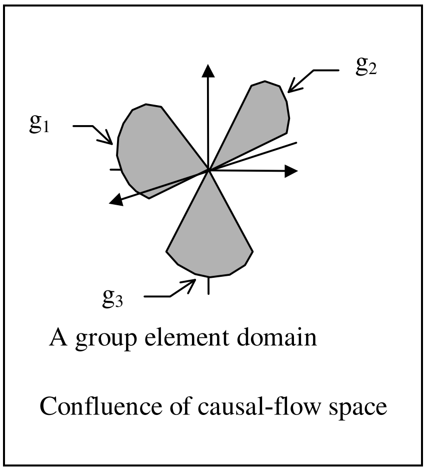

Figure 3. Example of confluence of causal-flows (cf. states) for a three dimensional real causal-flow space.

As a second example, suppose a smile is a predisposition toward the disposition of laughter, and similarly a frown is a predisposition toward anger, looking miserable a predisposition to crying, etc. The relationship between these can be represented by the simplified group table:

		M_1 Laugh	M_2 Angry	M_3 Cry	M_4 Wink
x_1	Mellow	–	–	–	+
x_2	Smile	+	–	–	–
x_3	Frown	–	+	–	–
x_4	Miserable	–	–	+	–
x_5	Squint	–	–	–	+

The plus and minus entries indicate whether a group action strengthens or weakens the current state. In addition, the table can be made square and diagonal by noticing that the cardinality of X_f and M_f are approximations to a continuum of table rows and columns, and that permuting either amounts to re-ordering the inputs or outputs, which is treated as an invariant here. This simplification enables the form of the alteration process to be described by a generic mechanism:

$$M_j(t+\delta t) =$$

$$f(\alpha M_j(t) + \beta \tfrac{1}{N} \textstyle\sum_{i=1}^{n} x_i(t) w(\tfrac{i-1}{n-1}, \tfrac{j-1}{p-1}))\,, \qquad (2)$$

$$N = \textstyle\sum_{j=1}^{p} \sum_{i=1}^{n} x_i(t) w(\tfrac{i-1}{n-1}, \tfrac{j-1}{p-1})\,, \qquad (3)$$

$$w(x,y) = e^{-\frac{1}{2}\left(\frac{x-y}{\sigma}\right)^2}\,,\ f(x) = \frac{1}{1+e^{-k(c+x)}}\,,$$

where N is a normalising factor (discussed below), c = –0.7, k = 12.0, with rate parameters $\alpha \in [0.8,1.0]$, $\beta \in [0.90,0.95]$, and $\sigma = 1/(p2\sqrt{2\ln 2})$, such that the Gaussian pulse width at half height is proportional to 1/p, where p is the cardinality of the group. This is analogous to the dynamical equation for Hopfield's net, except a weighting function is used that reflects the group structure – in this case a simple diagonal, which is also symmetrical.

Hopfield was interested in the stable states of the net and showed how its dynamics could be described by a Lyapunov function that was analogous to the potential 'dynamical' energy of the system. In contrast, here interest lies with the form of the change in the causal structure of the system after each epoch. From the above equation this can be expressed as:

$$dM(r,\theta,\phi,t) \approx$$

$$\frac{K}{N}\iiint_R x(r,\theta,\phi,t) w(r,\theta,\phi) r^2 \sin\phi \, dr d\phi \, d\theta \,,$$

$$K = \frac{\beta}{1-\alpha}\,, \qquad (4)$$

where integration accounts for the influence from a spherical region R (parameterised by r, θ and φ) on the DID under consideration (viewed as occupying an infinitesimal volume), with the simplification that it is operating in its linear range ($\frac{df(...)}{dt} \approx 1$), and that the causal substructure (i.e. the structure of the group table), the rate structure (α) and the normalising factor N, are constant. That is, N corresponds to the potential 'causal' energy of the system, it strives to keep the traditional potential 'dynamical' energy constant. The result of integrating the square of $dM(r,\theta,\phi,t)$ over some volume of interest could be treated as a measure of the 'kinetic' causal energy for the system – it increases with the degree of change in the pattern of causal influences occurring over a given period.

4. Dispositions versus Functions

In two important respects, the dispositional approach to semiotics refines its basis on functionalism, as the view that mental states are defined by the relationship between their causes and effects. Firstly, it highlights the need to account for the ontological levels within the semiotic system – for example, in order to situate the interpreter. Accordingly, a distinction can be made between the implementation level (which supports the causal structure), the dispositional level (the key property of which are DIDs), and the phenomenal level (the key property of which is the content of signs). Consequently, and secondly, mind becomes the product of an occurrent process with an extension in time and spanning at least three ontological levels. This conception of mind casts doubt on Turing machine functionalism (equating mental states with machine states) and computational theories of mind that suggest mental states are the result of a computation – for a discussion on this see Fetzer [8]. However, it does suggest that it might be possible to instil digital machines with minds provided the necessary semiotic processes and causal relations are in place for the right reasons. Accommodating semiotic processes has a deep structuring effect on the possible nomic structure of mind – see Whobrey [16].

Implementing mind in digital machines amounts to replicating the causal structure set up by the network of DIDs that underlie the semiotic processes, which define the mind, as typified in the previous equations. Trying to achieve this directly within the conventional functional programming paradigm suffers a major problem: DIDs cannot be equated with functions since they have a different operational semantics – DIDs are occurrent whereas functions are evaluated. For example, although functional programmes are functional, they still distinguish between data and code at the operational level. However, "dispositional programming" is concerned with specifying the occurrent causal structure of the system. From the functional programming world this was classified as having a "dynamic process topology" and adversely affected the degree of inherent parallelism that could be exploited when optimising performance.

Two alternatives to implementing DIDs are to either construct a special purpose machine, or to replicate the causal structure through simulation of the system equations. In the latter case a more programmatic approach would be to construct an abstract machine for DIDs along with a "dispositional" programming language. This has much in common with combinator-codes when used as machine codes for functional languages such as Miranda and Haskell, i.e. modern replacements for LISP – see Simon Peyton-Jones [13]. The main difference being that the combinators would have to be given an occurrent execution semantics. In other words, rather than following an execution model involving reductive evaluation strategies and operations applied to data, all the combinators would be active concurrently and continuously. In addition, to model the graded nature of DIDs, would require 'graded' combinators. Effectively, group operators would be replacing the combinators.

A starting point for a dispositional programming language would be a combination of Fetzer's language for universal dispositions – see Fetzer [5], Lukasiewicz's multi-valued logic – see Nicholas Rescher [14], group theory and graded-combinators. One way to situate dissociated dispositions formally is to extend the language *U*, for universal dispositions presented in Fetzer's probabilistic causal calculus, by adding parameterised dispositions and reference class definitions, and an axiom for maintaining the numerical identity of things through time – see Fetzer [5, p. 62].

The abstract language would then require an interpretation in terms of the dispositional properties of the system. Along these lines, Whobrey [16] presents a formal foundation for the system properties within a causal setting. For example, a type of system level property p_{ij}^n is defined as any process for which an operator ω_{ij}^n is definable on the level domain E_i^n. An instance of the property type is then determined by applying the operator to a specific set of elements: $p_{ijx}^n = \omega_{ij}^n \varsigma_{ij}^n (C_{ijx}^n, E_i^n)$, where the selector ς_{ij}^n picks out the specific collection of elements, and C_{ijx}^n describes their configuration. This can be parameterised with respect to time. The ongoing action of the selection operator thereby defines the numerical identity of the property through time. From this the causal structure of the system can be described in relation to the lawful constraints imposed by the lower ontological levels. In light of this, the causal properties that must be preserved in order to replicate semiotic processes (and therefore mind) on digital machines can be deduced.

5. Conclusion

To develop systems that can use knowledge intelligently, it was suggested that knowledge must be cast in terms of semiotic processes as part of the process of signification in the spirit of Peirce. This amounted to engineering causal systems, as suggested by Fetzer, via the interaction of networks of instantiated dispositions characterised by the evolution of patterns of causal interaction, and taking into account the ontological levels of the system. This is in contrast to the interpretation of explicit representational structures typically employed in current computational systems, and theories of content based on this approach, which succumb to the interpreter-regress problem. Under the dispositional approach to implementing semiotic processes, there is no such thing as an interpreter at that level of description. This highlights the alternate conceptual approach a developer must take when dealing with knowledge. Instead of focusing on the representation of knowledge and its subsequent interpretation, the developer must consider the signification of knowledge coded as an aspect of a semiotic process. This would benefit from a dispositional oriented programming paradigm, but as yet, such a paradigm has only been written about.

6. References

[1] Dreyfus, G. Guyon, I. Nadal, J.P. Personnaz, L. (1988). Storage and retrieval of complex sequences in neural networks. *Physical Review A*, Vol.38, No.12, 6365-6372.

[2] von Eckardt, B. (1993). *What is cognitive science?* Bradford Books, MIT Press.

[3] Eco, U. (1976). *A theory of semiotics*. Indiana University Press.

[4] Fetzer, J.H. (1977). A world of dispositions. *Synthese*. Vol.34, 397-421.

[5] Fetzer, J.H. (1981). *Scientific knowledge*. Dordrecht, Holland: D.Reidel.

[6] Fetzer, J.H. (1990). *Artificial intelligence: its scope and limits*. Kluwer Academic Publishers.

[7] Fetzer, J.H. (1991). Primitive concepts: habits, conventions, and laws. In: *Definitions and Definability: Philosophical Perspectives*, ed. J.H.Fetzer, D.Shatz, G.Schlesinger. 51-68.

[8] Fetzer, J.H. (1998). People are not computers: (most) thought processes are not computational procedures. *Journal of Experimental & Theoretical Artificial Intelligence.* Vol.10, 371-391.

[9] van Gelder, T. Port, R.F. (1995). It's about time: an overview of the dynamical approach to cognition. *Mind as Motion.* R.F.Port and T.van Gelder, MIT Press, 1-43.

[10] Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings National Academy Science USA, Biophysics*. Vol.79, 2554-2558.

[11] Kosko, B. (1987). Adaptive bi-directional associative memories. *Applied Optics.* Vol.26, No.23, 4947-4960.

[12] Norton, A. (1995). Dynamics: an introduction. In: *Mind as Motion*, ed. R.F.Port and T.van Gelder, MIT Press, 45-68.

[13] Peyton Jones, S.L. (1987). *The implementation of functional programming languages*. Prentice-Hall.

[14] Rescher, N. (1969). *Many-valued logic*. McGraw-Hill.

[15] Searle, J.R. (1984). *Minds, brains and science: the 1984 Reith lectures*. BBC publication.

[16] Whobrey, D.J.R. (1999). *Aspects of qualitative consciousness: a computer science perspective*. PhD Thesis. Department of Computer Science. City University, London. Online: www.soi.city.ac.uk or www.netkonect.net/~blueberry.

[17] Whobrey, D.J.R. (2000). Machine mentality and the nature of the ground relation. *Minds and Machines*. Forthcoming.

[18] Will, H.J. (2000). Auditability and controllership: extracting knowledge from accouting information. In, *Proceedings of Systems, Cybernetics & Informatics*, 2000.