Assessing Responses to Situated Cognition

Tim Menzies

Dept. of Artifical Intelligence, School of Computer Science and Engineering,
The University of New South Wales, Sydney, Australia, 2052
timm@cse.unsw.edu.au; http://www.cse.unsw.edu.au/~timm *

September 17, 1996

Abstract

Situated cognition (SC) claims that knowledge is mostly context-dependent and that symbolic descriptions elicited prior to direct experience are less important than functional units developed via direct experience with the current problem. If this were true, then we would need to modify the knowledge modeling approaches of KA which assume that re-using old symbolic descriptions are a productivity tool for new applications. There are numerous tools which, if added to conventional knowledge modeling, could be said to handle SC (e.g. machine learning, abduction, verification & validation tools, repertory grids, certain frameworks for decision support systems, expert critiquing systems, and ripple-down-rules). However, we require an experiment to assess the effectiveness of these tools as a response to SC.

1 Introduction

"What is wanted is not the will to believe, but the will to find out, which is the exact opposite." – Bertrand Russell

"Measure what is measurable, and make measurable what is not so." – Galileo

Proponents of situated cognition (SC) in the AI field (e.g. [Suchman, 1987, Suchman, 1993, Clancey, 1987, Clancey, 1991, Clancey, 1993, Agre, 1990, Agre, 1993, Winograd & Flores, 1987, Compton & Jansen, 1990, Birnbaum, 1991, McDermott, 1987]) assert that symbolic descriptions elicited prior to direct experience are less important than functional units developed via direct experience with the current problem (§3). More precisely, they argue that:

Claim 1 The SC Premise: *Human cognition cannot be accurately modeled by context-independent assertions (§3.1) because...*

Claim 2 Weak SC:... *The inferences of a symbolic model interacting with its environment are heavily constrained/ controlled/ changed, by the inputs from that environment. That is, using knowledge in a particular context will significantly change that knowledge (§3.2). Given this, then...*

Claim 3 Strong SC:... *Since the influence of the environment is so great, we should use pure reactive systems that interact directly with the environment without first reflecting over some symbolic description (§3.3).*

SC is hence a challenge to knowledge acquisition strategies that rely of the reuse of old symbolic descriptions (e.g. problem solving methods (PSMs) or ontologies) when building new applications (§2). There are numerous *potential* responses to the challenge of the SC including ignoring it (§4.1), verification & validation tools (§4.2), repertory grids (§4.3), expert critiquing systems (§4.4), machine learning (§4.5), certain frameworks for decision support systems (§4.6), and ripple-down-rules (§4.7)). However, we cannot assess the effectiveness of these potential responses since the few "experiments" in the KA field are poorly controlled (§3.4). A new experiment is therefore proposed which can assess the utility of these various responses (§5.2).

*Proceedings of the AAAI sponsored Banff Knowledge Acquisition Workshop, November, 1996. Available from *http://www.cse.unsw.edu.au/ ~timm/pub/ docs/papersonly.html*

2 Brief Notes on Knowledge Modeling

This section is a brief review of knowledge modeling. For more information, see the *Related Work* section of [Wielinga *et al.*, 1992] and [Menzies, 1995b]. See also the ontology literature (e.g. [Gruber, 1993]) which assumes that declarative descriptions of portions of old expert systems are useful for building new applications.

In Newell's KL approach, intelligence is modeled as a search for appropriate *operators* that convert some *current state* to a *goal state*. Domain-specific knowledge as used to select the operators according to *the principle of rationality*; i.e. an intelligent agent will select an operator which its knowledge tells it will lead the achievement of some of its goals. When implemented, this KL is built on top of a *symbol-level* containing data structures, algorithms, etc. However, to a KL agent, these sub-cognitive symbol-level constructs are the tools used "sub-consciously" as it performs its KL processing [Newell, 1982].

Newell's subsequent exploration of the KL lead to a general rule-based language called SOAR [Rosenbloom *et al.*, 1993] which was the basis for the problem-space computational model (PSCM) [Yost & Newell, 1989]. Programming SOAR using the PSCM involves the consideration of multiple, nested problem spaces. Whenever a "don't know what to do" state is reached, a new problem space is forked to solve that problem. Newell concluded that the PSCM was the bridge between SOAR and true KL modeling [Newell *et al.*, 1991,Newell, 1993].

There is a difference between PSCM (hereafter, $\mathcal{KL}_A$) and $\mathcal{KL}_B$, a KL-modeling variant which groups together a set of authors who argue for basically the same technique; i.e. Clancey's model construction operators [Clancey, 1992], Steels' components of expertise [Steels, 1990], Chandrasekaran's task analysis, SPARK/ BURN/ FIREFIGHTER (SBF) [Marques *et al.*, 1992] and KADS [Wielinga *et al.*, 1992]. The fundamental premise of $\mathcal{KL}_B$ is that a knowledge base should be divided into domain-specific facts and domain-independent PSMs.

In terms of this paper, the key difference between $\mathcal{KL}_A$ and $\mathcal{KL}_B$ is how much architecture they impose on a solution: $\mathcal{KL}_B$ imposes more than $\mathcal{KL}_A$. PSMs are only implicit in $\mathcal{KL}_A$. The observation that a PSCM system is performing (e.g.) classification is a user-interpretation of a lower-level inference (operator selection over a problem space traversal) [Yost & Newell, 1989]. In $\mathcal{KL}_B$, PSMs specify the data structures required for each method. In $\mathcal{KL}_B$, once a PSM is initially specified, it is assumed to be set in stone for the life of the project.

It will be argue below that an SC-aware KA tool must minimise the its architectural assumptions. Further, whatever is built within those architectural assumptions must be customisable. One basis issue with $\mathcal{KL}_B$ is that extensive customisation is not supported, particularly of the PSM. Our preferred response to SC includes a PSM customisation tool (§5.1).

3 Situated Cognition

3.1 About the SC Premise

Dreyfus argues that the context-dependent nature of human knowledge makes it fundamentally impossible to reproduce in symbolic descriptions [Dreyfus, 1979]. Searle takes a similar stand, claiming that the only device that can replicate human intelligence is another human [Searle, 1980,Searle, 1982,Searle, 1995] since only humans can share the same context. Birnbaum stresses "the role of a concrete case in reasoning" [Birnbaum, 1991, p58] and how logical AI cannot correctly handle such specifics, particularly when we have a specific conflicting belief.

Relativity, Heisenburg's uncertainty principle, the indeterminacy of quantum mechanics and Gödel's theorem demonstrate hard limits to the complete expression of truth. Many twentieth century thinkers have therefore adopted a relativist knowledge position. Kuhn notes that data is not interpreted neutrally, but (in the usual case) processed in terms of some dominant intellectual paradigm [Kuhn, 1962]. Popper [Popper, 1963] argues that, ultimately, we cannot prove the "truth" of anything since "proofs" must terminate on premises. If we request proofs of premises, then we potentially recurse forever. Hence, on purely pragmatic grounds, people are forced into an acceptance of certain premises. Note that the chosen premises may radically influence the conclusions reached. Agnew, Ford & Hayes offer their summary of contemporary thinking in the history, philosophy and sociology of science as:

> Expert-knowledge is comprised of context-dependent, personally constructed, highly functional

but fallible abstractions [Agnew *et al.*, 1993].

Easterbrook [Easterbrook, 1991] argues that it is undesirable to demand that knowledge bases are consistent.

> This insistence that expertise must be consistent and rational imposes restrictions of the knowledge acquired. The knowledge acquisition process becomes not so much the modeling of the expert's behaviour, but the synthesis of a domain model which need not resemble any mental model used by the expert [Easterbrook, 1991, p264].

The experience with expert systems is that the process of building consensus between individuals or creating an explicit record of it in a knowledge base introduces biases/errors. Silverman cautions that systematic biases in expert preferences may result in incorrect/incomplete knowledge bases (§4.4). Preece & Shinghal [Preece & Shinghal, 1992] document five fielded expert systems that contain numerous logical anomalies (see Figure 1). These expert systems still work, apparently because in the context of their day-to-day use, the anomalous logic is never exercised.

	MMU	TAPES	NEURON	DISPLAN	DMS1
Size (literals)	105	150	190	350	540
Logical subsumption errors	0	5	0	4	5
Missing rule errors	0	16	0	17	0
Circularities in reasoning errors	0	0	0	20	0

Figure 1: Samples of Errors in Fielded Expert Systems. From [Preece & Shinghal, 1992].

Shaw reports an experiment where a group of geological experts built models for the same domain, then reviewed each other's KBs as well as their own twelve weeks later [Shaw, 1988]. Note the two context changes: from expert to expert and also a change of twelve weeks. For the twelve week self-review study, it was found that an expert's understandability and agreement with their own knowledge was less than total (see Figure 2.A). For example, expert E_1 only understands three-fifths of her own thinking three months ago. For the cross-expert review, it was found that experts disagree significantly with each other (see Figure 2.B). In this cross-review study, it was found that levels of understanding may be low (e.g. expert E_1 only understands expert E_2's knowledge base 31.2% of the time). Levels of agreement were found to be even lower. For example, expert E_1 only agreed with expert E_2's knowledge base 8.3% of the time.

A. 12 weeks later, self-review

Expert	% understands	% agrees
E_1	62.5	81.2
E_2	77.8	94.4
E_3	85.7	78.6

B. Cross-expert review

Expert pairs	% understands	% agrees
E_1, E_2	62.5	33.3
E_2,E_1	61.1	26.7
E_1,E_3	31.2	8.3
E_3,E_1	42.9	33.3
E_2,E_3	44.4	20.0
E_3,E_2	71.4	33.3

Figure 2: The Shaw study [Shaw, 1988].

The Shaw study suggests that building a knowledge base representing consensus knowledge can be difficult. There is evidence for this elsewhere. For example, between the various camps of $\mathcal{KL}_B$ researchers, there is little agreement on the internal details. Contrast the list of "reusable" problem solving methods from KADS [Wielinga *et al.*, 1992] and SBF [Marques *et al.*, 1992] (termed "knowledge sources' and "mechanism" respectively). While there is some overlap, the lists are different. Also, the number and nature of the problem solving methods is not fixed. Often when a domain is analysed using $\mathcal{KL}_B$, a new problem solving method is required [Linster & Musen, 1992]. Further, different interpretations exist of the same problem solving method. For example:

Model	% disorders identified	% knowledge fragments identified
1 (Epistemological)	50	28
2 (KADS)	55	34
3 (no model)	75	41

Figure 3: Analysis via different models in the Corbridge study [Corbridge *et al.*, 1995].

- The problem solving method proposed by Bredeweg [Bredeweg, 1992] for prediction via qualitative reasoning is different to the qualitative prediction problem solving method proposed by Tansley & Hayball [Tansley & Hayball, 1993].
- The KADS problem solving methods for diagnosis [Wielinga *et al.*, 1992] is very different to the assumption-space exploration model proposed by the model-based diagnosis community [Hamscher *et al.*, 1992]. Note that the KADS diagnosis model continues to change at a rapid rate (e.g. [Benjamins, 1995]).

Knowledge developed in one context may not be usefully reusable in another. Corbridge *et. al.* report a study in which subjects had to extract knowledge from an expert dialogue using a variety of abstract pattern tools [Corbridge *et al.*, 1995]. In that study, subjects were supplied with transcripts of a doctor interviewing a patient. From the transcripts, it was possible to extract 20 respiratory disorders and a total of 304 "knowledge fragments" (e.g. identification of routine tests, non-routine tests, relevant parameters, or complaints). Subjects were also supplied with one of three problem solving methods representing models of the diagnostic domain. Each model began with the line "To help you with the task of editing the transcript, here is a model describing a way of classifying knowledge". Model one was an "epistemological model" that divided knowledge into various control levels of the diagnosis process. Model one was the "straw man"; it was such a vague description of how to do analysis that it should have proved useless. Model two was a KADS problem solving method for diagnosis. Model three was "no model"; i.e. no guidance was given to subjects as to how to structure their model. The results are shown in Figure 3. The statistical analysis performed by Corbridge *et. al.* found a significant difference between the performance of groups 3 compared to groups 1 and 2. Further, no significant difference could be found between the group using the poor problem solving method (model 1) and the group that using a very mature problem solving method (model 2). That is, sophisticated and mature descriptions of previously used knowledge (i.e. the KADS diagnosis description) were not found to be a productivity tool.

3.2 About Weak SC

While human beings have found it useful to use symbolic descriptions when co-ordinating their activities, it is not necessarily true that those symbolic descriptions are used internally by a single human in their own reasoning processes. Clancey [Clancey, 1987, Clancey, 1991] and Winograd & Flores [Winograd & Flores, 1987] argue that it is a mistake to confuse the symbolic descriptions which humans use to co-ordinate their activities and reflect about their actions (i.e. language) with how humans might generate their minute-to-minute behaviour. That is, we should not confuse our *conversations* about our thoughts with the actual *content* of those thoughts.

The Winograd & Flores line is that computers are not modeling tools *per se* but are really communication tools that facilitate the mediation of the exchange of ideas. Similarly, Clancey rejects the view that human inference is best replicated as matching/retrieving. Rather, says Clancey, these structures are created on-the-fly as posthoc symbolic justifications of a process which is not symbolic:

> The neural structures and processes that coordinate perception and action are created during activity, *not* retrieved and rotely applied, merely reconstructed, or calculated via stored rules and pattern descriptions [Clancey, 1993, p94].

Clancey's view is not resolved merely by declaring that knowledge representations are approximate surrogate models of reality (e.g. as proposed by [Davis *et al.*, 1993]). Rather, Clancey believes that symbolic structures are not only approximations of human knowledge but also that human knowledge changes as a result of applying it.

> Every action is an interpretation of the current situation, based on the entire history of our interactions. On some sense every action is automatically an inductive, *adjusted* process [Clancey, 1987, p238].

Researchers into decision support tools make a case something like **weak SC**. They argue that human "knowledge" appears in some social context and that context can effect the generated "knowledge". Phillips [Phillips, 1984] and Bradshaw *et. al.* [Bradshaw *et al.*, 1991] characterise model construction as a communal process that generates symbolic descriptions that explicate a community's understand of a problem. If the community changes then the explicit record of the communities shared understanding also changes; i.e. "truth" is socially constructed. Such an explicit expression of current beliefs may prompt further investigation and model revision; i.e. writing down models of "truths" can cause "truth" to change. Decision support tools are discussed later (§4.6).

Suchman [Suchman, 1987, Suchman, 1993, Agre, 1990] argues that real-world planning systems have to model their environment as well as their own goals. For example, a photocopier advisor system must...

> ...focus on the ways in which the photocopier and its user work together to maintain a shared understanding of what is going on between the tow of them and the copier... Far from executing a fully operational plan for effecting a fixed goal, the photocopier users continually reinterpreted their situation and based their various actions on their evolving interpretations [Agre, 1990].

If **weak SC** was false, then we should see that using knowledge does not change that knowledge; i.e. knowledge maintenance for static domains should terminate when it arrives at "truth". Compton [Compton *et al.*, 1989] reports studies that documented the changes made to models of biochemistry diagnosis systems. The Garvan ES-1, expert system was developed using a traditional iterative prototyping knowledge engineering methodology. Rules that began as simple modular chunks of knowledge evolved into very complicated and confusing knowledge (e.g. Figure 4). Note that this system was developed in a static domain; i.e. the system was a post-processor to a biochemical assay unit that did not change for the lifetime of the project. Despite this, the Garvan ES-1 expert system never reached a logical termination point, despite years of maintenance. There was always one more major insight into the domain, one more major conceptual error, and one more significant addition [Compton *et al.*, 1989]. A graph of the size of that knowledge base versus time (Figure 5) is consistent with *either* a linear growth curve or a logarithmic curve. Note that a linear curve would support the **SC premise**, while a logarithmic growth which would falsify **weak SC** since that would suggest that the KB is approaching "the truth". However, even if this was true for Garvan ES-1, note that the asymptote is very slow (see the *Logarithmic fit* of Figure 5). Even if we can approach "the truth", it seems it may take years to do so.

Garvan ES-1 was decommissioned before enough data could be collected to test if the growth curve was linear or logarithmic. Compton is monitoring the maintenance of PIERS [Preston *et al.*, 1993], a much larger system (which is version 2 of the above diagnosis system). A $y = x^{0.5}$ growth in KB size has been noted in that system. Significantly, the user-group sponsoring the project have created a permanent line item in their budget for maintenance. They anticipate that routinely every day, an expert will review the generated diagnoses and change some of the KB. That is, they believe that the model will never be finished/correct [Compton, 1994].

Experiments in machine learning endorse the proposition that *any* version of a model can be improved after more experience. Machine learning programs input training data to generate a model. Catlett's research [Catlett, 1991] explored the following area. Given a large amount of training data, is it necessary to use it all? That is, after a certain number of examples, is further experience (i.e. training data) superfluous? To test this, Catlett used C4.5 [Quinlan, 1986] to generate 20 decision trees for eleven machine learning problems using either (i) all the training cases or (ii) half the cases (randomly selected). Each generated tree was assessed using the test cases. In all cases, Catlett found that a statistically more accurate model could be generated using all the training data, rather than some randomly chosen subset (Figure 6). Note that while the theory learnt from N cases may be only marginally better than the theory learnt from $N/2$ cases (average reduction in error = 0.97%), the size of the better theory is 30% to 90% bigger (average increase in tree size = 53%); i.e. more examples prompted a significant reorganisation of the model (exception: the demon domain). That is, we may never know enough to create the correct model and that experience can significantly and continually modify old symbolic descriptions of knowledge.

A. Originally

```
RULE(22310.01)
IF  (bhthy or
     utsh_bhft4 or
     vhthy) and not on_t4
           and not surgery
           and (antithyroid or
                hyperthyroid)
THEN DIAGNOSIS("...thyrotoxicosis")
```

B. Same rule, 3 years later

```
RULE(22310.01)
IF  ((((T3 is missing)
       or (T3 is low and
           T3_BORD is low))
       and TSH is missing
       and vhthy
       and not (query_t4 or on_t4 or
                 or surgery or tumour
                 or antithyroid
                 or hypothyroid
                 or hyperthyroid))
      or ((((utsh_bhft4 or
            (Hythe and T4 is missing
             and TSH is missing))
            and (antithyroid or
                 hyperthyroid))
          or  utsh_bhft4
          or  ((Hythe or borthy)
                and T3 is missing
                and (TSH is undetect
                     or TSH is low)))
          and not on_t4 and not
              (tumour or surgery)))
     and (TT4 isnt low or T4U isnt low)
THEN DIAGNOSIS("...thyrotoxicosis")
```

Figure 4: A rule maintained for 3 years. From [Compton *et al.*, 1989].

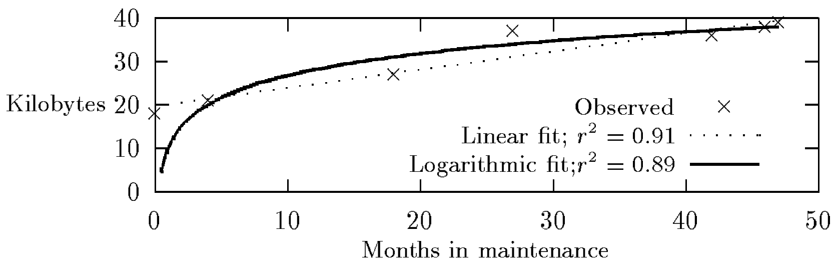

Figure 5: Garvan ES-1 knowledge base size

			%error rates				Tree size			
			Using N cases		Using $N/2$ cases		Using N cases		Using $N/2$ cases	
Domain	N training cases	test cases	mean	sd	mean	sd	mean	sd	mean	sd
demon	5000	2000	0.19	0.09	0.37	0.2	20.8	5.02	21.3	5.44
diff	5000	2000	1.58	0.31	2.27	0.38	87.7	6.13	59.8	6.10
othello	3000	2000	16.79	1.02	20.79	1.35	352.5	22.09	208.9	21.16
heart	3000	2000	2.89	0.62	4.39	0.54	74.2	9.61	45.9	6.88
sleep	3000	2000	25.24	0.65	27.62	0.85	234.7	20.79	135.2	17.69
hyper	5000	2000	1.08	0.19	1.29	0.26	26.1	4.83	15	4.304
hypo	5000	2000	0.6	0.19	0.7	0.2	27.7	4.69	19.1	4.02
binding	5000	2000	2.11	0.31	2.57	0.43	46.9	8.84	30.9	7.063
replace	5000	2000	1.41	0.34	1.76	0.34	30.7	7.66	22.1	6.27
euthy	5000	2000	0.56	0.18	0.91	0.25	37.4	3.7	28	4.42
wave	5000	2000	24.8	1.008	25.93	0.91	625	41.08	326.9	23.63

Figure 6: The Catlett Study

3.3 About Strong SC: An Over-Reaction?

§3.1 endorsed the **SC premise** and §3.2 endorsed **weak SC**. However, this endorsement does not necessarily imply an endorsement of **strong SC**. Like Vera & Simon in §4.1.3, this paper argues that symbolic systems are still a useful paradigm. It is not necessarily true what (e.g.) Birnbaum [Birnbaum, 1991] and McDermott [McDermott, 1987] argue; i.e. that the obvious alternative to logical AI is some type of procedural/functional semantics (i.e. **strong SC**). McDermott's motivation for a move away from symbols is based on his view that there has been "skimpy results so far and ...it is going to be very difficult to do much better in the future" [McDermott, 1987, p151][1]. Coming from McDermott, this is a telling criticism since, prior to this article, he was one of the leading proponents of that logical school [Charniak & McDermott, 1987].

With the benefit of a little historical hindsight, we can defeat some of McDermott's 1987 arguments. McDermott repeatedly uses Forbus' 1984 Qualitative Process Theory (QPT) [Forbus, 1984] as an worthy example of an algorithmic/non-logical system. McDermott originally demanded in 1984 that Forbus record the logical axioms underlying QPT. However, in 1987, McDermott comments that "...the task, seemingly so feasible (is) actually impossible" [McDermott, 1987, p152]. Note that, 12 years later, QPT was later implemented via a compilation into QSIM [Crawford *et al.*, 1992]. QSIM was a special-purpose theorem-prover built by Kuipers in 1986 for processing qualitative differential equations [Kuipers, 1986]. In 1993, Kuipers [Kuipers, 1993, p134] acknowledged is a textbook application of Mackworth's constraint-logic system [Mackworth, 1977, Mackworth, 1992]. That is, QPT was an instantiation of a logic-based system. However, when it was first developed, this was not known.

The lesson of the QPT story is that logical/symbolic descriptions could handle seemingly functional semantics. Descriptive symbolic systems that could be said to handle **weak SC** without requiring **strong SC** are described below (§4).

3.4 Experimental Evidence Against SC

There are two outstanding experimental studies that challenge the SC case that knowledge is context dependent. Runkel reports large amounts of verbatim reuse using a toolkit of problem solving strategies that separated search control from other domain knowledge [Runkel, 1995]. Marques *et. al.* report significantly reduced development times for expert systems using the 13 mechanisms in the SBF toolkit (eliminate, schedule, present, monitor, transform-01, transform-02, compare-01, compare-02, translate-01, translate-02, classify, select, dialog-mgr). In the nine applications studied by Marques *et. al.*, development times changed from one to 17 days (using SBF) to 63 to 250 days (without using SBF) [Marques *et al.*, 1992]. To the best of our knowledge, these two studies represent the current high-water marks in software reuse in both the

[1] Possibly, Vera & Simon would find McDermott's claim here as "preposterous" as the similar claim of Agre (§4.1.3)

conventional software engineering and knowledge engineering paradigms[2].

Nevertheless, neither experiment is a convincing demonstration of general knowledge reuse. The Runkel experiment had poor control of the *resources used* while the SBF experiment had poor control of the *product produced*. All the applications studied in the SBF work were variations of the theme of computer configuration; i.e. the Marques studied restricted the context of the test. In the Runkel study, it is not clear who built the studied applications. If it was Runkel himself, then it is possible that the author's internal (possibly non-symbolic) model informed the usage of the tools. Also, Runkel does not report the time required to use his approach; i.e. the Runkel study cannot be used to depend the proposition that reusing old symbolic descriptions is a productivity tool for new applications.

4 Responses to Situated Cognition

One response to SC is to argue that current techniques work, so why do we need to change them? This paper discounts this argument (§4.1) and move on to techniques that address **weak SC**. **Weak SC** suggests that generating a explicit record of a specification is less of an issue than changing that specification over time. Therefore, any technique which can support specification change is a potential response to SC; e.g. verification & validation tools (§4.2), repertory grids (§4.3), expert critiquing systems (§4.4), machine learning (§4.5), certain frameworks for decision support systems (§4.6), and ripple-down-rules (§4.7). Note that:

- Most of these techniques come from authors who do not write about SC (exception: ripple-down-rules). However, all this work emphasises that the ability to assess and alter a model is as important or more important than the ability to initially specify it.
- These techniques are only *potentially* appropriate responses to SC. A method will be described below to assess these techniques (§5.2).

4.1 Response: Ignore it

4.1.1 Not a KA Problem

Clancey cautions [Clancey, 1989a] that we should not confuse pragmatic discussions about techniques for knowledge acquisition (e.g. [Clancey, 1985, Clancey, 1989b, Clancey, 1992]) with discussions about the basic nature of human intelligence (e.g. [Clancey, 1987, Clancey, 1991, Clancey, 1993]). Clancey prefers to reserve discussions on SC for the creation of human-equivalent robots which react to real-world situations since, he says,

> SC argues that human activity is not strictly mediated by inference over descriptions nor is activity a compiled result of such inference [Clancey, 1996].

Clancey's remarks notwithstanding, this paper argues that SC has a significant impact on KA. If **weak SC** is true, then we cannot expect to reuse old symbolic descriptions of ontologies or PSMs as a productivity tool for some current application. Instead of focusing on reusing old knowledge, KA SC-style should focus on how we build and change models. That is, expertise is not a function of using a large library of old knowledge as argued in [Larkin *et al.*, 1980] and favoured by the $\mathcal{KL}_B$ approach. Rather, expertise is the ability to quickly adapt old models to new situations.

A KA methodology that acknowledges SC must offer details about creating and changing a knowledge base. A review of the KA literature suggests that most of the effort is in knowledge analysis and not knowledge maintenance (exceptions: [de Brug *et al.*, 1986, Compton *et al.*, 1989]). Current KA practice has not acknowledged SC since, if it did, there would be more work in knowledge maintenance.

[2] Elsewhere [Menzies & Haynes, 1994], certain inflated claims of reuse [Stark, 1993] from the object-oriented community have been criticised.

4.1.2 Evidence for SC Not Conclusive

It could be argued that the evidence for **weak SC** is not convincing. For example, a reviewer of this paper wrote:

> Discussion of the pilot study done by Corbridge (§3.1) involves results are too premature to bolster the SC claim. Discussion of one PSMs in SBF and KADS (§3.4) also seems less than convincing evidence for the SC claim.

This comment is true: the evidence above is not convincing. However, at the very least, the above evidence is suggestive that we need to make a careful review of the perceived successes of current KA approaches. Given the current lack of good experimental evidence demonstrating the utility of $\mathcal{KL}_B$ (§3.4), we need to do more experiments (§5.2).

4.1.3 "In Practice, SC is not a Problem"

Proponents of $\mathcal{KL}_B$ may argue that they have no case to answer. By some measures, $\mathcal{KL}_B$ is a successful paradigm. For example, Wielinga *et. al.* report that, as of 1992, KADS has been used in some 40-to 50 KBS projects, 17 of which are described in published papers [Wielinga *et al.*, 1992]. Further, if the situation is as bad as suggested above, then how is it that we have so many seemingly successful expert systems (e.g. MYCIN [Yu *et al.*, 1979], CASNET [Weiss *et al.*, 1978], PROSPECTOR [Campbell *et al.*, 1982, Duda *et al.*, 1985], XCON [Bachant & McDermott, 1984], VT [Marcus *et al.*, 1987], PIGE [Menzies *et al.*, 1992])?

This kind of argument is the basis of Vera & Simon's criticisms of SC [Vera & Simon, 1993b, Vera & Simon, 1993a, Vera & Simon, 1993c]. They describe as "preposterous" [Vera & Simon, 1993a, p95] a claim by Agre that "nobody has described a system capable of intelligent action at all- and that nothing of the sort is going to happen soon" [Agre, 1993, p69]. We suspect that they would also object to McDermott's lament about "skimpy results so far" (§3.3). Vera & Simon argue that the physical symbol system hypothesis (PSSH) [Newell & Simon, 1972] has been a fruitful paradigm which can reproduce many known behaviours of experts. They decline to reject PSSH for **strong SC** since, if they adopted, e.g. Clancey's situated paradigm [Clancey, 1993], then they are unclear on what predictions can be made and what experiments can be performed. That is, they argue that SC is unfalsifiable and unscientific[3].

Nevertheless, it would be a mistake for proponents of $\mathcal{KL}_B$ to use the Vera & Simon arguments as support for their reuse paradigm. Vera & Simon are only arguing against **strong SC** (which they call "situated action"). A symbolic system that can implement **weak SC** would still satisfy Vera & Simon's broad definition of a symbolic system while challenging the $\mathcal{KL}_B$ paradigm.

Further, just because a system based on explicit symbolic descriptions works, this says nothing about the best way to *build* and *maintain* those symbolic descriptions. Clancey acknowledges the role of symbolic descriptions in working systems [Clancey, 1987, p278] [Clancey, 1992]. Symbolic descriptions, Clancey argues, are useful to planning about the future and reflecting on action rather than immediately reacting to a new situation.

> Human reasoning is immensely more successful by our ability to simulate what might happen, to visualize possible outcomes and prepare for them. We do this by reflecting, saying what we expect, and responding to what we say [Clancey, 1987, p247].

However, Clancey's symbolic descriptions are not as fixed as those in Ontolingua [Gruber, 1993] or the inference layer of KADS.

> It remains to explain how (the symbolic descriptions) *develop*.... Most learning programs grammatically describe how representations accumulate within a fixed language. They don't explain how representations are created, or more generally, the evolution of new routines not described by the given grammar [Clancey, 1991, p279].

Knowledge acquisition is the key point that is ignored by Vera & Simon. They comment on the successes of *working* symbolic descriptions of human knowledge, not on the effort involved in *constructing* those

[3] While this paper agrees that the PSSH is still a useful paradigm (hence the rejection of **strong SC**), it is not necessarily true that SC research is not scientific. Subsequently, a empirical experiment for evaluating SC will be defined (§5.2)

descriptions. Despite careful attempts to generalise principles of knowledge acquisition, (e.g. [Stefik *et al.*, 1982]), expert systems construction remained a somewhat hit-and-miss process. By the end of the 1980s, it was recognised that our design concepts for knowledge-based systems were incomplete [Buchanan & Smith, 1989]. For example, Steels [Steels, 1994] cites an example where an expert could not solve a problem over the phone but, as soon as they walked into the room where the trouble was, could solve it instantly. Examples like this encourage the knowledge-relativists within KA to argue that we have under-valued the role of context in the creation of our symbolic descriptions.

4.2 Response: Use Verification & Validation Tools

Weak SC suggests that a specification considered correct at time T_1 may become potentially incorrect at T_2. It has been argued previously [Menzies & Compton, 1995] that such potentially inaccurate models must be tested, lest they generate inappropriate output for certain circumstances. Testing can only demonstrate the presence of bugs (never their absence) and so must be repeated whenever new data is available. That is, testing is an essential, on-going process through-out the lifetime of a knowledge base.

Preece and Zlatereva describe test programs based on the logical structure of rule-based expert systems. Preece's verification tools detect anomalies in those structures [Preece, 1992] while Zlatereva's validation tools analyse that structure to generate a test suite which will exercise all parts of the rule-base [Zlatereva & Preece, 1994]. Verification tools search for syntactic anomalies within a knowledge base such as tautologies, redundancies, and circularities in the dependency graph of literals in a knowledge base [Preece & Shinghal, 1992]. Many of Preece's verification tools can be mapped into a graph-theoretic analysis of the dependency graph $\mathcal{D}$ of literals in a KB used in HT4 (e.g. Figure 7.A). For example, a test for "unreachable conclusions" can be converted into the following graph-theoretic process. Compute the components (separate sub-graphs) of $\mathcal{D}$. If a component contains conclusions but no system inputs, then those conclusions are unreachable. Also, a test for "circularities" can be converted into a computation of the transitive closure of $\mathcal{D}$. "Looping" means finding a literal in its own transitive closure. Verification is not a definitive test for a KBS. Preece reports example where working expert systems contained syntactic anomalies, yet still performed adequately (recall Figure 1).

Validation tools assess knowledge via some external semantic criteria; e.g. testing that a knowledge base model of X can reproduce known behaviour of X. If such a test suite of behaviour is missing, then non-monotonic reasoning techniques can be used to explore the dependency graph between KB literals to find sets of input literals which will exercise the entire knowledge [Ginsberg, 1990, Zlatereva, 1992]. However, an expert still has to decide what output is appropriate for each generated input. This can introduce a circularity in the testing procedure. After an expert describes their world-view in a model, that same expert will be a asked to specify the results of certain inputs. If the expert then uses the same model to predict the output, then they would be using a potentially faulty model to generate a potentially faulty prediction about the output.

Our preferred validation approach is for the input-output test pairs to be generated totally separately to the current model; e.g. from real-world observations of the entity being modeled in the KB. Based on work by Feldman & Compton [Feldman *et al.*, 1989], a general validation framework based on the HT4 abductive inference engine has been developed. Elsewhere [Menzies, 1996a], we have given an overview of abductive research [O'Rourke, 1990, Bylander *et al.*, 1991, Eshghi, 1993, Selman & Levesque, 1990, Poole, 1990b, Menzies, 1996a]. Here, we offer an approximate characterisation of abduction as the search for consistent subsets of some background theory that are relevant for achieving some goal. If multiple such subsets can be generated, then a $\mathcal{BEST}$ assessment operator selects the preferred world(s). For example, suppose HT4 wants to validate that certain $\mathcal{OUT}$put goals can be reached from using the $\mathcal{IN}$puts shown in the dependency graph $\mathcal{D}$ of Figure 7.A. In that Figure, `x` $\stackrel{++}{\rightarrow}$ `y` denotes that `y` being `up` or `down` can be explained by `x` being `up` or `down` respectively and `x` $\stackrel{--}{\rightarrow}$ `y` denotes that `y` being `up` or `down` could be explained by `x` being `down` or `up` respectively. HT4 can find the following proofs connecting $\mathcal{OUT}$s to $\mathcal{IN}$s: $\mathcal{P}_1$=`aUp` $\rightarrow$ `xUp` $\rightarrow$ `yUp` $\rightarrow$ `dUp`, $\mathcal{P}_2$= `aUp` $\rightarrow$ `cUp` $\rightarrow$ `gUp` $\rightarrow$ `dUp`, $\mathcal{P}_3$= `aUp` $\rightarrow$ `cUp` $\rightarrow$ `gUp` $\rightarrow$ `eUp`, $\mathcal{P}_4$= `bUp` $\rightarrow$ `cDown` $\rightarrow$ `gDown` $\rightarrow$ `fDown`, $\mathcal{P}_5$= `bUp` $\rightarrow$ `fDown`.

These proofs may contain assumptions, i.e. literals that are not known $\mathcal{FACTS}$. Continuing the example of Figure 7.A, if $\mathcal{FACTS}$=$\mathcal{IN} \cup \mathcal{OUT}$, then {`xUp,yUp,cUp, cDown, gUp, gDown`} are assumptions. If we can't believe that a variable can go `up` and `down` simultaneously, then we can declare {`cUp, cDown, gUp, gDown`} to be conflicting (denoted $\mathcal{A}_c$). Figure 7.A shows us that `g` is fully dependent on `c`. Hence the key

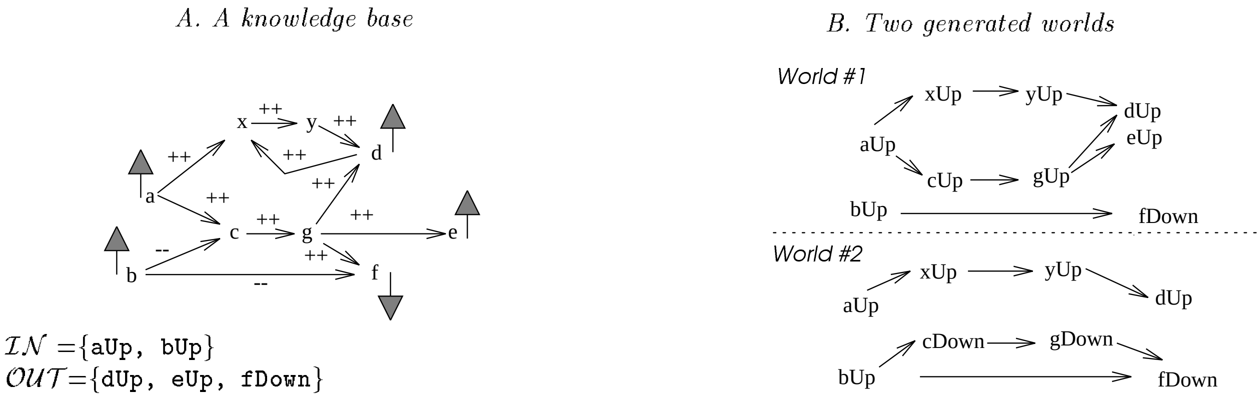

Figure 7: Worlds generation

conflicting assumptions are {`cUp, cDown`} (denoted *base controversial assumptions* or $\mathcal{A}_b$). We can used $\mathcal{A}_b$ to find consistent belief sets called worlds $\mathcal{W}$. A proof $\mathcal{P}_i$ is in $\mathcal{W}_j$ if that proof does not conflict with the environment $\mathcal{ENV}_j$ (an environment is a maximal consistent subset of $\mathcal{A}_b$). In our example, $\mathcal{ENV}_1$={`cUp`} and $\mathcal{ENV}_2$={`cDown`}. Hence, $\mathcal{W}_1=\{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_5\}$ and $\mathcal{W}_2=\{\mathcal{P}_1\ \mathcal{P}_4\ \mathcal{P}_5\}$ (see Figure 7.B)[4].

Abductive validation is simply the application of the above algorithm with a $\mathcal{BEST}$ assessment operator that returns the world(s) with the maximum *cover*; i.e. overlap with $\mathcal{OUT}$. The overlap of $\mathcal{W}_1$ and $\mathcal{OUT}$ is {`dUp,eUp,fDown`} and the overlap $\mathcal{W}_2$ and $\mathcal{OUT}$ is {`dUp,fDown`}; i.e. $\mathcal{W}_1^{cover} = 3 = 100\%$ and $\mathcal{W}_2^{cover} = 2 = 67\%$. The maximum cover is 100%; i.e. their exist a set of assumptions ({`cUp`}) which let us explain all of $\mathcal{OUT}$ and this theory has passed HT4-style validation. Note that this procedure corresponds to answering the following question: "how much of the known behaviour of X can be reproduced by out model of X?".

This abductive validation was applied to Smythe '89, a model of glucose regulation published in an international, refereed journal [Smythe, 1989]. Using an earlier version of HT4 (which they called QMOD and we call HT1) Feldman & Compton reported that only 69% of the known observations could be explained by Smythe '89n [Feldman *et al.*, 1989]. In our re-work of that study, and-vertex processing and multiple causes processing was added, thus allowing the processing of more of the known observations. With those changes, HT4 found that only 55% of the observations were explicable [Menzies, 1996b]. When these errors were shown to Smythe, he found them novel and exciting [Feldman *et al.*, 1989]; i.e. the domain expert found that these errors were significant. This is both a disturbing and exciting finding. It is disturbing in the sense that if the very first large-scale medical theory analysed by HT4 contains significant numbers of errors, then it raises doubts as to the accuracy of theories in general (a result which would be consistent with the **SC premise**).

4.3 Response: Use Repertory Grids

Gaines & Shaw explore techniques for resolving conflicts in terminology. The conceptual systems of different experts are explicated and compared using a technique called entity-attribute grid elicitation [Gaines & Shaw, 1989]. Experts are asked to identify dimensions along which items from the domain can be distinguished. The two extreme ends of these dimensions are recorded left and right of a grid. New items from the domain are categorised along these dimensions. This may elicit new dimensions of comparisons from the expert which will cause the grid to grow (see [Shaw & Gaines, 1992] for a sample of such grids). Once the dimensions stabilize, and a representative sample of items from the domain have been categorised, then the major distinctions and terminology of a domain has been defined. Differences between the conceptual views of different experts can be identified (e.g. their categorisations are different). Gaines & Shaw describe automatic tools for generating plots representing the proximity of different expert's conceptual systems [Gaines & Shaw, 1989].

Gaines & Shaw focuses on identifying and resolving conflicts in the meaning of individual terms, not on conflicts in the semantics of the models built using combinations of those terms. A model-level conflict detection facility such as abductive validation requires knowledge of how terms are combined.

[4] The connection of HT4 to DeKleer's ATMS system [DeKleer, 1986] is explored elsewhere [Menzies, 1996a]

4.4 Response: Expert Critiquing Systems

Silverman [Silverman, 1990, Silverman, 1992] advises that attached to an expert system should be 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 [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. Dialogue generators are very domain-specific. Silverman's research seems to be aimed at an implementation-independent analysis of the process of "critiquing" a program; i.e. "critiquing" as an add-on to existing systems, not as a built-in that is fundamental to the whole KBs life cycle. While this approach is useful, a more extensible approach would to change the structure of knowledge- bases systems such that critiquing is built into the system (see §5.1). In the case where the design of the system can be altered to integrate a testing module, the abductive approach of HT4 is an alternative approach to critiquing. Silverman's "deep models" are the theory that generates the proofs (e.g. Figure 7.A) while the difference analyser is the $\mathcal{BEST}$ assessment operator which reports what behaviours can't be covered.

4.5 Response: Use Machine Learning

Validation and verification techniques can only automatically *find* faults. Machine learning (ML) techniques can fully or partially automate the creation or the *fixing* of a specification. Given some input facts, some goals, and some prior knowledge, then ML can use induction, analogy, deduction, or heuristic techniques to generate a revision to the prior knowledge [Michalski, 1993].

If numerous examples are available (say, hundreds to thousands), then empirical inductive techniques such as mathematical regression, genetic algorithms, neural nets or other techniques (e.g. nearest-neighbor algorithms, decision-tree generation, simple Bayesian reasoning [Kohavi *et al.*, 1996]) can propose a new theory. These techniques have not come to replace standard knowledge acquisition for several reasons. Firstly, most naturally-occurring domains are data-poor. Automatic empirical inductive generalisation in such data-poor domains is an unreliable technique. Secondly, once a theory is revised, the revisions must be acceptable to a human reader. Empirical inductive generalisation techniques such as neural nets, genetic algorithms, or decision tree learners may generate a revision of prior knowledge that is too big or too awkward to read. Further, most empirical inductive generalisation machine learning algorithms (e.g. the C4.5 decision tree of Figure 6) make no attempt to preserve current beliefs (exception: inductive logic programming [Muggleton, 1991]). It may be unacceptable to permit a learning algorithm to scribble all over a knowledge base, particularly those portions which the user has some commitment to.

In domains that lack sufficient input facts for empirical inductive generalisation, then deductive ML algorithms exist which (i) build an explanatory structure across the current knowledge base then (ii) edit this structure to generate a refinement to the knowledge base. This can be done using automatic tools (e.g. explanation-based generalisation [van Harmelen & Bundy, 1988]) or semi-automatic tools where the user's opinions are used as part of the theory refinement loop. Heuristic KB refinement (e.g. KRUST [Craw *et al.*, 1994, Palmer & Craw, 1995] and expert critiquing systems (§4.4)) are a kind of "machine learning" algorithm in which domain-specific principles are used to fault a KB and assist a human in fixing the faults.

Note that the new theory learnt by deductive ML algorithms can only ever be a subset of the prior knowledge over which explanatory structures can be build. For example, the worlds of Figure 7.B represent the consistent explanation structures we can generate from Figure 7.A. Each such set is just a subset of the edges in the dependency graph between the literals of Figure 7.A. If we cached these worlds, then we could say we have "learnt" that in the case of Figure 7.A, there are two possibilities depending on the value of `c`.

In domains that lack both sufficient input facts for empirical inductive generalisation and prior knowledge, then the only other way to build a theory is to ask an expert; i.e. standard knowledge acquisition. In decision support systems (§4.6), for example, developers are not recording a model of an existing domain. Rather, they are using software tools to build a model for a newly, poorly understood domain which has not been previously documented.

4.6 Response: Treat KA as Decision Support Systems

Workers in decision support systems deliberately try to model the context of the decision making process. DSS theory believes that management decision making is not inhibited by a lack of information. Rather, it is confused by an excess of irrelevant information [Ackoff, 1967]. Modern decision-support systems (DSS) aim to filter useless information to deliver relevant information (a subset of all information) to the manager. Simon originally characterised decision making as a three stage process: intelligence (scanning environment), design (develop alternative courses), and choice (selection of alternative) [Simon, 1960]. Our preferred definition of a decision-support system is based on Brookes [Brookes, 1986]' who developed it from Simon's and Mintzberg's model [Mintzberg, 1975]. The goal of a DSS is *management comfort*, i.e. a subjective impression that all problems are known and under control. More specifically, managers need to seek out problems, solve them, then install some monitoring routine to check that the fix works. A taxonomy of tasks used in that process is shown in Figure 8.

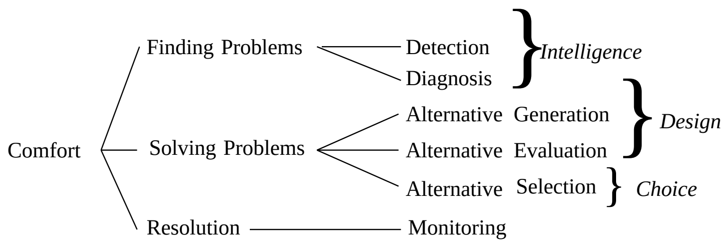

Figure 8: Components of management comfort

Other DSS workers have a similar view. Boose, Bradshaw, Koszaek, and Shema (BBKS) [Boose *et al.*, 1992] discuss DSS architectures suitable for groups. The BBKS system lacks an execution module for the generated models as part of the group decision support environment. Boose *et. al.* assume that once the group's mode is elicited, it will be subsequently exported into an executable form. Portions of the BBKS and the Brookes' models overlap. The BBKS system lets groups manipulate their group model, its inter-relationships, and the group's criteria for selecting the best alternative. BBKS stress that:

> The process of generating and scoring alternatives are at the heart of most decision processes. [Boose *et al.*, 1992]

That is, more important than representing and executing a model is an ability to assess a model. Note that the $\mathcal{BEST}$ operator of HT4 directly implements this alternative generation, assessment, and selection procedure. Further, abduction can be used for other DSS tasks such as diagnosis [Console & Torasso, 1991] and monitoring[5].

4.7 Response: Ripple-Down-Rules

$\mathcal{KL}_B$ (§2) typically assumes that prior to building a system, an extensive analysis stage develops a design for the system. Compton reports experiments with a completely reversed approach. In ripple-down-rules (RDR), there is no analysis period. Starting with the single rule "if true then no classification", KA in an RDR system consists only of fixing faulty rules using an *unless* patch attached at the end of a rule condition. Patches are themselves rules which can be recursively patched. Experts can never re-organise the tree; they can only continue to patch their patches. If a new *case* motivates a new patch, that this case is stored with the new patch. Compton argues that these (RDR) trees models the context of knowledge acquisition. When a case is processed by an RDR tree, its context is the set of cases in the patches exercised by the new case. When looking for new patches, experts can only choose from the *difference* of the attributes in the current case and the attributes exercised down to the current faulty rule.

[5] Given N worlds representing possibilities, monitor incoming data. Reject the world(s) whose assumptions conflict with the new data. The remaining worlds are the current possibilities.

RDR trees are a very low-level representation. Rules cannot assert facts that other rules can use. In no way can a RDR tree be called a model in anything like a $\mathcal{KL}_B$ sense. Yet this low-level model-less approach has produced large working expert systems in routine daily use. For example, the PIERS system at St. Vincent's Hospital, Sydney, models 20% of human biochemistry sufficiently well to make diagnoses that are 99% accurate [Preston *et al.*, 1993]. RDR has succeeded in domains where previous attempts, based on much higher-level constructs, never made it out of the prototype stage [Patil *et al.*, 1981]. Further, while large expert systems are notoriously hard to maintain [de Brug *et al.*, 1986], the no-model approach of RDR has never encountered maintenance problems. System development blends seamlessly with system maintenance since the only activity that the RDR interface permits is patching faulty rules in the context of the last error. For a 2000-rule RDR system, maintenance is very simple (a total of a few minutes each day). Compton argues that his process of "patching in the context of error" is a more realistic KA approach than assuming that a human analyst will behave in a perfectly rational way to create some initial correct design [Compton & Jansen, 1990].

5 Assessing Responses to Situated Cognition

Weak SC suggests that, as far as possible, the symbolic structures inside an expert system must be changeable. Any representational system assumes certain primitives which can't be changed. $\mathcal{KL}_B$ assumes that a PSM does not change over the lifetime of a project. Hence, our preferred response to SC is a PSM-maintenance environment called ripple-down-rationality, or RD-RA.

RD-RA is described below (§5.1). This description is only a preliminary sketch since it is new work-in-progress. We present it here in order to motivate our SC experiment (§5.2). If the reader disagrees with our proposal, we invite them to consider how they would assess the success or failure of RD-RA.

5.1 RD-RA: Ripple-Down-Rationality

For reasons of generality, we base RD-RA around the HT4 abductive validation engine. Note that many common knowledge representations can be mapped into dependency graphs like Figure 7.A. For example, horn clauses can be viewed as a graph where the conjunction of sub-goals leads to the head goal. In the special (but common) case where the range of all variables is known (e.g. propositional rule bases), this graph can be converted into a ground form where each vertex is a literal. Invariants may be added to represent sets of literals that are mutually exclusive (e.g. `cUp` and `cDown`). Such graphs are commonly computed for the purposes of optimisation or verification (§4.2).

An interesting feature of abduction is that it both a validation and an inference engine. It maps exactly into Clancey's characterisation of expert systems as devices that build a system-specific model (SSM) or Breuker's component's of solutions [Breuker, 1994, Clancey, 1992]. As evidence of this, we can express the details of a wide range of KBS tasks in this abductive framework; e.g. intelligent decision support systems (§4.6 & [Menzies, 1995a]), diagrammatic reasoning [Menzies & Compton, 1994], single-user knowledge acquisition, and multiple-expert knowledge acquisition [Menzies, 1995c], certain interesting features of human cognition [Menzies, 1995d]. natural-language processing [Ng & Mooney, 1990], design [Poole, 1990a], visual pattern recognition [Poole, 1990c], analogical reasoning [Falkenhainer, 1990], financial reasoning [Hamscher, 1990], machine learning [Hirata, 1994], case-based reasoning [Leake, 1993], expert critiquing systems (§4.4), prediction, classification, explanation, tutoring, qualitative reasoning, planning, monitoring, set-covering diagnosis, consistency-based diagnosis, verification and validation (§4.2 & [Menzies, 1996a, Menzies, 1996b]). Further, abduction handles certain hard and interesting cases; such as the processing of indeterminate, under-specified, globally inconsistent, poorly measured theories. Inferencing over such theories implies making assumptions and handling mutually exclusive assumptions in different worlds.

RD-RA is a combination of RDR with HT4 and a graph-theoretic version of SOAR. Recall Newell's principle of rationality (§2): part of intelligence is the appropriate selection of operators to decide between possible inferences. HT4's $\mathcal{BEST}$ operator's can be characterised by how much information they require to execute. A *node-level* operator could assess the utility of using some edge based on some numeric weight on that edge. A *proof-level* operator could assess the utility of using some edge based on its contribution to a growing proof and how that growing proof compares to other known proofs (e.g. like in beam search). A *worlds-level* operator could assess the utility of a world based on various criteria (e.g. the validation $\mathcal{BEST}$

in §4.2).

In RD-RA, a PSM is implemented as sets of $\mathcal{BEST}$ operators. If we use RDR rules to control the modifications of the different levels of $\mathcal{BEST}$ operators then, potentially, we a maintenance environments for PSMs. Recall that each $\mathcal{BEST}$ operator is a small procedure that classifies each proposed inference (edge, world) as "acceptable" or "cull". A little RDR KB could maintain each $\mathcal{BEST}$ operator.

5.2 The Assessment Experiment

Consider a KB comprising a state space connecting literals and a PSM which controls the traversal of that state space. **Weak SC** has no impact of KA and RD-RA is unnecessary if, in the usual case, the edges and the PSM do not change radically after some initial analysis period.

Let us characterise two opposing KA processes:

1. An *analysis intensive process* (AIP) where most of the knowledge base is found via an analysis that precedes building the executing system; e.g. KADS [Wielinga *et al.*, 1992]. In an AIP system built in the HT4 framework, most of the edges of the theory and all of the PSM would be added before the system has processed $\mathcal{IN}$put, $\mathcal{OUT}$put pairs.

2. A *maintenance intensive process* (MIP) where most of the knowledge base or the structure of the PSMs are found via fixing the runtime system once an error is found; e.g. RDR (§4.7) or RD-RA (§5.1). In an MIP system built in the RD-RA framework, most of the edges of the theory and much of the RDR trees controlling the $\mathcal{BEST}$ operators would not be added till after the system has processed $\mathcal{IN}$put, $\mathcal{OUT}$put pairs (recall that in RD-RA, a PSM is implemented as sets of $\mathcal{BEST}$ operators).

Note that if we add modification dates to all edges, PSMs, and $\mathcal{IN}$put, $\mathcal{OUT}$put pairs then we can auto-detect if a software artifact is generated via MIP or AIP.

Both MIP and AIP contribute edges to a knowledge base dependency graph. This knowledge base is subsequently evaluated via our validation process (§4.2). We would declare MIP or AIP to be *satisfactory* if it can generate competent systems from the same specification (e.g. one of the Sisyphus projects [Linster, 1992]). Note that we can assess competence via the HT4 abductive validation algorithm.

Our validation process also lets us identify *good* edges (§4.2). Further, we can declare a $\mathcal{BEST}$ operator to be *good* if it was exercised in the generation of *good* edges. We would declare either the MIP or AIP process *superior* if it produced more good edges/operators sooner than the other.

We would declare **weak SC** irrelevant to the practice of KA if curves of "number of good edges changed" vs "months in production" or "revisions to good operators" vs "months in production" (e.g. a variant on Figure 5) flatten out very quickly. We can also falsify the **SC premise** if AIP KA is *superior* to MIP KA when both are applied to the same application.

Note that the above process has to be repeated a number of times over similar applications developed by skilled programmers who know their tools; i.e. such as in the Sisyphus experiments.

6 Conclusion

Existing KA methods, including $\mathcal{KL}_B$, can deliver successful applications (§4.1.3). However, which is the best method? The crucial test for a particular KA technique is not whether or not it can deliver applications. Rather, it should be "can method X deliver applications better than the alternatives?".

In this paper two alternative approaches to KA have been characterised: (i) analysis-intensive processing (the dominant view) and (ii) maintenance intensive processing (a minority view). It has been argued that there is enough evidence for **weak SC** to motivate a review of analysis intensive processing. Current experimental evidence is not sufficient to inform such a review. Hence, several *potential* responses to the challenge of SC have been discussed and a experiment that could determine the impact (if any) of *weak SC* on KA has been proposed.

In order for this test to be fair to the current $\mathcal{KL}_B$ paradigm, we will need to build a set of translators that convert the constructs found in current KBS and information systems methodologies down into structures like the HT4 graphs. The implementation of a library of such translators is our current TROJAN project.

7 Acknowledgments

This paper has benefited from the careful comments of Bill Clancey, Paul Compton, John O'Neill, and the anonymous referees.

References

[Ackoff, 1967] Ackoff, R. (1967). Management Misinformation Systems. *Management Science*, pages 319–331.

[Agnew *et al.*, 1993] Agnew, N., Ford, K., & Hayes, P. (1993). Expertise in Context: Personally Constructed, Socially elected, and Reality-Relevant? *International Journal of Expert Systems*, 7.

[Agre, 1990] Agre, P. (1990). Book Review: Lcy A. Scuhman, Plans and Situated Actions: The Problems of Human-Machine Communication. *Artificial Intelligence*, 43:369–384.

[Agre, 1993] Agre, P. (1993). The Symbolic WorldView: Reply to Vera and Simon. *Cognitive Science*, 17:61–69.

[Bachant & McDermott, 1984] Bachant, J. & McDermott, J. (1984). R1 Revisited: Four Years in the Trenches. *AI Magazine*, pages 21–32.

[Benjamins, 1995] Benjamins, V. (1995). Problem-Solving Methods for Diagnosis and their Role in Knowledge Acquisition. *International Journal of Expert Systems: Research & Applications*, 8(2):93–120.

[Birnbaum, 1991] Birnbaum, L. (1991). Rigor Mortis: A Response to Nilsson's 'Logic and Artificial Intelligence'. *Artificial Intelligence*, 47:57–77.

[Boose *et al.*, 1992] Boose, J., Bradshaw, J., Koszareck, J., & Shema, D. (1992). Knowledge Acquisition Techniques for Group Decision Support. In Gaines, B., Musen, M., & Boose, J., (Eds.), *Proceedings of the 7th Knowledge Acquisition for Knowledge-Based Systems Workshop*, pages 2.1–2.22.

[Bradshaw *et al.*, 1991] Bradshaw, J., Ford, K., & Adams-Webber, J. (1991). Knowledge Representation of Knowledge Acquisition: A Three-Schemata Approach. In *6th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, ,October 6-11 1991, Banff, Canada*, pages 4.1 – 4.25.

[Bredeweg, 1992] Bredeweg, B. (1992). *Expertise in Qualitative Prediction of Behaviour*. PhD thesis, University of Amsterdam.

[Breuker, 1994] Breuker, J. (1994). Components of Problem Solving and Types of Problems. In *8th European Knowledge Acquisition Workshop, EKAW '94*, pages 118–136.

[Brookes, 1986] Brookes, C. (1986). Requirements Elicitation for Knowledge Based Decision Support Systems. Technical Report 11, Information Systems, University of New South Wales.

[Buchanan & Smith, 1989] Buchanan, B. & Smith, R. (1989). Fundamentals of Expert Systems. In A. Barr, P. C. & Feigenbaum, E., (Eds.), *The Handbook of Artificial Intelligence, Volume 4*, volume 4, pages 149–192. Addison-Wesley.

[Bylander *et al.*, 1991] Bylander, T., Allemang, D., M.C. Tanner, M., & Josephson, J. (1991). The Computational Complexity of Abduction. *Artificial Intelligence*, 49:25–60.

[Campbell *et al.*, 1982] Campbell, A., Hollister, V., Duda, R., & Hart, P. (1982). Recognition of a Hidden Material Deposit by and Artificially Intelligent Program. *Science*, 217:927–929.

[Catlett, 1991] Catlett, J. (1991). Inductive learning from subsets or Disposal of excess training data considered harmful. In *Australian Workshop on Knowledge Acqusition for Knowledge-Based Systems, Pokolbin*, pages 53–67.

[Charniak & McDermott, 1987] Charniak, E. & McDermott, D. (1987). *Introduction to Artificial Intelligence*. Addison-Wesley.

[Clancey, 1985] Clancey, W. (1985). Heuristic Classification. *Artificial Intelligence*, 27:289–350.

[Clancey, 1987] Clancey, W. (1987). Book Review of Winograd & Flores, Understanding Computers and Cognition: A New Foundation for Design. *Artificial Intelligence*, 31:233–250.

[Clancey, 1989a] Clancey, W. (1989a). The knowledge level reinterpreted: Modeling how systems interact. *Machine Learning*, 4(3/4):285–293.

[Clancey, 1989b] Clancey, W. (1989b). Viewing Knowledge Bases as Qualitative Models. *IEEE Expert*, pages 9–23.

[Clancey, 1991] Clancey, W. (1991). Book Review of Israel Rosenfield, The Invention of Memory: A New View of the Brain. *Artificial Intelligence*, 50:241–284.

[Clancey, 1992] Clancey, W. (1992). Model Construction Operators. *Artificial Intelligence*, 53:1–115.

[Clancey, 1993] Clancey, W. (1993). Situated Action: A Neuropsychological Interpretation (Response to Vera and Simon). *Cognitive Science*, 17:87–116.

[Clancey, 1996] Clancey, W. (1996). Personal communcaition.

[Compton, 1994] Compton, P. (1994). Personal communication. regarding the status of the PIERS system.

[Compton *et al.*, 1989] Compton, P., Horn, K., Quinlan, J., & Lazarus, L. (1989). Maintaining an Expert System. In Quinlan, J., (Ed.), *Applications of Expert Systems*, pages 366–385. Addison Wesley.

[Compton & Jansen, 1990] Compton, P. & Jansen, R. (1990). A Philosophical Basis for Knowledge Acquisition. *Knowledge Acquisition*, 2:241–257.

[Console & Torasso, 1991] Console, L. & Torasso, P. (1991). A Spectrum of Definitions of Model-Based Diagnosis. *Computational Intelligence*, 7:133–141.

[Corbridge *et al.*, 1995] Corbridge, C., Major, N., & Shadbolt, N. (1995). Models Exposed: An Empirical Study. In *Proceedings of the 9th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge Based Systems.*

[Craw *et al.*, 1994] Craw, S., Sleeman, D., Boswell, R., & Carbonara, L. (1994). Is knowledge refinement different from theory revision? In Wrobel, S., (Ed.), *Proceedings of the MLNet Familiarization Workshop on Theory Revision and Restructuring in Machine Learning (ECML-94)*, pages 32–34.

[Crawford *et al.*, 1992] Crawford, J., Farquhar, A., & Kuipers, B. (1992). QPC: A Compiler from Physical Models into Qualitative Differential Equations. In Faltings, B. & Struss, P., (Eds.), *Recent Advances in Qualitative Physics.* The MIT Press.

[Davis *et al.*, 1993] Davis, R., Shrobe, H., & Szolovits, P. (1993). What is a Knowledge Representation? *AI Magazine*, pages 17–33.

[de Brug *et al.*, 1986] de Brug, A. V., Bachant, J., & McDermott, J. (1986). The Taming of R1. *IEEE Expert*, pages 33–39.

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

[Dreyfus, 1979] Dreyfus, H. (1979). *What Computers Can't D: A Critique of Artifical Reason.* Freeman.

[Duda *et al.*, 1985] Duda, R., Hart, P., & Reboh, R. (1985). Letter to the Editor. *Artificial Intelligence*, 26:359–360.

[Easterbrook, 1991] Easterbrook, S. (1991). Handling conflicts between domain descriptions with computer-supported negotiation. *Knowledge Acquisition*, 3:255–289.

[Eshghi, 1993] Eshghi, K. (1993). A Tractable Class of Abductive Problems. In *IJCAI '93*, volume 1, pages 3–8.

[Falkenhainer, 1990] Falkenhainer, B. (1990). Abduction as Similarity-Driven Explanation. In O'Rourke, P., (Ed.), *Working Notes of the 1990 Spring Symposium on Automated Abduction*, pages 135–139.

[Feldman *et al.*, 1989] Feldman, B., Compton, P., & Smythe, G. (1989). Hypothesis Testing: an Appropriate Task for Knowledge-Based Systems. In *4th AAAI-Sponsored Knowledge Acquisition for Knowledge-based Systems Workshop Banff, Canada.*

[Forbus, 1984] Forbus, K. (1984). Qualitative Process Theory. *Artificial Intelligence*, 24:85–168.

[Gaines & Shaw, 1989] Gaines, B. & Shaw, M. (1989). Comparing the Conceptual Systems of Experts. In *IJCAI '89*, pages 633–638.

[Ginsberg, 1990] Ginsberg, A. (1990). Theory Reduction, Theory Revision, and Retranslation. In *AAAI '90*, pages 777–782.

[Gruber, 1993] Gruber, T. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220.

[Hamscher, 1990] Hamscher, W. (1990). Explaining Unexpected Financial Results. In O'Rourke, P., (Ed.), *AAAI Spring Symposium on Automated Abduction*, pages 96–100.

[Hamscher *et al.*, 1992] Hamscher, W., Console, L., & DeKleer, J. (1992). *Readings in Model-Based Diagnosis.* Morgan Kaufmann.

[Hirata, 1994] Hirata, K. (1994). A Classification of Abduction: Abduction for Logic Programming. In *Proceedings of the Fourteenth International Machine Learning Workshop, ML-14*, page 16. Also in *Machine Intelligence* 14 (to appear).

[Kohavi *et al.*, 1996] Kohavi, R., Sommerfield, D., & Dougherty, J. (1996). Data Minining using MLC++: A Machine Learning Library in C++. In *Tools with AI 1996.*

[Kuhn, 1962] Kuhn, T. (1962). *The Structure of Scientific Revolutions.* Cambridge Press.

[Kuipers, 1986] Kuipers, B. (1986). Qualitative Simulation. *Artificial Intelligence*, 29:229–338.

[Kuipers, 1993] Kuipers, B. (1993). Qualitative Simulation: then and now. *Artificial Intelligence*, 59:133–140.

[Larkin *et al.*, 1980] Larkin, J., McDermott, J., Simon, D., & Simon, H. (1980). Expert and Novice Performance in Solving Physics Problems. *Science*, 208:1335–1342.

[Leake, 1993] Leake, D. (1993). Focusing Construction and Selection of Abductive Hypotheses. In *IJCAI '93*, pages 24–29.

[Linster, 1992] Linster, M. (1992). A review of Sisyphus 91 and 92: Models of Problem-Solving Knowledge. In Aussenac, N., Boy, G., Gaines, B., Linser, M., Ganascia, J.-G., & Kordratoff, Y., (Eds.), *Knowledge Acquisition for Knowledge-Based Systems*, pages 159–182. Springer-Verlag.

[Linster & Musen, 1992] Linster, M. & Musen, M. (1992). Use of KADS to Create a Conceptual Model of the ONCOCIN task. *Knowledge Acquisition*, 4:55–88.

[Mackworth, 1977] Mackworth, A. (1977). Consistency in Networks of Relations. *Artificial Intelligence*, 8:99–118.

[Mackworth, 1992] Mackworth, A. (1992). The Logic of Constraint Satisfaction. *Artificial Intelligence*, 58:3–20.

[Marcus *et al.*, 1987] Marcus, S., Stout, J., & McDermott, J. (1987). VT: An Expert Elevator Designer That Uses Knowledge-Based Backtracking. *AI Magazine*, pages 41–58.

[Marques *et al.*, 1992] Marques, D., Dallemagne, G., Kliner, G., McDermott, J., & Tung, D. (1992). Easy Programming: Empowering People to Build Their Own Applications. *IEEE Expert*, pages 16–29.

[McDermott, 1987] McDermott, D. (1987). A Critique of Pure Reason. *Computational Intelligence*, 3:151–160.

[Menzies, 1995a] Menzies, T. (1995a). Applications of Abduction #1: Intelligent Decision Support Systems. Technical Report TR95-16, Department of Software Development, Monash University.

[Menzies, 1995b] Menzies, T. (1995b). Limits to Knowledge Level-B Modeling (and KADS). In *Proceedings of AI '95, Australia*. World-Scientific.

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

[Menzies, 1995d] Menzies, T. (1995d). Situated Semantics is a Side-Effect of the Computational Complexity of Abduction. In *Australian Cognitive Science Society, 3rd Conference*.

[Menzies, 1996b] Menzies, T. (1996b). On the Practicality of Abductive Validation. In *ECAI '96*.

[Menzies, 1996a] Menzies, T. (September, 1996a). Applications of Abduction: Knowledge Level Modeling. *International Journal of Human Computer Studies*.

[Menzies *et al.*, 1992] Menzies, T., Black, J., Fleming, J., & Dean, M. (1992). An Expert System for Raising Pigs. In *The first Conference on Practical Applications of Prolog*.

[Menzies & Compton, 1994] Menzies, T. & Compton, P. (1994). *A Precise Semantics for Vague Diagrams*, pages 149–156. World Scientific.

[Menzies & Haynes, 1994] Menzies, T. & Haynes, P. (1994). The Methodologies of Methodologies; or, Evaluating Current Methodologies: Why and How. In *Tools Pacific '94*, pages 83–92. Prentice-Hall.

[Menzies & Compton, 1995] Menzies, T. J. & Compton, P. (1995). The (Extensive) Implications of Evaluation on the Development of Knowledge-Based Systems. In *Proceedings of the 9th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge Based Systems*.

[Michalski, 1993] Michalski, R. (1993). Toward a Unified Theory of Learning: Multistrategy Task-adaptive Learning. In Buchanan, B. G. & Wilkin, D. C., (Eds.), *Readings in Knowledge Acquisition and Learning: Automatic Construction and Improvement of Expert System*. Morgan Kaufmann Publishers.

[Mintzberg, 1975] Mintzberg, H. (1975). The Manager's Job: Folklore and Fact. *Harvard Business Review*, pages 29–61.

[Muggleton, 1991] Muggleton, S. (1991). Inductive Logic Programming. *New Generation Computing*, 8:295–318.

[Newell, 1982] Newell, A. (1982). The Knowledge Level. *Artificial Intelligence*, 18:87–127.

[Newell, 1993] Newell, A. (1993). Reflections on the Knowledge Level. *Artificial Intelligence*, 59:31–38.

[Newell & Simon, 1972] Newell, A. & Simon, H. (1972). *Human Problem Solving*. Prentice-Hall Englewood Cliffs, N.J.

[Newell *et al.*, 1991] Newell, A., Yost, G., Laird, J., Rosenbloom, P., & Altmann, E. (1991). Formulating the Problem Space Computational Model. In Rosenbloom, P., Laird, J., & Newell, A., (Eds.), *The Soar Papers*, volume 2, pages 1321–1359. MIT Press.

[Ng & Mooney, 1990] Ng, H. & Mooney, R. (1990). The Role of Coherence in Constructing and Evaluating Abductive Explanations. In *Working Notes of the 1990 Spring Symposium on Automated Abduction*, volume TR 90-32, pages 13–17.

[O'Rourke, 1990] O'Rourke, P. (1990). Working Notes of the 1990 Spring Symposium on Automated Abduction. Technical Report 90-32, University of California, Irvine, CA. September 27, 1990.

[Palmer & Craw, 1995] Palmer, G. & Craw, S. (1995). Utilising Explanation to Assist the Refinement of Knowledge-Based Systems. In *EUROVAV-95: The Third European Symposium on Validation and Verification of Knowledge-Based Systems*, pages 201–211.

[Patil *et al.*, 1981] Patil, R., Szolovitis, P., & Schwartz, W. (1981). Causal Understanding of Patient Illness in Medical Diagnosis. In *IJCAI '81*, pages 893–899.

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

[Poole, 1990a] Poole, D. (1990a). Hypo-Deductive Reasoning for Abduction, Default Reasoning, and Design. In O'Rourke, P., (Ed.), *Working Notes of the 1990 Spring Symposium on Automated Abduction.*, volume TR 90-32, pages 106–110.

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

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

[Popper, 1963] Popper, K. (1963). *Conjectures and Refutations,*. Routledge and Kegan Paul.

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

[Preece & Shinghal, 1992] Preece, A. & Shinghal, R. (1992). Verifying Knowledge Bases by Anomaly Detection: An Experience Report. In *ECAI '92*.

[Preston *et al.*, 1993] Preston, P., Edwards, G., & Compton, P. (1993). A 1600 Rule Expert System Without Knowledge Engineers. In Leibowitz, J., (Ed.), *Second World Congress on Expert Systems*.

[Quinlan, 1986] Quinlan, J. (1986). Induction of Decision Trees. *Machine Learning*, 1:81–106.

[Rosenbloom *et al.*, 1993] Rosenbloom, P., Laird, J., & Newell, A. (1993). *The SOAR Papers*. The MIT Press.

[Runkel, 1995] Runkel, J. (1995). Analyzing Tasks to Build Reusable Model-Based Tools. In *Proceedings of the 9th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop Banff, Canada.*

[Searle, 1980] Searle, J. (1980). Minds, Brain, and Programs. *The Behavioral and Brain Sciences*, 3:417–457.

[Searle, 1982] Searle, J. (1982). The Myth of the Computer. *The New York Review of Books*, pages 3–6. April 29.

[Searle, 1995] Searle, J. (1995). 'The Mystery of Consciousness': An Exchange. *The New York Review of Books*, pages 83–84.

[Selman & Levesque, 1990] Selman, B. & Levesque, H. (1990). Abductive and Default Reasoning: a Computational Core. In *AAAI '90*, pages 343–348.

[Shaw, 1988] Shaw, M. (1988). Validation in a Knowledge Acquisition System with Multiple Experts. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, pages 1259–1266.

[Shaw & Gaines, 1992] Shaw, M. & Gaines, B. (1992). Repgrid-net: Combining Conceptual Modeling with Electronic Mail to Provide Decision Support. In *7th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada*, pages 24.1–24.18.

[Silverman, 1990] Silverman, B. (1990). Critiquing Human Judgmet Using Knowledge-Acquisition Systems. *AI Magazine*, pages 60–79.

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

[Simon, 1960] Simon, H. (1960). *The New Science of Management Decision.* Prentice Hall.

[Smythe, 1989] Smythe, G. (1989). Brain-hypothalmus, Pituitary and the Endocrine Pancreas. *The Endocrine Pancreas.*

[Stark, 1993] Stark, M. (1993). Impacts of Object-Oriented Technologies: Seven Years of Software Engineering. *J. Systems Software*, 23:163–169.

[Steels, 1990] Steels, L. (1990). Components of Expertise. *AI Magazine*, 11:29–49.

[Steels, 1994] Steels, L. (1994). How Can we Make Further Progress in Knowledge Acquisition? In Mizoguchi, R., Motoda, H., Boose, J., Gaines, B., & Compton, P., (Eds.), *Proceedings of the Third Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop, JKAW '94*, pages 65–71.

[Stefik *et al.*, 1982] Stefik, M., Aikins, J., Balzer, R., Benoit, J., Birnhaum, L., Hayes-Roth, F., & Sacerdoti, E. (1982). The Organisation of Expert Systems, A Tutorial. *Artificial Intelligence*, 18:135–127.

[Suchman, 1987] Suchman, L. (1987). Book Review of Winograd & Flores, Understanding Computers and Cognition: A New Foundation for Design. *Artificial Intelligence*, 31:227–232.

[Suchman, 1993] Suchman, L. (1993). Response to Vera and Simon's Situated Action: A Symbolic Interpretation. *Cognitive Science*, 17:71–75.

[Tansley & Hayball, 1993] Tansley, D. & Hayball, C. (1993). *Knowledge-Based Systems Analysis and Design.* Prentice-Hall.

[van Harmelen & Bundy, 1988] van Harmelen, F. & Bundy, A. (1988). Explanation-Based Generalisation = Partial Evaluation. *Artificial Intelligence*, pages 401–412.

[Vera & Simon, 1993a] Vera, A. & Simon, H. (1993a). Situated Action: A Response to Reviewers. *Cognitive Science*, 17:77–86.

[Vera & Simon, 1993b] Vera, A. & Simon, H. (1993b). Situated Action: A Symbolic Interpretation. *Cognitive Science*, 17:7–48.

[Vera & Simon, 1993c] Vera, A. & Simon, H. (1993c). Situated Action: Reply to William Clancey. *Cognitive Science*, 17:117–133.

[Weiss *et al.*, 1978] Weiss, S., Kulikowski, C., & Amarel, S. (1978). A Model-Based Method for Computer-Aided Medical Decision-Making. *Artificial Intelligence*, 11.

[Wielinga *et al.*, 1992] Wielinga, B., Schreiber, A., & Breuker, J. (1992). KADS: a Modeling Approach to Knowledge Engineering. *Knowledge Acquisition*, 4:1–162.

[Winograd & Flores, 1987] Winograd, T. & Flores, F. (1987). On Understanding Computers and Cognition: A New Foundation for Design: A respose to the reviews. *Artificial Intelligence*, 31:250–261.

[Yost & Newell, 1989] Yost, G. & Newell, A. (1989). A Problem Space Approach to Expert System Specification. In *IJCAI '89*, pages 621–627.

[Yu *et al.*, 1979] Yu, V., Fagan, L., Wraith, S., Clancey, W., Scott, A., Hanigan, J., Blum, R., Buchanan, B., & Cohen, S. (1979). Antimicrobial Selection by a Computer: a Blinded Evaluation by Infectious Disease Experts. *Journal of American Medical Association*, 242:1279–1282.

[Zlatereva, 1992] Zlatereva, N. (1992). Truth Mainteance Systems and Their Application for Verifying Expert System Knowledge Bases. *Artificial Intelligence Review*, 6.

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

Some of the Menzies papers can be found at *http:// www.cse.unsw.edu.au/ ~timm/pub/ docs/papersonly.html.*