The (Extensive) Implications of Evaluation on the Development of Knowledge-Based System

Tim Menzies, Paul Compton

Artificial Intelligence Laboratory,
School of Computer Science and Engineering,
University of New South Wales, PO Box 1, Kensington, NSW, Australia, 2033
{timm|compton}@cse.unsw.edu.au

Abstract: We argue that adding a requirement of evaluation and testing fundamentally changes KBS practice. In particular: (i) a fundamental change to the symbol-level representation in KBS; (ii) a rejection of certain unnecessary knowledge-level distinctions; (iii) a fundamental change to the inference engine of KBS; and (iv) a basic computational limit to the size and internal complexity of the models we create via knowledge acquisition.

1. INTRODUCTION

It would be convenient if KBS evaluation was neutral with respect to KBS practice. If an evaluation module was merely a post-hoc bolt-on, then its design could be deferred until after a system was developed. However, if evaluation adds extra requirements and restrictions to the KBS process, then the design of an evaluation module must be integrated with the system it will test.

This paper argues for the inconvenient latter position. Models constructed in *vague domains* (defined below) are possibly inaccurate. Possibly inaccurate models must be tested, lest they produce inappropriate output in some circumstances. We will develop a general abductive[1] definition for "test" in such vague domains. This characterisation leads us to two unexpected results:

- At a symbol-level, abduction executes over an and-or network with edges E and vertices V. Experimentally, we show that the limits to test are 1000 (approx) vertices ($|V|$) and average fanout ($|E|/|V|$) less than 7. The fanout limit seems fundamental to abduction while the $|V|$ limit could be increased via optimisations of our current implementation. However, given the fundamentally exponential nature of abduction, we do not expect large increases in the $|V|$ limit. Since we can't test models that are larger than $|V| = 1000$ (approx) and $|E|/|V| = 7$, we should not build models larger than these limits.
- A generalised validation module can serve as the inference module a range of knowledge-level tasks including KBS validation, prediction, model-based diagnosis, explanation, classification, learning, case-based reasoning, financial reasoning, natural language comprehension, design, and recognition. More generally, we find that abduction directly operationalises the model extraction process that Breuker and Clancey argue is at the core of expert systems inference. While we could separate inference and validation modules, we would economise on our effort by using the validation module for the inferencing. That is, our definition of "generalised test" *replaces* rather than merely *augments* the inference module of a KBS.

These results are restricted to vague domains. However, we will argue that most domains tackled by modern KBS are vague; i.e. these results are widely applicable.

This paper is structured as follows. Sections 2 and 3 defines *vague domains* and *models* respectively. Section 4 discusses what should be tested. Section 5 characterises testing in vague domains as abduction and describes our

[1] Consider a system with two facts a, b and a rule R_1: *If* $a \Rightarrow b$. *Deduction* is the inference from a to b. *Induction* is the process of learning R_1 given examples of a and b occurring together. *Abduction* is inferring a, given b (Levesque 1989). Abduction is a not a certain inference and its results must be checked by an inference assessment operator (see *BEST*, below).

experiments with the limits to test-as-abduction. Section 6 discusses the applicability of abduction to knowledge-level tasks. Section 7 discusses related work.

2. VAGUE DOMAINS

A vague domain has one or more of the following properties:

- It is *poorly measured*; i.e. known data from that domain is insufficient to confirm or deny that some inferred internal state of its model is valid.
- Its models are *hypothetical*; i.e. the domain lacks an authoritative oracle that can declare knowledge to be "right" or "wrong"). Note that in a well-measured domain, the authoritative oracle could be a database of measurements.
- Its models are *indeterminate*; i.e. the model cannot choose between a number of different possible outputs. For example, the qualitative model of Figure 1 is indeterminate. In the case of *a*↑ & *b*↑ we have two competing qualitative influences on *c*: (i) *a*↑ can cause *c*↑ while (ii) *b*↑ can cause *c*↓. Lacking quantitative information about the relative size of these competing forces, one world must be created for each possible value of *c* ; i.e. {*c*↑,*c*θ,*c*↓}[2] (Iwasaki 1989).

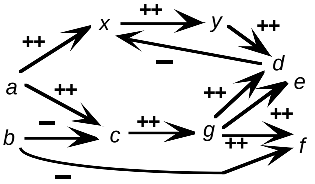

Figure 1: *Connection between entities with legal states* up , down, *or* steady. *Edges denote acceptable explanations. For example: (1)* X -- Y *iff* Y *being up or down could be explained by* X *being down or up respectively; (2)* X ++ Y *iff* Y *being up or down could be explained by* X *being up or down respectively.*

For an example of a vague domain, consider multiple expert knowledge acquisition (KA) for neuroendocrinology. Model construction/revision in neuroendocrinology (the study of the interactions of nerves and glands) is complicated by (i) the different paradigms of researchers in different countries[3]; (ii) the hypothetical nature of the domain[4]; (iii) and a lack of data[5] which means that not all portions of known models may be exercised. Consequently, correction of model anomalies may take months or even years. In the interim, researchers must use the best available current models, even if they are known to contain unresolved anomalies.

We find that, with one partial exception (see below), all the domains we have studied in detail in the last decade are vague. For example, neuroendocrinology, economics and ecology are very vague domains. The (in)famous "Limits to Growth" study attempted to predict the international effects of continued economic growth (Meadows, Meadows et al. 1972). Less than 0.1% of the data required for that study was available (Coles 1974). Puccia & Levins comment on the utility of exhaustive data collection on ecological modelling:

> *In a complex system of only a modest number of variables and interconnections, any attempt to describe the system completely and measure the magnitude of all the links would be the work of many people over a lifetime* (*(Puccia and Levins 1985)*, p5).

[2] Notation: *x*↑ = "the value of x has gone up"; x↓ = "the value of x has gone down"; *x*θ = "the value of x is steady".

[3] Feuding experts do not willing surrender controversial portions of their models.

[4] Neuroendocrinological process may be controlled by minute levels of certain chemicals. Some of the values measured are in the pico-MOLE range (10^{-12}) and hard to detect. Detecting inter-relationships between entities is therefore difficult.

[5] Data collection in neuroendocrinology can be very expensive and, hence, incomplete. In one extreme case, 300,000 sheeps brains had to be filtered to extract 1.0 milligrams of purified thyroptin-releasing hormone (Krieger 1980). In the usual case, delicate measurements have to be made by skilled staff using expensive equipment. In the QMOD study used as our test suite, data on a glucose model (Smythe 1989) was collected from six journal articles. In all, none of the flow constants were known and only 34% of the variables were measured (Feldman, Compton et al. 1989; Feldman, Compton et al. 1989).

They claim that this observation from ecological modelling also applies to sociological models. For example, it is well known that many crimes go unreported. A literature review on crime statistics shows that the resources required to gather empirical data on the level on unreported crime is prohibitively high (Menzies 1985).

In our other KBS work, we have built expert systems for process control (Menzies and Markey 1987), farm management (Menzies, Black et al. 1992), consumer lending, and medical applications (Compton, Horn et al. 1989; Preston, Edwards et al. 1993). All this work can be characterised as the construction of models in data-poor domains:

- ICI Chemicals Australia required an automatic controller for one of its petrochemical plants. A mathematical controller could have been constructed, but would have necessitated the purchase of a set of parameter values from the engineering firm that built the plant. The purchase price was so high that ICI funded the development of a heuristic rule-based controller (Menzies and Markey 1987).
- As of 1988, Australia raised 4.8 million pigs a year (net worth $AUS 500 million). This represented one-third of one percent of the international pig herd (1,440,000,000 pigs). Despite the enormity of the international porcine enterprise, much of the internal physiology of the pig remains unmeasured. Building and verifying an expert system for raising pigs required the collection of new data, especially for that model (Menzies, Black et al. 1992). This data collection/ model development process took decades of work by CSIRO's senior experts in pig nutrition. The package was then sold on a one-off basis to an American manufacturer of feed stocks; i.e. once the data was collected, it was promptly bought and hidden by a private corporation for their exclusive use.
- Our commercial experiments with machine learning for consumer lending demonstrated that consumer loan approval is a poorly-measured domain. One financial organisation we worked for had large databases of prior loan approvals and defaulted loans. Much of this data pre-dated certain crucial changes in Federal legislation making it irrelevant for current use. Construction of expert systems for consumer lending was therefore a process of creating new models for new domains that were yet to be measured.

Note that there is no theoretical barrier to the accurate measurement of any value in any of the poorly-measured domains listed above. However, model construction is a resource-bounded activity. Organisations have limited staff, time, and money. The problems with data collection catalogued above may reflect a fundamental problem with numbers; i.e. *there exist useful numbers that we may wish to measure but lack the resources to collect.* Resource-bounded data collection implies a vague domain.

The one exception we are aware of to our general rule of "all non toy-domains are characterised by a lack of data and are therefore vague" is the diagnosis of electrical circuits (Hamscher, Console et al. 1992). Electrical circuits are an artificially constructed domain and it is theoretically possible to add instrumentation to circuitry to make all values accessible. However, even in that domain, we can find some concern with handling incomplete models (Console, Dupre et al. 1989; Abu-Hakima 1993).

We are not alone in this characterisation of the expert systems endeavour as "vague modelling". Clancey describes as expert systems as imprecise/qualitative models[6] (Clancey 1989). The knowledge modelling school (described below) acknowledge the imprecision of the models created via KA (Bradshaw, Ford et al. 1991; Wielinga, Schreiber et al. 1992; Davis, Shrobe et al. 1993). An acceptance of knowledge base models as context-dependant constructs (Phillips 1984; Shaw 1988; Bradshaw, Ford et al. 1991; Agnew, Ford et al. 1993) tacitly implies hypothetical domains (since non-hypothetical models would be context-independent).

Summary: Most KBS domains are vague. Indeed, if a domain can be characterised accurately, then the heuristic approach that characterises KBS would not be required.

[6] In using the phrase "qualitative model", Clancey is appropriating a term used extensively by the naive physics community. Numerous researchers explored qualitative reasoning in the domain of first-order linear differential equations. For more on this research, see (Coiera 1989; Iwasaki 1989).

3. WHAT IS A MODEL?

We cannot make general statements about testing models unless we commit to some definition of a model. This section gives our definition. This definition is in two parts: a symbol-level view and a philosophical view.

3.1. Symbol-Level View

We view a "model" M as a device for generating *explanations* of known *behaviour* B of the entity being modelled. B is a set of pairs of known inputs and outputs $<IN_i, OUT_i>$ that represent measurements of the entity being modelled. An *explanation* of $<IN_i, OUT_i>$ is the union of the proof trees whose roots are in IN_i and whose leaves are single members of OUT_i.

More precisely, M is a directed, possibly cyclic graph whose edges E represent possible explanations and whose vertices V are either (i) literals from the expert's domain of discourse or (ii) *and*-vertices. The graph may be implicit or explicit. In the explicit case, all edges are pre-enumerated and cached. In the implicit case, operators exist that can generate new vertices or edges from existing vertices and edges[7]. M can represent (amongst other things) the dependency graph between literals in a propositional theory, a rule-based expert system[8], a unfolded first-order system[9], a declarative frame-based systems[10], a qualitative reasoning system (see section 5), or some vague sketch of how an expert sees their domain (Menzies 1994). Elements of B are subsets of V. All proofs are subsets of E. A proof that contains an and-vertex must also contain all the parents of that vertex. Proofs that use non-and-vertices must contain zero or one parents of that vertex.

Devices that can produce explanations can also be used to predict behaviour[11]. In the case of explicit graphs, if we set IN_j to the known input and OUT_j to $V - IN_j$, then the generated "explanations" are predictions of what could result from IN_j.

Logical proofs can't contain loops. Hence, we cannot explain time-series data (e.g X went up, then later it went down, then it went up again) without significantly increasing the size of our models (i.e. create one vertex for each literal in M at each measured time intervals).

A model from a vague domain (M_V) may not necessarily be parsimonious, complete, deterministic, or consistent. The model may be able to explain only a subset of OUT_i . Of the explainable outputs, M_V may be able to generate $1 \leq N < \infty$ explanations ($N = 0$ means no explanation, $N > 1$ means multiple explanations). Further, our explanations may be able to generate contradictions. Therefore:

- Generalised test's search for explanations is really the processing of extracting from M a subset of E that covers the greatest number of elements from OUT_i .
- Explanations must not violate a library of invariants I. For example, I_1 says that we should not believe the literal p and $\neg p$ simultaneously. Also, a variable that can take one of N mutually exclusive states generates N literals in M_V. I_2 says that we should not simultaneously believe any more than one of those states. For convenience, I is defined in the negative; i.e. $I(x,y)$ is true when the combination of x and y violate some constraint.
- In the case of $N > 1$ explanations, some domain specific *BEST* operator chooses the preferred explanations. Example *BESTs* include returning the explanations that require the least number of unknown variables ($BEST_1$); with fewest number of inputs ($BEST_2$); with shortest proof size ($BEST_3$); with the largest number of explained effects ($BEST_4$:); or which avoid edges with low likelihood ($BEST_5$). $BEST_5$ assumes that such meta-knowledge about edges is available; e.g. some edges were

[7] E.G. in SOAR, operators can permit transitions from some current state to another newly created state (Rosenbloom, Laird et al. 1985).

[8] Less its conflict resolution strategy.

[9] That is, unfolded until it is ground (i.e. all variables bound).

[10] In partial-match systems, the disjunction of slots can lead to a frame. In total-match frame systems, the conjunction of slots can lead to a frame. In both, inference to a subclass can lead to inference to the superclass (e.g. i*f emu then bird*).

[11] An insight we first gained from Poole (Poole 1990).

proposed as part of a theory you wish to fault. $BEST_i$ may be indeterminate; i.e. some subset X of all known explanations may be ranked as the "best" explanations and $|X| > 1$. In the case of more than one world being "best", then generalised test returns them all.

Our results regarding the limits to testing are taken from the easy case where (i) the graph is explicit; (ii) I has an arity of two (i.e. given one literal, we can deterministically infer which other literals are inconsistent); and (iii) B does not include time-series data. In this easy case, numerous optimisations are possible[12]. To extend our results to full first-order theories/ implicit graphs/ invariants of arbitrary arity/ testing time series data, simply increase the runtimes.

3.2. Philosophical View

Models are synthetic constructs created by people which may or may not reflect some aspects of the entity being modelling. This view is consistent with recent trends in knowledge acquisition (KA) away from expertise-transfer[13] to model-construction[14] (Gaines 1992). In the modern KA view, knowledge bases are only ever an approximate surrogates of reality (Bradshaw, Ford et al. 1991; Wielinga, Schreiber et al. 1992; Davis, Shrobe et al. 1993). "Knowledge" extracted during KA from an expert is a report customised to the specific problem, the specific justification, the expert, and the audience (Feldman, Compton et al. 1989; Compton and Jansen 1990). "Truth" as expressed by human experts varies according to who says it (Phillips 1984; Bradshaw, Ford et al. 1991; Agnew, Ford et al. 1993) and even when they say it (Shaw 1988).

Models are created by people and people can often reason idiosyncratically or sub-optimally. Kuhn notes that data is not interpreted neutrally, but (in the usual case) is processed in terms of some dominant intellectual paradigm (Kuhn 1962). Silverman cautions that systematic biases in expert preferences may result in incorrect/incomplete knowledge bases (Silverman 1992). Other writers issue similar warnings:

> *…expert-knowledge is comprised of context-dependent, personally constructed, highly functional but fallible abstractions (Agnew, Ford et al. 1993).*
>
> *Human reasoning does not always correspond to the prescriptions of logic. People … fail to see as valid certain conclusions that are valid, and they see as valid certain conclusions that are not[15]. ((Anderson 1985) p264)*
>
> *The same decision can be framed in several different ways; different frames can lead to different decisions[16]. ((Kahneman and Tversky 1982) p139)*

Apart from idiosyncrasies added by their authors, models will always exhibit one or more behaviours that the entity being modelled will not. This must be so since the model is different to the thing being modelled (the map is not the territory). Puccia & Levins comment:

> *A model is an intellectual construct we study instead of studying the world. Every model distorts the system under study in order to simplify it. ((Puccia and Levins 1985), p2)*

[12] E.G. (i) assign a unique integer id to each literal and use bitstring processing for all set manipulations; (ii) for each literal L_i, pre-compute and cache as a single bitstring the integer ids of the other literals inconsistent with L_i.

[13] E.G. The Feigenbaum perspective of "mining the jewels in the expert's head" (Feigenbaum and McCorduck 1983).

[14] E.G. KADS (Wielinga, Schreiber et al. 1992). For a critique of KADS and KADS-like proposals, see (Menzies and Compton 1994) and section 6.6.

[15] When presented a modus tollens syllogism (i.e. $P \rightarrow Q$, *not* Q, therefore *not* P) 39% of subjects incorrectly stated it was only sometimes true while 4% wrongly stated that it was never valid In another study, 90% of subjects incorrectly understood a syllogism, including trained logicians; presumably, the most rational of all human beings (Anderson 1985).

[16] In one study, physicians consistently choose one of two options according to the way a problem was framed. Both options were actually identical, but one was expressed in terms of absolute numbers and the other in terms of percentages. The problem was framed in terms of lives-saved or lives-lost. Physicians presented with the lives-saved frame were generally risk-avoiding; i.e. they elected to maximised the absolute number of lives saved. Physicians presented with the lives-lost frame were risk-seeking; i.e. elected to minimise the percentage of lives lost (Kahneman and Tversky 1982).

Our general thesis is that (i) the distortions added by the modelling process are non-trivial; and hence (ii) we must always test models. If the reader remains unconvinced, they are invited to review a one-line mathematical model of exponential population growth: $EQ_1 : dN/dT = rN$. In EQ_1, r is a constant that is positive or negative if the environment is benevolent or hostile respectively, T is time, and N is the population. Note that this model is wrong[17] since population growth must taper off as it approaches C the maximum carrying capacity of the environment; i.e. $EQ_2 : dN/dT = rN(1-(N/C))$. If the reader can correctly answer the following question, then we have anecdotal evidence for believing that humans can read and critique models: *is* EQ_2 *correct?* If the reader cannot find all errors in a one-line model (which they probably studied extensively in high school), then we should be suspicious of claims that the truth status of larger models can be accurately determined by people.

EQ_2 is incorrect[18]. Our experience has been that the error is not apparent to many people. Myers and Feldman & Compton provide us with more examples of models defying human critique:

- Myers reports controlled experiments in which 59 experienced data processing professionals hunted for errors in a very simple text formatter (63 line of PL/1 code). Even with unlimited time and the use of three different methods[19], the experts could only find (on average) 5 of the 15 errors in this 63 line model (Myers 1977).
- Feldman & Compton used a technique called *hypothesis testing* to show that neuroendocrinological theories published in international referred journals contain a surprisingly high percentage of errors. In one study, 109 of 343 (32%) of the known data points from six studied papers could not be explained using a glucose regulation modelled developed from international refereed publications (Feldman, Compton et al. 1989; Feldman, Compton et al. 1989; Smythe 1989). A subsequent study (Menzies 1995) corrected some modelling errors of Feldman & Compton to increase the inexplicable percentage from 32% to 45%. A similar study successfully faulted another smaller published scientific theory (Menzies, Mahidadia et al. 1992). Menzies' generalisation of the hypothesis testing technique is the generalised-test-as-abduction algorithm described below.

Puccia & Levins (Puccia and Levins 1985) use EQ_1 and EQ_2 to argue that models are not universal truths, but are merely useful constructs for a particular context. Demonstrating the universal truth of any model is impossible. All proofs of "truth" terminate in some premises which must be accepted on faith (Popper 1963)[20]. Note that this pessimism about the truth status of models is not justified for all models. Models tested over long periods of time may asymptote to some satisfactory competency. However, most of the knowledge inserted into expert systems does not fall into this category. Gaines & Shaw comment:

> *In a well-established scientific domain it is reasonable to suppose that there will be consensus among experts as to the relevant distinctions and terms- that is* objective knowledge t*hat is independent of individuals. However, the "expert systems" approach to systems development has been developed for domains where such objective knowledge is not yet available. (Gaines and Shaw 1989)*

For example non-objective domains, consider a sample of problems tackled by modern expert systems: how to configure a computer (Bachant and McDermott 1984); where to dig for minerals (Campbell, Hollister et al. 1982); how to diagnosis biochemical disorders (Preston, Edwards et al. 1993); how to best run a petro-chemical

[17] If you cannot detect the error before reading on, then Q.E.D.

[18] In the case of a hostile environment and over-population, $N > C$, $r < 0$, and our intuition is that the population will fall. However,$rN(1-(N/C)) > 0$; i.e. the maths says that population will *increase*! (Puccia and Levins 1985)

[19] (i) Reading the 30 line specification, then generating test cases which were run through an executable version of the program; (ii) As before, but also reading the 63 line code listing; (iii) As with (ii), but testing was done via manual walk-throughs/inspections. Programmers only used one of (i), (ii) or (iii). Programmers using (i) and (ii) worked alone. Programmers using method (iii) worked in groups of three.

[20] For example, consider one individual trying to reproduce all the experiments that lead to our current understanding of atomic physics. Such an undertaking could longer than a lifetime and would be beyond the resources of most individuals (e.g. building an five kilometre linear accelerator). Such a task has to be divided up and, sooner or later, our single researcher would have to *accept on faith* the validity of another researcher's statement that "while you were busy elsewhere, I did this, and I saw that. Trust me.".

plant, (Menzies and Markey 1987); what antibiotics we should prescribe (Buchanan and Shortliffe 1984); and our test domain of multiple expert KA for neuroendocrinology.

Our general claim is that any model is only trustworthy in the same context where it was developed and tested. As soon as a model moves "out-of-context", it may generate inappropriate results. Sadly, models are rarely labelled with their context boundaries. When all available knowledge and examples is used to generate a model, out-of-context is never indicated since it represents the area(s) unexplored during development. We believe that all models should be tested whenever new data is available regarding its proper behaviour. Pragmatically, this means that every working expert system should be accompanied by a test engine continually in operation.

Summary: Knowledge bases containing non-objective knowledge are hypothetical constructs which may generate inappropriate behaviour in certain contexts. Therefore, we need to test these knowledge bases as best we can prior to use. Note that these tests will never certify a model as correct (Popper 1963). We must always re-test a model when new data becomes available on its appropriate behaviour. This implies that "test" must be an on-going procedure for the entire life-cycle of a model.

4. WHAT TO TEST?

Any program can be assessed according to (i) an internal assessment of its internal structures or (ii) an external assessment of its ability to fulfil some function. Our preferred definition of "test" focuses on the ability of a model to provide explanations of known behaviour (i.e. $BEST_4$). This definition of "test" is silent regarding the "best" internal form of the model. This section defends that position.

The KBS verification and validation (V & V) community have numerous techniques for internal assessment. After constructing a dependency network between rules in a knowledge base, an automatic process can detect circularities, inappropriate dead-ends, missing logic (seen as isolated literals), repeated logic, and redundant/ subsumed logic (Suwa, Scott et al. 1982; Nguyen, Perkins et al. 1987; Preece and Shinghal 1992). More sophisticated systems import the rule-base into a truth maintenance system (TMS) and compute the worlds that include each conclusion. Zlatareva uses a JTMS-variant (Doyle 1979; Zlatareva 1992; Zlatareva 1993) while Ginsberg uses an ATMS (DeKleer 1986; Ginsberg 1987; Ginsberg 1990). For example, if no ATMS context includes a conclusion, then it can never be made. With their detailed knowledge of inter-dependencies, these TMS-style validation tools can also critique a test suite or propose new test cases. For example, if the minimum labels of the worlds within a KBS are not represented in a test suite, then the test suite is incomplete. Test cases can then be automatically proposed to fill any gaps. The advantage of this technique is that it can be guaranteed that test cases can be generated to exercise all branches of a knowledge base. The disadvantage of this technique is that, for each proposed new input, an expert must still decide what constitutes a valid output. This decision requires knowledge external to the model, lest we introduce a circularity in the test procedure (i.e. we test the structure of M using test cases derived from the structure of M). Further, auto-test-generation focuses on incorrect features in the current model. We prefer to use test cases from a totally external source since such test cases can highlight what is absent from the current model. For these reasons, we caution against automatic test case generation.

The standard external assessment technique is *test suite assessment*. The inputs and outputs of the rule base are identified and a library is built of input/output pairs representing the expected output given the input. The inputs are then run against the model and the output compared with the expectation. External testing is harder than internal testing since it implies making a decision about the correct behaviour of a system in a wide range of circumstances. This is an expert task and a lengthy analysis process. Further, in vague domains, information for test suite construction is limited. External testing is therefore harder to apply and is rarely reported in the literature (exceptions: MYCIN (Yu, Fagan et al. 1979), CASNET (Weiss, Kulikowski et al. 1978), Garvin ES1 (Compton, Horn et al. 1989), PEIRS (Preston, Edwards et al. 1993), PIGE (Menzies, Black et al. 1992), QMOD (Feldman, Compton et al. 1989; Feldman, Compton et al. 1989; Menzies, Mahidadia et al. 1992)). Nevertheless, external testing is better than internal testing:

- We have stressed above the hypothetical nature of expert system models. The crucial test of such models is not some report of their internal structure. Rather, we must assess the connection of such vague models to the entities they are trying to mimic.

- Programs in routine use can fail internal tests, yet still be deemed useful. Preece & Shingal detected multiple internal test failures in fielded expert systems (Preece and Shinghal 1992) (see Table 1). Yet these systems were passing their day-to-day external operational test (i.e. their behaviour was acceptable)[21].

	Application					
	MMU	*TAPES*	*NEURON*	*DISPLAN*	*DMS1*	*Hit Rate*
Size (literals)	105	150	190	350	540	
Subsumption	0	5/5	0	4/9	5/59	14/73 = 19%
Missing rules	0	16/16	0	17/59	0	33/75 = 44%
Circularities	0	0	0	20/24	0	20/24 = 83%

Table 1. *Some internal errors detected in fielded expert systems. Fractions represent* anomalies/faults. Anomalies *were detected automatically.* Faults *are anomalies that were assessed by the experts to be true errors. The* hit rate *is the fraction of all anomalies divided by all errors. Note that the* hit rate *is much less than 100%., From (Preece and Shinghal 1992).*

Zlatareva & Preece comment:

> *It is widely accepted that the only direct way to measure the actual level of KBS performance is to test the KBS on a set of test cases with known solutions (Zlatereva and Preece 1994).*

Opponents of this view may argue that (i) the validity of a model can be measured by more than just its ability to cover certain test cases; (ii) other measures such as parsimony, succinctness, and clear use of existing domain concepts are just as important as performance; and (iii) a narrow focus merely on performance could give rise to incorrect and truly bizarre models indeed. We have argued previously that tacit in this objection is a belief in the now-out-dated knowledge-transfer approach (Menzies and Compton 1994). We endorse the new knowledge modelling perspective and believe that it is folly to aim for the "right" or "correct" knowledge base for an expert system. More pragmatically, we note that external testing can resolve arguments faster than internal tests. Feuding experts may not decide between alternative models if they all cover the same number of cases. However, an expert will concede defeat when their favoured model explains less of the known behaviour than the opposition's model.

Summary: Coverage of a test suite (e.g. $BEST_4$) is a more important test than internal assessment.

5. GENERALISED TESTING IN VAGUE DOMAINS

This section develops a generalised computational model of "test" for vague domains.

External testing in non-vague domains is very simple (using a process we call $TEST_0$). First, we compute the transitive closure of IN_i (IN^*). If any inconsistencies are detected (i.e. violations of I) or if OUT_i is not covered $(OUT_i - IN^* \neq \emptyset)$ then the model is definitely wrong.

External testing in vague domains is more complicated. A model may generate inconsistencies or not cover OUT_i while still representing our best understanding of the domain to date. One model may be "better" than all others without being able to reproduce all known behaviours. While the entire model may generate inconsistencies, some subset of the model may still be able to offer consistent explanations of some subset of OUT_i. In vague domains, the goal of testing is not "is this model the absolutely correct" but "what contribution do portions of this model make to explaining portions of the known behaviour?".

Testing in vague domains is complicated by their poorly measured and indeterminate nature. In indeterminate models, *assumptions* must be made. In data poor domains, these guesses cannot be checked. Mutually exclusive assumptions must be maintained in separate *worlds*; i.e. maximal subsets of compatible proofs. Any world W_x

[21] Preece and Shinghal discuss how this might occur. A rule base's inference engine can tame a subsumption anomaly (e.g.) by always picking the rule with the largest satisfied condition. Missing logic may reflect the "do nothing" default case or reflect some reasoning that is out-of-scope for a particular domain. Circularities may exist as part of some looping process that terminates in a special condition (e.g. prompting the user for input until they provide us with a satisfactory answer) (Preece and Shinghal 1992) .

contain a set of literals L_x and can be characterised by (i) its inputs ($L_x \cap IN_i$); (ii) the number of outputs it covers ($L_x \cap OUT_i$); (iii) the assumptions it make ($L_x - IN_i - OUT_i$); and (iv) their *base controversial assumptions*, i.e. the assumptions in a proof that can violate the invariants *I*, but are not dependant on other controversial assumptions.

For example, consider the explicit and-or graph tacit in Figure 1 (see Figure 2) and the case of $<IN_1, OUT_1> = <\{a\uparrow, b\uparrow\}, \{d\uparrow, e\uparrow, f\downarrow\}>$. All states incompatible with $IN \cup OUT$ are marked *FORBIDDEN*, i.e. cannot be used in proofs[22]. Applying I_2, the *FORBIDDEN* set is $\{a\downarrow, a\theta, b\downarrow, b\theta, d\downarrow, d\theta, e\downarrow, e\theta, f\uparrow, f\theta\}$.

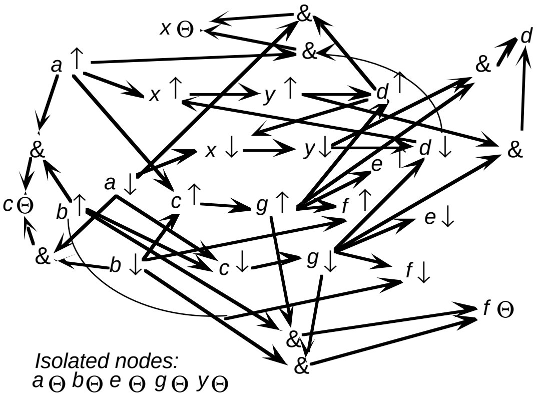

Figure 2: *The tacit and-or graph within Figure 1 assuming that (i) a vertex can go either* up, down, *or* steady*; (ii) the conjunction of an* up *and* down *can explain a* steady*; (iii) no change can be explained in terms of a* steady *(i.e.* steady *vertices have no children). And-vertices are denoted "&"; for example,* $a\uparrow$ *&* $b\uparrow \Rightarrow c\theta$ *(see far left-hand-side of diagram).* Steady *nodes that do not connect to the rest of the graph are shown bottom-left.*

The possible explanations are the proof trees *P* from Figure 2; i.e.: $P_1=\{a\uparrow, x\uparrow, y\uparrow, d\uparrow\}$, $P_2=\{a\uparrow, c\uparrow, g\uparrow, d\uparrow\}$, $P_3=\{a\uparrow, c\uparrow, g\uparrow, e\uparrow\}$, $P_4=\{b\uparrow, c\downarrow, g\downarrow, f\downarrow\}$, and $P_5=\{b\uparrow, f\downarrow\}$. The literals of *P* that do not appear in $<IN_1, OUT_1>$ are the assumptions $A = \{x\uparrow, y\uparrow, c\uparrow, g\uparrow, c\downarrow, g\downarrow\}$. Applying I_2, we see that the controversial assumptions A_c are $\{c\uparrow, g\uparrow, c\downarrow, g\downarrow\}$. Since *g* depends on *c* (see Figure 1), the base controversial assumptions A_b are $\{c\uparrow, c\downarrow\}$.

The minimal environments (*MinEnv*) are defined to be the empty set, plus all the maximal consistent subsets of A_b; i.e. $MinEnv_0 = \{\}$, $MinEnv_1 = \{c\downarrow\}$, $MinEnv_2 = \{c\uparrow\}$. Each $MinEnv_i$ has an associated exclusion X_i being all members of A_b which, when combined with X_i, would violate *I*; i.e. $X_0=\{c\uparrow, c\downarrow\}$, $X_1=\{c\uparrow\}$, $X_2=\{c\downarrow\}$. A proof belongs in world W_i if it does not intersect the assumptions that are illegal in that world; i.e. X_i. For example: (i) P_1 and P_5 belong in all worlds since they do not overlap with any member of *X*; (ii) P_2 does not belong in W_0 or W_1 since it uses $c\uparrow$. Note that W_0 represents the world where no controversial assumptions are made.

Our computed worlds are hence $W_0=\{P_1, P_5\}$, $W_1=W_0+\{P_2, P_3\}$, and $W_2=W_0+\{P_4\}$. The proofs in these worlds explain (cover) different numbers of outputs; i.e. $cover(W_0)=|d\uparrow, f\downarrow|=2$, $cover(W_1)=|d\uparrow, f\downarrow, e\uparrow|=3$, $cover(W_2)=|d\uparrow, f\downarrow|=2$. Applying $BEST_4$, we would declare that the model can fully explain $<IN_1, OUT_1>$ using W_1 (i.e. assuming $c\uparrow$).

More intuitively, we characterise *generalised test* as the extraction of some subset of the total knowledge base which can connect IN_i to OUT_i. Multiple subsets (worlds) are possible. We favour the subset that can reaches the largest subset of the outputs (see Figure 3).

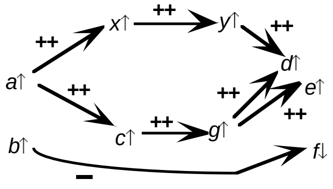

Figure 3: *The subset of figure 2 contained in* W_1*; i.e. the edges that offer explanations of the largest subset of* $OUT_1 = \{d\uparrow, e\uparrow, f\downarrow\}$ *given* $IN_1=\{a\uparrow, b\uparrow\}$*. Note while figure 2 condones inferences between b & c, g & f, and d & x, generalised test has elected not to use these connections.*

Note the difference between testing in vague and non-vague domains. The model of Figure 1 plus $<IN_1, OUT_1>$ can generate inconsistencies (e.g. $c\uparrow$ and $c\downarrow$) and would be categorically rejected by $TEST_0$. Generalised test is

[22] A knowledge engineer can add other vertices to *FORBIDDEN* in order to cull the search space.

more fine-grained. While all of Figure 1 is not correct, it is incorrect to say that it is entirely useless (i.e. as implied by *TEST$_0$*). The subset of Figure 1 shown in Figure 3 is useful for explaining known behaviour. If an expert wished to change Figure 1, then generalised test gives them a principled basis for deciding which sections of Figure 1 are useful or useless.

Generalised test cannot be fully characterised in terms of classical deduction which demands that if the rule *x if y* exists, then in every world where *x* is true, *y* is also true (Poole 1990). Generalised test is *abductive*: from the space of possible inferences supplied by the user, a subset has been selected according to some criteria in order to fulfil a certain task. More precisely:

> *Generalised test (abduction) using BEST$_4$ is the generation all worlds* $W_x \subseteq E$ *that are the union of edges in a subset of explanatory proofs* P *such that (i)* COVERED $\subseteq$ OUT$_j$*; (ii)* CAUSES $\subseteq$ IN$_j$; *(iii)* $W_x \cup$ CAUSES $\vdash$ COVERED using the edges in *W_x; (iv)* $W_x \cup$ USED $\nvdash$ false *i.e. does not violate* I*; (v)* W_x *is maximal with respect to number of included proofs; (vi) COVERED is maximal with respect to set size.*

We find that this definition of abduction is compatible with the standard definition of abduction as proposed by Poole (Poole 1990; Poole 1990), Eshghi (Eshghi 1993), and Selman & Levesque (Selman and Levesque 1990). Standard abduction generates any world that satisfies points (i) to (v). Generalised test is *exhaustive abduction* (Menzies 1994); i.e. all worlds are generated, then assessed. Recognising the connection between generalised test and abduction allows us to use complexity results from the abductive literature to make strong claims regarding the limits to abductive inference/ testing.

Abduction is known to NP-hard; i.e. very likely to be computational intractable in the worst-case. Recall that generalised test is a search for a subset of the supplied model which can explain the largest subset of known behaviour. Given a model with $|E|$ edges, then there are $2^{|E|}$ possible subsets; i.e. the number of subsets varies exponentially with model connectivity. For more formal proofs of the NP-hard nature of abduction, see (Selman and Levesque 1990; Bylander, Allemang et al. 1991).

An interesting feature of abduction is that this worst-case behaviour is often the usual case: most known abductive inference engines exhibit exponential runtimes for real-world inputs, even for sophisticated algorithms (e.g. the ATMS (Selman and Levesque 1990)). Hence, many of the articles in (O'Rourke 1990) are concerned with heuristic optimisations of abduction. Eshghi report a class of polynomial-time abductive inference problems, but this class of problems requires at least a non-cyclic and-or graph. (Eshghi 1993). Bylander reports techniques for tractable abduction (Bylander, Allemang et al. 1991), but many of these techniques (e.g. rule-out knowledge to cull much of the search space) are not applicable to hypothetical models in poorly measured domains (e.g. neuroendocrinology).

Early versions of our generalised test used a basic chronological backtracking approach (i.e. no memoing) that was very slow. Basic chronological backtracking has the disadvantage that any feature of the space learnt by the search algorithm is forgotten when backtracking on failure. The current abductive implementation, HT4, deduces the base controversial assumptions as a side-effect of proof-tree generation. That is, the system learns features of the search space without backtracking. This system runs 130 times faster than our previous Feldman & Compton system (Feldman, Compton et al. 1989; Feldman, Compton et al. 1989) since world switching does not require extensive further computation. The runtime behaviour of HT4 has been studied extensively in (Menzies 1995). Two experimental results from that study are relevant here:

- The *Changing N study* artificially generated 94 models of varying numbers of vertices ($N = |V|$) while keeping B ($|E|/|V|$) constant at 2.25. *Changing B (other models) study* used the *Changing N study* model generator to produce six new models. These new models were then mutated between a fanout of $2 \leq B \leq 10$.
- Two statistics were collected: *runtime* and *% explicable*. Runtimes were collected to see if generalised test was indeed exponential as predicted by the formal complexity results of abduction. The maximum number of explainable outputs (*% explicable*) was collected since it was suspected that an indeterminate non-trivial model could offer an explanation for any behaviour at all. If so, then generalised test would "pass" all models; i.e. it losses its ability to distinguish between different types of models.

Figure 4 shows the results. Experimentally, we see that HT4 is limited to $N < 850$ and $B < 7$ (though after $B > 4.5$, critiquing power is very low). $B = 7$ seems to be the limit, beyond which a model can explain all behaviours. We call $B = 7$ the *Pendrith limit* and argue that it is a fundamental limit to testing. The exponential behaviour seen in Figure 4.i indicates that HT4 does not avoid the NP-hard nature of abductive inference. Note that these runtimes were not a result of a naive implementation. HT4 is the fourth generalised test engine we have constructed and contains numerous optimisations including: (i) avoiding chronological backtracking; (ii) discovering the definitions of the minimal worlds (the base controversial assumptions); (iii) pre-computing and caching the search space (the explicit and-or graph) prior to timing the inference runs; and (iv) using bitstrings to optimise the set processing (for details, see (Menzies 1995)). The N limit is only an approximation which may be broken by (e.g.) faster machines and faster languages. However, due to the exponential nature of the process, we would not expect orders of magnitude increases in N. Therefore we hesitate to place a firm limit on the model size that is testable (hence our earlier statement that generalised test is limited to models with less than $N = |V| = 1000$ (approx) vertices).

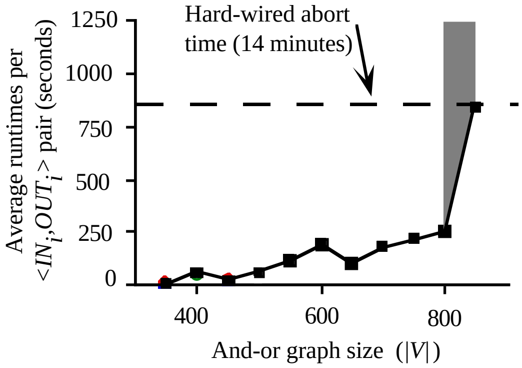

Figure 4.i: *Runtimes for 1991* $<IN_i,OUT_i>$ *pairs. Language: Smalltalk/V. Machine: Macintosh Powerbook 170. None of the models over |V|=800 terminated within the built-in "give-up" time limit of 14 minutes. Conclusions: (i) the average runtime at |V| = 850 is > 14 minutes; (ii) the runtime curve grows into the gray area shown on the right; (iii) the runtime curve seems exponential.*

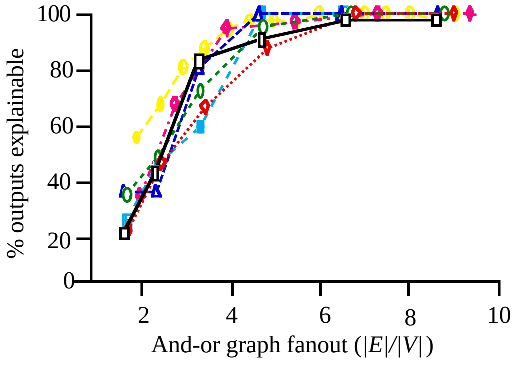

Figure 4.ii: *% effects explainable for 1631* $<IN_i,OUT_i>$ *pairs. Conclusions: (i) Most behaviours can be explained after a fanout = 4.4; (ii) nearly all behaviours can be explained after fanout = 6.8.*

On the positive side, based on known sizes of fielded expert systems (Preece and Shinghal 1992), we have argued elsewhere that these N and B limits are larger than models we see in contemporary knowledge engineering practice; i.e. we can use generalised testing for knowledge bases at least as big as those seen in current practice (Menzies 1995). Further, the level of critique offered by generalised test can be non-trivial. Figure 4.ii shows that at low B values, up to 75% of behaviour may be falsifiable.

6. APPLICATIONS OF ABDUCTION/GENERALISED TEST

The previous section noted that the computational kernel of generalised test is abduction. We have discussed above how to use generalised test for verification and prediction. In this section we will argue that at a pragmatic engineering level, it is useful to view inference in various domains as abduction (i.e. as being isomorphic with generalised test).

6.1. Model-Based Diagnosis

The connection between abduction and model-based diagnosis is well documented. Pople and Reggia acknowledge that their "diagnosis" systems are really abduction (Pople 1973; Reggia 1985). Poole's abductive framework *THEORIST* can be used as a diagnosis tool (Poole 1988; Poole 1989; Poole 1990). Console and Torasso characterise the two main types of diagnosis as variants of the same abductive inference algorithm (Console and Torasso 1991). Both types of diagnosis input (i) a system description of the system to be diagnosed (i.e. a model[23]); (ii) a set of observations *OBS*; (iii) and a context *CXT* in which the *OBS* were made. Two sets are then deduced: (i) a set of observables that must be avoided Ψ^- (i.e. any observables inconsistent with *OBS)*; and (ii) a set of observables that must be covered Ψ^+. *Consistency-based diagnosis* (e.g. (Genesereth 1984; DeKleer and Williams 1987; Reiter 1987)) sets $\Psi^+ = \emptyset$ while *set-covering diagnosis* (e.g. (Console, Dupre et al. 1989; Poole 1989)) sets Ψ^+ to *OBS*. Set-covering diagnosis is best when the knowledge base contains knowledge of faulty operations while consistency-based diagnosis is best for knowledge bases containing knowledge of normal operation. For a comparison of the two approaches, see Figure 5.

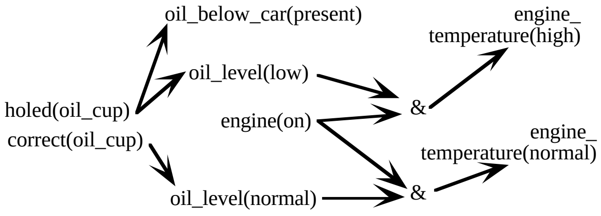

Figure 5: *If* OBS ={oil_below_car(present)}, *and* CXT = {engine(on)}, *and we restrict the diagnosis to* oil_cup, *then set covering diagnosis returns* {holed(oil_cup)} *while consistency-based diagnosis returns* {holed(oil_cup)} *or* {correct(oil_cup)}. *Example from* (Console and Torasso 1991).

Konologie argues for the primacy of set-covering diagnosis. He notes that consistency-based diagnosis returns answers that may not be relevant to causal explanations of *OBS* (Koniligue 1992) (e.g. *correct(oil_cup)* from Figure 5 has little bearing on the problem of *oil_below_car(present)*).

Having stressed the differences, we note that generalised-test can be used for either consistency-based or set-covering diagnosis (see Table 2).

	Generalised test configured for set-coverage diagnosis	*Generalised test configured for consistency-based diagnosis*
IN	CXT	CXT
OUT	OBS	V - *CXT*
FORBIDDEN	$\Psi^- = \{x \mid x \in V, y \in CXT \cup OBS ,I(x,y)\}$	$\Psi^- = \{x \mid x \in V, y \in CXT \cup OBS ,I(x,y)\}$
BEST	all world(s) that cover all of $\Psi^+ = OBS$	all worlds

Table 2: *Generalised test configured for different model-based diagnosis tasks of a model with vertices* V. *For set-coverage, explanations are attempted for all* OBS. *For consistency-based diagnosis, explanations are attempted for all non-input vertices. The* FORBIDDEN *definition says that any vertex that contradicts* CXT ∪ OBS *is forbidden. Note that generalised test is a generalisation of model-based diagnosis since (i)* BEST *permits the generation of worlds with partial coverage of* OBS*; and (ii) the* BEST *operator allows for the customisation of the world preference criteria.*

6.2. Explanation

Leake (Leake 1991) and Paris (Paris 1987; Paris 1989) discuss explanation algorithms where explanation presentation is constrained to those explanations which contain certain *significant structures*. Paris's significant structures are determined at design time while Leake assigns significance at runtime.

- Paris's experimental results suggest that expert's use parts-based explanations while novices use process-based explanations. Edges in her system are tagged as being part either *process-based* or *parts-based.* Knowledge of the expertise of the audience is used to tag each vertex as "known to user" or "unknown to user". When faced with a choice of edges to be used in an explanation, Paris's explanation algorithm selects either a process-based trace or a parts-based trace according to an examination of the local vertices in the network. If the local vertices are "unknown", then the process-based descriptions are preferred.
- Leake assigns significance at runtime according to a set of eight pre-defined algorithms. For example, when the goal of the explanation is to minimise undesirable effects, the runtime significant structures are any pre-conditions to anomalous situations .

[23] Console and Torasso further divide a model into a set of components COMP and knowledge of the behavioural modes BM of those components. We would input <BM,COMP> and output our and-or graphs.

Leake clearly acknowledges the connection of his work with abduction (Leake 1993). Note that both Leake's and Paris's algorithms can be characterised in terms of operators that select some subset of the possible inferences according to a user/goal-specific criteria; i.e. they are compatible with our *BEST* formalism.

6.3. Classification

Applying the same notion of "significant structures", we can adapt our symbol-level algorithm to a variety of classification algorithms. Consider the dependency graph of a rule-base developed for classification purposes. The possible output classification *CLASSES* are a subset of all the literals in the knowledge base (e.g. all the literals with in-edges, but no out-edges).

- To perform simple classification with generalised test, we use the prediction algorithm described above, plus a *BEST* that favours the worlds with the most number of *CLASSES*. Note that this would be a single and multiple-classification system. Multiple classifications that were specified as incompatible would appear in separate worlds. Therefore, the *N* classifications in a world would be mutually compatible (*N=1* means single classification, *N >1* means multiple classification).
- To perform a style of heuristic classification, we add *is-a* edges to the knowledge base from sub-classes to super-classes (e.g. *animal if dog; organicThing if animal or plant*). Now we use a *BEST* that uses favours the worlds with the most number of *CLASSES* and *is-a* edges.

6.4. Other Knowledge-Level Tasks

Leake characterises case-based reasoning as an abductive process where the possible explanations are assessed via a library of prior cases (Leake 1993). Hamscher notes that certain sub-tasks in financial reasoning (financial assessment, going concern evaluation, auditing, and the explanation of unexpected financial results) are all abductive tasks (Hamscher 1990). Numerous papers in (O'Rourke 1990) describe natural language comprehension in terms of abduction.

Hirata characterises inductive logic programming (a technique for learning first-order theories) as an abductive process where the search space for explanations is either in the current theory (*selecting abduction*), an analogous theory (*finding abduction*), or a theory especially created from a generalisation of known theories (*generating abduction*). More generally, Hirata argues that scientific theory formation is an abductive process (Hirata 1994).

Poole maps design and recognition into abduction:

- Design is the process of hypothesising components which would imply a design goal (Poole 1990).
- Visual pattern recognition is a process of hypothesising scene objects which would lead to the perceived image. Poole demonstrates that this perspective gives the same results as other visual imagery researchers (Poole 1990).

6.5. Model Extraction

The list of applications that can be processed by abduction is surprisingly long. In retrospect, however, it is a requirement of our architecture that it supports such a long list. "Generalised test" using test suite assessment implies a generalised execution module. If we could not map our test procedure into a wide range of inference types, then our architecture would be a failure.

We believe that the wide-applicability of testing-as-abduction is no mere coincidence. Abduction directly operationalises the subset extraction process that is the core of Clancey's *model construction operators* (Clancey 1992) and Breuker's *components of problem solving types* (Breuker 1994). Clancey characterises expert system inference as constructing the system-specific model (SSM) from a general qualitative model (QM) in the KB. Breuker explores the relationships between problem solving techniques used in expert systems (i.e. modelling, planning, design, assignment, prediction, assessment, monitoring and diagnosis) (Breuker 1994). He offers an abstract description of the "components of a solution" generated by these techniques. An a*rgument structure* is extracted from a *case model* (analogous to Clancey's SSM) represents some understanding of a problem. This case model is generated from some *generic domain model* (analogous to Clancey's QM). A *conclusion* is some portion of the argument structure that is relevant to the user. In the case where all the solution components are represented as a ground theory of literals whose dependency graph has edges E, then:

$$\mathit{edges(answer)} \subseteq \mathit{edges(argument\ structure)} \subseteq \mathit{edges(case\ model)} \subseteq (\mathit{edges(generic\ domain\ model)} = E)$$

where *edges(X)* denotes the edges of the dependency graph present in X. Note the commonality between Clancey's and Breuker's view: expert system inference is the extraction of some subset theory from a super theory.

Returning now to HT4, we note that this algorithm also extracts sub-models from super-models (e.g. Figure 3 is a subset of Figure 2). However, what we call "worlds", Clancey calls "SSMs" and Breuker calls "case models". The extracted models (worlds) are relevant to a particular problem, defined as $<<IN_i$, $OUT_i>$, $BEST_j>$, and is guaranteed to be consistent. Note that each solution is some subset of the generic domain model M which is the space of all possible solutions. Also, deduction over an HT4 world can generate a Breuker-style *argument structure* for a particular *conclusion* (some vertex in M). Further, abduction/HT provides a uniform structure for processing of many of problem solving types listed by Breuker. Such uniformity simplifies the construction of interfaces between the inputs and outputs of different problem solving types. Breuker argues that such interfacing is essential since most problem solving types are used in combination to perform some task.

6.6. Discussion

Generalised test requires an inference engine that (i) inputs dependency knowledge and contradiction knowledge between a set of literals, then (ii) executes its inferences in separate, internally consistent, worlds. Formally, this is abduction. We have seen above that an inference engine that supports abduction is a useful tool for KBS validation, prediction, set-covering model-based diagnosis, consistency-based model-based diagnosis, explanation, classification, learning, case-based reasoning, financial reasoning, natural language comprehension, design, and recognition. Table 3 contrasts this list with the inference models proposed by knowledge-level methodologies.

System Analysis	Abduction
-- Diagnosis	-- Validation
----- Single Model Diagnosis	-- Prediction
------ Systematic Diagnosis	-- Classification
-------- Localisation	---- Simple classification
-------- Causal Tracing	---- Heuristic classification
-- Verification	-- Explanation
-- Classification	-- Diagnosis
---- Simple Classification	---- Set-covering-based diagnosis
---- Heuristic Classification	---- Consistency-based diagnosis
-- Prediction	-- Design
---- Prediction of Behaviour	-- Recognition
---- Prediction of Values	-- Case-based reasoning
System Synthesis	-- Natural language understanding
-- Design	
(i)	(ii)

Table 3: *Comparison of (i) knowledge-level inference models from conventional knowledge-level modelling (from* (Tansley and Hayball 1993)*) and (ii) knowledge-level inference models based on our symbol-level abductive inference.*

We approve of some of the clusterings in the hierarchy of Table 3.i. For example, localisation and causal tracing are basically the same inference, except the former using *part-of* knowledge while the latter uses *causal* knowledge. In terms of or framework, both execute over the same and-or graph but the user's interpretation of the edges differs. However, we doubt (e.g.) the separation of design from diagnosis in Figure 3.i since we have seen above that they are closely connected.

To be fair, the authors of Table 3.i acknowledge that many of their distinctions have an arbitrary nature. For example, they note that (i) heuristic classification could be used for diagnosis; or (ii) that scheduling, planning, and configuration are actually the same problem, divided on two dimensions ("goal states known or not" and ""temporal factors considered or not"- see Figure 12.3 of (Tansley and Hayball 1993)). However, they do not take the next step and simplify their hierarchy according to these observed similarities in the processing. We believe that such a simplification would remove certain complexities from conventional knowledge-level modelling:

- Table 3.ii is broader and flatter than Table 3.i; i.e. a hierarchy of inference models based on abduction requires fewer distinctions than conventional knowledge-level modelling.
- When we compare the differences between sub-hierarchies between Table 3.ii and Table 3.i, we see that our sub-hierarchies are much smaller variants of their roots than in conventional knowledge-level modelling. For example, the difference between set-covering-based diagnosis and consistency-based diagnosis is very small (see Table 2). In contrast, sibling inference models in conventional knowledge-level modelling may have totally different inference models (e.g. KADS has totally different inference models for *correlation* and *verification*, even though they are both sub-types of *identification*).

7. RELATED WORK

7.1. Ripple-Down Rules

We have argued above that generalised test replaces rather than merely augments existing KBS practice. Compton's ripple-down-rules (RDR) approach is another example where a comprehensive test program replace the need for other methodologies (Compton and Jansen 1990). RDR is interesting in that, to our knowledge, it is the only KA technique we know that ensures 100% behaviour coverage as the system evolves. Compton's design is optimised for maintenance of propositional rule-bases only. Experts cannot browse and recognise their models inside the patch tree. Our work here began as an experiment in a maintenance environment where experts could browse and freely modify the knowledge. Despite these restrictions, RDR has been used to develop systems that are larger and more successful than other techniques applied to the same domains (Preston, Edwards et al. 1993; Menzies and Compton 1994).

7.2. Other KB Test Regimes

We have mentioned above the work of the V&V community on KBS assessment and noted their emphasis on internal testing. For reviews of the state of the art in V&V, see (Preece 1992; Zlatereva and Preece 1994).

Gaines & Shaw explore techniques for resolving conflicts in terminology using repertory grids. The conceptual systems of different experts are explicated and compared using the grids (Gaines and Shaw 1989). Their work focuses on resolving conflicts in the meaning of individual terms, not on conflicts in the semantics of the models built using those terms as primitives.

Boose *et al.* describes group decision support environments containing suites of tools combine to form a KA environment (Boose, Bradshaw et al. 1992). Boose's *et al.* system focuses on the development of models of the group decision support process. Their environment lacks an execution module for the generated models as part of the group decision support environment; i.e. Boose *et al* assume that once the group's mode is elicited, it will be subsequently exported into an executable form. We believe that an active abductive evaluation module would enhance their architecture. Groups could execute their under-specified intuitions to gain feedback on their ideas.

Silverman advises that attached to an expert system is an *expert critiquing system* which he defines as:

> *…programs that first cause their user to maximise the falsifiability of their statements and then proceed to check to see if errors exist. A good critic program doubts and traps its user into revealing his or her errors. It then attempts to help the user make the necessary repairs*[24]. (Silverman 1992)

Silverman divides an expert critiquing system into (i) a *deep model* which can generate behaviour; (ii) a *differential analyser* which compares the generated behaviour with the expected behaviour; and (iii) a *dialogue generator* that explains the errors and assists in correcting them. Silverman's research seems to be focused on an implementation-independent analysis of the process of "critiquing" a program. His focus appears to be on defining "critiquing" as an add-on to existing systems. We believe that critics should not be tacked on as an after thought since built-in critics could guide the KA process. Our work could be described as the development of

[24] ECSs are therefore much broader than the definition instantiated by ATTENDING (Miller 1986) which had no mechanism for doubting its own internal knowledge base.

general implementation principles for *deep models* and *differential analysers*. Mahidadia explores *dialogue generation* in this domain using inductive logic programming (Mahidadia, Sammut et al. 1992; Mahidadia, Sammut et al. 1994).

7.3. World Assessment

Operators for implementing preference criteria for assessing possible worlds has been widely discussed in the literature. Most researchers argue that the best worlds must at least cover all the known output. Some argue that the "best" explanation is the smallest one (e.g. (Reggia, Nau et al. 1983; Console, Dupre et al. 1991)). Poole (Poole 1985) and Console *et. al.* (Console, Dupre et al. 1991) have proposed the additional criteria that the "best" explanation also uses the most specific terms from a taxonomic hierarchy; e.g. they prefer explanations in terms of *emu* rather than in terms of the more general term *bird*.

We prefer not to hard-wire world assessment into our formalism. We take the line of Leake and Paris to argue that "best" is a person-specific and goal-specific criteria. While experts may prefer explanations in terms of the most specific term in a hierarchy, novices may prefer more general explanations. We have argued above maximal output coverage may be a more important world assessment criteria than parsimony or total output coverage. World assessment knowledge is still domain-specific knowledge and should be customisable.

7.4. SOAR

We have argued that abduction is an appropriate single inference procedure for a wide-variety of knowledge-based tasks. The designers of SOAR make a similar claim regarding their state space traversal (Rosenbloom, Laird et al. 1985; Laird, Newell et al. 1987). SOAR was built to operationalism Newell's original notion of the knowledge level (Newell 1982). The primitives of the SOAR rule-based language explicitly represent Newell's model of human cognition as a search for appropriate operators that convert an agent's current state to a desired goal state. Like our abductive approach, minor manipulations of SOAR's operator space (which is controlled by rules) are all that is required to fundamentally change the inferencing (Laird, Newell et al. 1987). Also, like SOAR, we do not build knowledge bases around the knowledge-level inference models used by conventional knowledge-level modelling techniques (e.g. KADS). The observation that a knowledge base is performing (e.g.) classification is a user-interpretation of a lower-level inference (Yost and Newell 1989).

Generalised-test-as-abduction differs from SOAR in two ways. (i) SOAR's executes over an implicit and-or graph while we prefer to execute over an explicit and-or graph. Efficiency is a non-trivial issue in generalised test (see the remarks in the last section regarding the NP-hard nature of abduction). Building and caching the search space prior to inferencing is one technique for taming complexity[25]. (ii) Given a vertex with *N* out edges (or, in SOAR-speak, a state space with *N* associated operators), generalised test assesses the utility of each edge using a deferred global analysis. SOAR must make its operator assessment at the local level. SOAR's run-time selective generation of the and-or graph has efficiency advantages since it culls unacceptable alternatives as they are first encountered. Our approach is slower, but the explicit representation of all alternatives permits allows for global assessment criteria (e.g. $BEST_4$).

7.5. Belief Networks

Belief networks (BNs) deduce causality from a statistical analysis of the frequency distributions of variables in a sample to deduce acyclic "networks" (which are really trees) of causal relationships between variables (Pearl and Verma 1991). BNs assume sufficient measurements are available for the statistical analysis; i.e. they are inappropriate for vague domains. Also, current state-of-the-art BNs assumes acyclic models (Geigner, Paz et al. 1993) and models in our test domain are usually cyclic (e.g. Figure 1). Further, the theories generated by BNs make do not preserve current beliefs (see above remarks regarding preserving the background theory). We foresee that various users would treasure their favourite portions of their model(s) (typically, the ones they have developed and successfully defended from all critics). It would be unacceptable to permit a learning algorithm

[25] Note that this is only practical for *finite theories*. An example of an *infinite theory* is the space of all even numbers; e.g. `even(2). even(X) :- even(Y), X is Y * 2.`

scribble all over this knowledge. Learning programs for this domain must strive to preserve the current background theory (hence Mahidadia's interest in ILP for this domain).

7.6. Default Logic

Our base controversial assumptions and worlds are akin to ATMS labels (DeKleer 1986) and default logic extensions (Reiter 1980) respectively. However, we differ from ATMS/ default logic in that our worlds only contain *relevant* literals (i.e. only those literals that exist on pathways between inputs and outputs). This means that, unlike default logic extensions, not all consequences of a literal exist in a world containing that literal. For example, consider the following example:

```
model:        b if a; c if a; e if (b or c);
              d if b; f if c; z if y.
contradicts:  {d,f}
<IN2,OUT2> =  <{a},{e}>
```

Generalised test would generate two proofs which could exist in the one world, i.e. $W_1 = \{P_1, P_2\}$, $P_1=\{a,b,e\}$, $P_2=\{a,c,e\}$. Standard ATMS/ default logic would analyse all literals in the model to generate two extensions (one with *d* and the other with *f*). Both of these extensions would contain the same proofs of *e* in terms of *a*. We view these two worlds as irrelevant and wasted computation.

8. CONCLUSION

We have argued that testing is fundamental to the KBS process. Given the modern view of KA as the construction of approximate models, it follows that the contents of our KBs are potentially inaccurate reflections of the entities being modelled. Potentially inaccurate models must be tested prior to their wide-spread use. Further, we have argued that in order to prevent models moving inappropriately "out of context", we must continually and routinely test delivered models whenever new data is available on the entity that they are trying to model. Automatic test engines are required for such a continuous test regime.

Viewed in its most general form, "test" in vague domains is not neutral to KBS practice:

- Limits to test are the limits to model construction in vague domains. Experimentally, we have seen testing limits of $|V|$ = 1000 (approx) and $|E|/|V|$ = 7. We have argued that most KBS domains are vague; i.e. most of KBS practice is similarly limited.
- From a pragmatic engineering viewpoint, we have argued that is useful to replace multiple single-purpose mechanisms with one general-purpose mechanism, particularly when (i) the services offered by the general mechanism is a superset of the services offered by the multiple mechanisms; and (ii) the general mechanism is simpler than the single-purpose mechanisms. Current KBS inference engines do not support generalised test while generalised test also supports the model extraction process that Breuker and Clancey argue is at the core of expert systems inference. That is, we should execute our inference modules with generalised test modules.

If testing is such a fundamentally limiting process as we have argued above, then why has this not been reported previously in the literature? Clearly, testing is the exception rather than the rule in the KA community. This is an observable fact which we find surprising. To us, the need for "test" follows inevitably from the knowledge modelling approach. We suspect that many so-called "knowledge modellers" still believe in expertise-transfer since only in expertise-transfer are we so confident in our models that we do not need to test them[26]. We seek here to discourage such beliefs.

Finally, we comment on the pessimism of some researchers who lament the end of the age of certainty[27]. If we reject a Platonic view of the universe and its somewhat spurious belief in an absolute "truth", we need not plunge

[26] O'Hara notes that some knowledge representation theorists still make occasional claims that their knowledge representation theory has some psychological basis (even though, when pressed, their public line is that representations are models/ surrogates only) (O'Hara and N. 1993).

[27] For example, Agnew *et. al.* forcibly argue for the non-absolute nature of human knowledge. However, between the lines, we read that they do not like their general conclusion, but reluctantly acknowledge its inevitability

into confusion. Our implementation experience has been that structured testing environments such as RDR and generalised-test-as-abduction are highly ordered entities. Recall our comparative analysis of abductive inference models versus knowledge level inference modules. We argued that abductive inference modules can unify and clarify knowledge level inference models. That is, non-Platonic architectures are still amenable to rigorous analysis. The opposite to Platonic certainty need not be chaos.

9. REFERENCES

Abu-Hakima, S. (1993). *Automating Knowledge Acquisition in Diagnosis*. **Fourth International Workshop on Principles of Diagnosis**, University College of Wales, Aberyswyth, Wales, United Kingdom,

Agnew, N.M., K.M. Ford and P.J. Hayes (1993). *Expertise in Context: Personally Constructed, Socially elected, and Reality-Relevant*? **International Journal of Expert Systems** 7(1):

Anderson, J.R. (1985). **Cognitive Psychology and its Implications**. New York, W.H. freeman and Company.

Bachant, J. and J. McDermott (1984). *R1 Revisited: Four Years in the Trenches*. **AI Magazine** (Fall): 21-32.

Boose, J.H., J.M. Bradshaw, J.L. Koszareck and D.B. Shema (1992). *Knowledge Acquisition Techniques for Group Decision Support*. **Proceedings of the 7th Knowledge Acquisition for Knowledge-Based Systems Workshop**, Banff, Canada,

Bradshaw, J.M., K.M. Ford and J. Adams-Webber (1991). *Knowledge Representation of Knowledge Acquisition: A Three-Schemata Approach*. **6th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, ,October 6-11 1991**, Banff, Canada,

Breuker, J. (1994). *Components of Problem Solving and Types of Problems*. **8th European Knowledge Acquisition Workshop, EKAW '94**,

Buchanan, B.G. and E.H. Shortliffe (1984). **Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project**. Addison-Wesley.

Bylander, T., D. Allemang, M.C. Tanner and J.R. Josephson (1991). *The Computational Complexity of Abduction*. **Artificial Intelligence** 49: 25-60.

Campbell, A.N., V.F. Hollister, R.O. Duda and P.E. Hart (1982). *Recognition of a Hidden Material Deposit by and Artificially Intelligent Program*. **Science** 217(3 September): 927-929.

Clancey, W. (1989). *Viewing knowledge bases as qualitative models*. **IEEE Expert** (Summer): 9-23.

Clancey, W.J. (1992). *Model Construction Operators*. **Artificial Intelligence** 53: 1-115.

Coiera, E. (1989). *Reasoning with Qualitative Disease Histories for Diagnostic Patient Monitoring*. Department of Computer Science, University of NSW.

Coles, H., S (1974). **Thinking About the Future: A Critique of the Limits to Growth**. Sussex University Press.

Compton, P., K. Horn, J.R. Quinlan and L. Lazarus (1989). *Maintaining an Expert System*. **Applications of Expert Systems**. London, Addison Wesley. 366-385.

Compton, P.J. and R. Jansen (1990). *A philosophical basis for knowledge acquisition*. **Knowledge Acquisition** 2: 241-257.

Console, L., D. Dupre and P. Torasso (1989). *A theory of diagnosis for incomplete causal models*. **Proc. 11th IJCAI**, Detroit,

Console, L., D.T. Dupre and P. Torasso (1991). *On the Relationship Between Abduction and Deduction*. **Journal of Logic Programming** 1(5): 661-690.

Console, L. and P. Torasso (1991). *A spectrum of definitions of model-based diagnosis*. **Computational Intelligence** 7(3): 133-141.

Davis, R., H. Shrobe and P. Szolovits (1993). *What is a Knowledge Representation*? **AI Magazine** (Spring): 17-33.

DeKleer, J. (1986). *An Assumption-Based TMS*. **Artificial Intelligence** 28: 163-196.

DeKleer, J. and B.C. Williams (1987). *Diagnosing multiple faults*. **Artificial Intelligence** 32(1): 97-130.

Doyle, J. (1979). *A Truth Maintenance System*. **Artificial Intelligence** 12: 231-272.

Eshghi, K. (1993). *A Tractable Class of Abductive Problems*. **IJCAI '93**, Chambery, France,

Feigenbaum, E. and P. McCorduck (1983). **The Fifth Generation**. New York, Addison-Wesley.

Feldman, B.T., P.J. Compton and G.A. Smythe (1989). *Towards Hypothesis Testing: JUSTIN, Prototype System Using Justification in Context*. **Proceedings of the Joint Australian Conference on Artificial Intelligence, AI '89**,

Feldman, B.Z., P.J. Compton and G.A. Smythe (1989). *Hypothesis Testing: an Appropriate Task for Knowledge-Based Systems*. **4th AAAI-Sponsored Knowledge Acquisition for Knowledge-based Systems Workshop**, Banff, Canada, October 1989.,

Gaines, B. (1992). *AAAI 1992 Spring Symposium Series Reports: Cognitive Aspects of Knowledge Acquisition*. **AI Magazine.** 24.

Gaines, B.R. and M.L.G. Shaw (1989). *Comparing the Conceptual Systems of Experts*. **IJCAI '89**, Detroit, USA,

(Agnew, Ford et al. 1993).

Geigner, D., A. Paz and J. Pearl (1993). *Learning Simple Causal Structures*. **International Journal of Intelligent Systems** 8: 231-247.

Genesereth, M.R. (1984). *The Use of Design Descriptions in Automated Diagnosis*. **Artificial Intelligence** 24: 411-436.

Ginsberg, A. (1987). *A new approach to checking knowledge bases for inconsistentcy and redundancy*. **Proc. 3rd Annual Expert Systems in Government Conference**, IEEE Computer Society.

Ginsberg, A. (1990). *Theory Reduction, Theory Revision, and Retranslation*. **AAAI '90**,

Hamscher, W. (1990). *Explaining Unexpected Financial Results*. **AAAI Spring Symposium on Automated Abduction**,

Hamscher, W., L. Console and D. J., Ed. (1992). **Readings in Model-Based Diagnosis**. San Mateo CA, Morgan Kaufmann.

Hirata, K. (1994). *A Classification of Abduction: Abduction for Logic Programming*. **Proceedings of the Fourteenth International Machine Learning Workshop, ML-14**,

Iwasaki, Y. (1989). *Qualitative Physics*. **The Handbook of Artificial Intelligence**. Addison Wesley. 323-413.

Kahneman, D. and A. Tversky (1982). *The Psychology of Preferences*. **Scientific American** 246: 136-142.

Konolige, K. (1992). *Abduction versus Closure in Causal Theories*. **Artificial Intelligence** 53: 255-272.

Krieger, D.T. (1980). *The Hypothalmus and Neuroendocrinology*. **Neuroendocrinology**. Sunderland, Massachusetts, Sinauer Associates, Inc. 3-122=.

Kuhn, T. (1962). **The Structure of Scientific Revolutions**. New York, Cambridge Press.

Laird, J.E., A. Newell and P.S. Rosenbloom (1987). *Soar: An Architecture for General Intelligence*. **Artificial Intelligence** 33(1): 1-64.

Leake, D.B. (1991). *Goal-Based Explanation Evaluation*. **Cognitive Science** 15: 509-545.

Leake, D.B. (1993). *Focusing Construction and Selection of Abductive Hypotheses*. **IJCAI '93**,

Levesque, H. (1989). *A Knowledge-Level Account of Abduction (preliminary version)*. **IJCAI '89**, Detroit, Michigan, USA,

Mahidadia, A., C. Sammut and P. Compton (1994). *Applying Inductive Logic Programming to Causal Qualitative Models in Neuroendocrinology*. **AI '94**, Armidale, NSW,

Mahidadia, A.J., C. Sammut and P. Compton (1992). *Building and Maintaining Causal Theories*. **AAAI Symposium on Knowledge Assimilation**, Stanford University, Spring, 1992.,

Meadows, D.H., D.L. Meadows, J. Randers and W.W. Behrens (1972). **The Limits to Growth**. Potomac Associates.

Menzies, G.D. (1985). *An Econometric Analysis of the Dark Figure of Crime*. Honours Thesis; University of New England.

Menzies, T. (1995). *Principles for Generalised Testing of Knowledge Bases*. PhD Thesis, University of New South Wales.

Menzies, T., A. Mahidadia and P. Compton (1992). *Using Causality as a Generic Knowledge Representation, or Why and How Centralised Knowledge Servers Can Use Causality*. **Proceedings of the 7th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop**, Banff, Canada, October 11-16,,

Menzies, T.J. (1994). *Exhaustive Abduction: A Practical Model Validation Tool*. **ECAI '94 Workshop on Validation of Knowledge-Based Systems**, Amsterdam, Holland,

Menzies, T.J. (1994). *A Precise Semantics for Vague Diagrams*. **AI'94**, Armidale, Australia,

Menzies, T.J., J. Black, J. Fleming and M. Dean (1992). *An Expert System for Raising Pigs*. **The first Conference on Practical Applications of Prolog**, London, UK.,

Menzies, T.J. and P. Compton (1994). *Knowledge Acquisition for Performance Systems; or: When can "tests" replace "tasks"*? **Proceedings of the 8th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop**, Banff, Canada,

Menzies, T.J. and B.R. Markey (1987). *A Micro-Computer, Rule-Based Prolog Expert-System for Process Control in a Petrochemical Plant*. **Proceedings of the Third Australian Conference on Expert Systems, May 13-15**, Sydney, Australia,

Miller, P.L. (1986). **Expert Critiquing Systems**. Spriner-Verlag.

Myers, G.J. (1977). *A Controlled Experiment in Program Testing and Code Walkthroughs/Inspections*. **Communications of the ACM** 21(9, September): 760-768.

Newell, A. (1982). *The knowledge level*. **Artificial Intelligence** 18: 87-127.

Nguyen, T.A., W.A. Perkins, T.J. Laffey and D. Pecora (1987). *Knowledge Base Verification*. **AI Magazine** (Summer): 69-75.

O'Hara, K. and S. N. (1993). *AI Models as a Variety of Psychological Explanation*. **IJCAI'93**, Chambery, France,

O'Rourke, P. (1990). *Working Notes of the 1990 Spring Symposium on Automated Abduction*. University of California, Irvine, CA.

Paris, C. (1987). *Combining discourse strategies to generate descriptions along a naive/expert spectrum*. **IJCAI '87**, Milan, Italy,

Paris, C.L. (1989). *The Use of Explicit User Models in a Generation System for Tailoring Answers to the User's Level of Expertise*. **User Models in Dialog Systems**. Springer-Verlag. 200-232.

Pearl, J. and T.S. Verma (1991). *A Theory of Inferred Causation*. **Principles of Knowledge Representation and Reasoning, Proceedings of the Second International Conference**, Morgan Kaufmann.

Phillips, L.D. (1984). *A Theory of Requisite Decision Models*. **Acta Psychologica** 56: 29-48.

Poole, D. (1985). *On the Comparison of Theories: Preferring the Most Specific Explanation*. **IJCAI '85**,

Poole, D. (1988). *Representing knowledge for logic-based diagnosis*. **Proc. of the Int. Conf. on Fifth Generation Computer Systems**, Tokyo,

Poole, D. (1989). *Normality and Faults in Logic-Based Diagnosis*. **IJCAI '89**, Detroit, USA,

Poole, D. (1990). *Hypo-Deductive Reasoning for Abduction, Default Reasoning, and Design*. **Working Notes of the 1990 Spring Symposium on Automated Abduction.**, UC Irvine.

Poole, D. (1990). *A Methodology for Using a Default and Abductive Reasoning System*. **International Journal of Intelligent Systems** 5: 521-548.

Pople, H.E. (1973). *On the mechanization of abductive logic*. **IJCAI '73**,

Popper, K.R. (1963). **Conjectures and Refutations,**. London, Routledge and Kegan Paul.

Preece, A.D. (1992). *Principles and Practice in Verifying Rule-based Systems*. **The Knowledge Engineering Review** 7(2): 115-141.

Preece, A.D. and R. Shinghal (1992). *Verifying Knowledge Bases by Anomaly Detection: An Experience Report*. **ECAI '92**, Vienna,

Preston, P., G. Edwards and P. Compton (1993). *A 1600 rule expert system without knowledge engineers*. **Second World Congress on Expert Systems**, Lisbon, Pergamon.

Puccia, C.J. and R. Levins (1985). **Qualitative Modelling of Complex Systems: An Introduction to Loop Analysis and Time Averaging**. Cambridge, Mass., Harvard University Press.

Reggia, J., D.S. Nau and P.Y. Wang (1983). *Diagnostic expert systems based on a set covering model*. **Int. J. of Man-Machine Studies** 19(5): 437-460.

Reggia, J.A. (1985). *Abductive Inference*. **Proceedings of the Expert Systems in Government Symposium**, Washington, D.C.,

Reiter, R. (1980). *A Logic for Default Reasoning*. **Artificial Intelligence** 13: 81-132.

Reiter, R. (1987). *A theory of diagnosis from first principles*. **Artificial Intelligence** 32(1): 57-96.

Rosenbloom, P., J. Laird, J. McDermott, A. Newell and E. Oruich (1985). *R1-Soar: An Experiment in Knowledge-Intensive Programming in a Problem-Solving Architecture*. **IEEE Transactions on Pattern Analysis and Machine Intelligence** PAMI-7(No. 5, September): 561-569.

Selman, B. and H.J. Levesque (1990). *Abductive and Default Reasoning: a Computational Core*. **AAAI '90**,

Shaw, M.L.G. (1988). *Validation in a Knowledge Acquisition System with Multiple Experts*. **Proceedings of the International Conference on Fifth Generation Computer Systems**,

Silverman, B.G. (1992). *Survey of Expert Critiquing Systems: Practical and Theoretical Frontiers*. **Communications of the ACM** 35(4): 106-127.

Smythe, G.A. (1989). *Brain-hypothalmus, Pituitary and the Endocrine Pancreas*. **The Endocrine Pancreas**. New York, Raven Press.

Suwa, M., A.C. Scott and E.H. Shortliffe (1982). *An Approach to Verifying Completeness and Consistency in a Rule-based Expert System*. Department of Computer Science, University of Stanford.

Tansley, D.S.W. and C.C. Hayball (1993). **Knowledge-Based Systems Analysis and Design**. Prentice-Hall.

Weiss, S.M., C.A. Kulikowski and S. Amarel (1978). *A Model-Based Method for Computer-Aided Medical Decision-Making*. **Artificial Intelligence** 11(145-172):

Wielinga, B.J., A.T. Schreiber and J.A. Breuker (1992). *KADS: a modelling approach to knowledge engineering*. **Knowledge Acquisition** 4(1): 1-162.

Yost, G.R. and A. Newell (1989). *A Problem Space Approach to Expert System Specification*. **IJCAI '89**,

Yu, V.L., L.M. Fagan, S.M. Wraith, W.J. Clancey, A.C. Scott, J.F. Hanigan, R.L. Blum, B.G. Buchanan and S.N. Cohen (1979). *Antimicrobial Selection by a Computer: a Blinded Evaluation by Infectious Disease Experts*. **Journal of American Medical Association** 242: 1279-1282.

Zlatareva, N. (1992). *CTMS: A General Framework for Plausible Reasoning*. **International Journal of Expert Systems** 5(3): 229-247.

Zlatareva, N. (1993). *Distributed Verification and Automated Generation of Test Cases*. **IJCAI '93 workshop on Validation, Verification and Test of KBs**, Chambery, France.

Zlatereva, N. and A. Preece (1994). *State of the Art in Automated Validation of Knowledge-Based Systems*. **Expert Systems with Applications** 7(2): 151-167.