Target: editorial for International Journal of Human Computer Studies, Special Issue on Evaluation of Knowledge Engineering Techniques, 2000, WP:b/99/evalke/eke(August 7, 1999).

Editorial: Evaluating Knowledge Engineering Techniques

Tim Menzies[‡], **Frank van Harmelen**[†]
[‡]NASA/WVU IV&V Facility, 100 University Drive, Fairmont WV 26554
[†]Department of Mathematics and Computer Science, Vrije Universiteit, Amsterdam, The Netherlands
<tim@menzies.com,frankh@cs.vu.nl>

Introduction

An editorial for a special issue should summarise the contents of that issue and expand on the area sampled by that issue. Hence:

- For a quick guide to knowledge engineering (KE) evaluation material, see Figure 1 and Figure 2.
- For notes on two major evaluation experiments, see Figure 3 and Figure 4.
- For our view on the current state of the art in KE evaluation, see the next section. Based on this review, Menzies [1999] has proposed a "next generation" KE evaluation experiment that address some of the drawbacks in the current state of the art.
- For a review of the papers in this special issue, see the end of this editorial.

Concerning the State of the Art

For the time it takes to read these pages, let's take a break. During this break, we will stop generating untold numbers of knowledge modeling publications. While we catch our breath, we will have time to ask a basic question:

> *Can we build better knowledge based systems (KBS) faster now than in the 80's?*

We argue below that knowledge engineers (KEs) use a range of techniques ($T_0, T_1, \ldots T_5$). Yet more techniques are constantly being evolved. How sure are we that any of the current KE techniques either work at all or work better than past practice?

Currently, our basic question has no clear answer. It is not easy to evaluate software development techniques, be they from software engineering or knowledge engineering (see the list of problems in [Shadbolt, O'Hara & Crow 2000]). However, many successful evaluations have been performed, as evidenced by numerous KE case studies studies (Figure 1) and the wealth of material on KE evaluation (Figure 2). In our experience, the core problem with performing evaluations is a concern that the whole principle of evaluation is misguided. KE researchers are often reluctant to design and perform rigorous evaluations, asking "what is the general point of such studies"? That is, how can results from one study be relevant to anything else except the combination of core technology, interface, user group, and problem domain used in that study. The generality of an experiment's conclusion is a valid concern. We will argue below that such general conclusions can be made, but only if KE researchers adopt a more general view of their work.

The rest of this paper tries to encourage that broader perspective. We begin with Newell's classic objection to poorly-formed evaluation studies: "Why you can't play 20 questions with Nature, and win". This objection is often cited as an argument against evaluations. Our refutation of that objection will lead to the definition of an important evaluation concept: T- the *essential theory*. Six such essential

---

Detailed surveys of evaluation techniques: [Fenton 1991, Cohen 1995].

Introductory remarks on evaluation: [Reich 1995, Fenton, Pfleeger & Glass 1994, Gaschnig, Klahr, Pople, Shortliffe & Terry 1983] (includes comments on experimental methods, software measurement, and the evaluation of expert systems). Also, in this issue, see the excellent general introductory remarks in [Shadbolt et al. 2000, Hori 2000] as well as the specific literature review on KE testing in [Caraca-Valente, Gonzalez, Morant & Pozas 2000].

Example evaluation studies: HPKB (Figure 3); the Sisyphus project (Figure 4); [Caraca-Valente et al. 2000, Gordon & Shortliffe 1985, Hori 2000, Lee & O'Keefe 1996, Menzies, Black, Fleming & Dean 1992, Menzies 2000, Preston, Edwards & Compton 1993, Reich 1995, Shadbolt et al. 2000, Stroulia & Goel 2000, Waugh, Menzies & Goss 1997, Weiss, Kulikowski & Amarel 1978].

Very good empirical evaluations: [Yu, Fagan, Wraith, Clancey, Scott, Hanigan, Blum, Buchanan & Cohen 1979, Corbridge, Major & Shadbolt 1995, Menzies 1996, Vicente, Christoffersen & Pereklita 1995, Sanderson, Verhapge & Fuld 1989].

Excellent empirical evaluations: [Hayes & Parzen 1997, Yost 1992].

Critiques of current KE evaluation practice: [Menzies 1998]. See also [Menzies 1999] for the definition of a "next generation" KE evaluation experiment that addresses some of the drawbacks in current practice.

Figure 1: Empirical studies: references

Banff KA workshop : Track on Evaluation of Knowledge-Acquisition Methodologies `http://ksi.cpsc.ucalgary.ca/KAW/`

Evaluation of Intelligent Systems: `http://eksl-www.cs.umass.edu:80/eis`

Evaluation Methods for Knowledge Engineering: `http://www.cse.unsw.EDU.AU/~timm/pub/eval`

The V&V Annotated Bibliography: `http://www.csd.abdn.ac.uk/~apreece/Research/vvbiblio.html`

Partial lists of interesting KE-related experiments: `http://www.cse.unsw.edu.au/~timm/pub/eval/summary.html` (this page uses a web page of references relating to evaluation: `http://www.cse.unsw.edu.au/~timm/pub/eval/refs.html`).

Figure 2: Web-based KE evaluation resources

theories in modern KE will then be listed. Though exceptions exist, most KE researchers work in one of these six niches. We will argue that researchers who look beyond their niche can find the data needed to evaluate their work.

Essential Theories

A common rationalization for a lack of evaluations is Newell's [1972] famous argument "Why you can't play 20 questions with Nature, and win". Reacting to an excess of experimental zeal in cognitive psychology, Newell argued forcibly against conducting experimental programmes that offered single answers to yes-no questions. Rather, said Newell, we should:

- Perform multiple studies of complex systems.
- Collect large amounts of data.
- Unify those results into some rich theoretical structure describing some process of interest. In the sequel, we will call such a rich theoretical structure the *essential theory* T.

Newell argued that large simulations to incrementally explore parts of an essential theory are far more insightful than (e.g.) twenty yes-no questions.

In reply, Cohen (personal communication) argues that Newell confused the specifics of an observation with the generality of the implication of that observation. So, to paraphrase Newell, you can play 20 questions (or less) with essential theories and still make scientific progress. However, in order to succeed with a small number of questions, those questions must relate to critical pathways in some essential theory. For example, consider the question "do heavy objects fall faster in vacuum than light objects?". The "yes" answer would have enormous implications for our vision of the universe since it would challenge base assumptions about our essential theory of gravity.

One advantage of defining an essential theory T is that it can be clear distinguished from a competing theory $\neg T$. T can be comparatively assessed by measuring variables shared in T and $\neg T$.

Essential KE Theories

For many years, we have been active KE researchers and have debated KE extensively with our colleagues. Based on what we have seen at:

- The international Knowledge Acquisition workshops (`http://ksi.cpsc.ucalgary.ca/KAW/`);
- DARPA's High Performance Knowledge-Based systems initiative (see Figure 3);
- and the Sisyphus project (Figure 4)

we claim there are at least six essential theories $T_0 \ldots T_5$ in contemporary KE. We take it that the core belief of proponents of each theory is that:

> T_i is a more productive technique for building KBS than $T_j, j \neq i$.

Figure 5 shows that the different T. Each T_i uses a different combination of software constructs:

Procedures: Libraries storing know-how represented as procedural code.

Axioms: Assertions about a particular domain.

Single general-purpose inference engines [1]

PSMs: Libraries of declarative representations of inferencing cliches[2]; e.g. the diagnosis PSM in Figure 6.

Ontologies: Libraries of abstracted data types seen in different domains[3]. PSMs and ontologies are linked: ontologies define the data types required by a PSM to execute in a domain. For example, the diagnosis PSM of Figure 6 needs (e.g.) *complaint*s and *observable*s to execute.

Figure 5 categorizes the different T_i according to what is discussed during KBS design sessions:

T_0: Rejects the declarative representations used in $T_i, i > 0$. In the 70's, T_0 was a large research area. Proponents of frame representations (e.g. [Winograd 1975, Minsky 1975]) argued that part of human expertise was "know-how" and these recipes of "how" to solve a problem were best modeled as (e.g.) Lisp procedures attached to frame slots. The debate

[1] E.G. Prolog [Kowalski 1988], OPS5 [Forgy 1982], SOAR [Laird & Newell 1983, Rosenbloom, Laird & Newell 1993], PSCM [Yost & Newell 1989], GSAT [Selman, Levesque & Mitchell 1992], ISAMP [Crawford & Baker 1994],...

[2] E.G. SPARK/BURN/FIREFIGHTER (SBF) [Marques, Dallemagne, Kliner, McDermott & Tung 1992]; generic tasks [Chandrasekaran, Johnson & Smith 1992]; configurable role-limiting methods [Swartout & Gill 1996, Gil & Melz 1996]; model construction operators [Clancey 1992]; CommonKADS [Wielinga, Schreiber & Breuker 1992, Schreiber, Wielinga, Akkermans, Velde & de Hoog 1994]; the PROTEGE family of systems [Eriksson, Shahar, Tu, Puerta & Musen 1995]; components of expertise [Steels 1990]; MIKE [Angele, Fensel & Studer 1996]; TINA [Benjamins 1995].

[3] E.G. [Lenat & Gutha 1990, Gruber 1993, Neches, Fikes, Finin, Gruber, Patil, Senator & Swartout. 1991, Uschold & Gruninger 1996, Noy & Hafner 1997, van Heust, Schreiber & Wielinga 1997]. In recent years, ontologies have also appeared in research into software architectures and design patterns [Menzies 1997].

Goal: To foster the development of technologies that can increase the rate at which we can write knowledge bases.

Baseline: Current KB authoring rates average at 5 axioms per hour, 10,000 axioms per year.

Aim: To increase the baseline by one to two orders of magnitude.

Organization: DARPA funds bi-annual meetings and two "intergration teams" (SAIC and Teknowledge) whose role is to build unified workbenches from the contributions of HPKB participants.

Participants: HPKB participants come, for the most part, from the United States. These participants, and the principle investigators, include *CYCORP*:Lenat; *Stanford*: Fikes, Koller, McCarthy, Wiederhold, Musen; *George Mason University*: Tecuci; *Carnegie Mellon University*: Mitchell; *University of Massachusetts*: Cohen; *ISI*: MacGregor, Patil, Swartout, Gil; *SRI*: Lowrance, desJardins, Goldszmidt; *Kestrel*: Espinosa; *MIT*: Doyle, Katz; *Northwestern University*: Forbus; *AIAI at Uni. Edinburgh*: Kingston, Tate; *TextWise*: Liddy, Hendler.

Biases: HPKB ignored, for the most part, the problem solving methods (PSMs) research (see the discussion below). PSMs are a major focus of the Sisyphus project(Figure 4).

Results: 1. In HPKB year one, the George Mason team generated the most new axioms added per day (787 binary predicates) using DISCIPLE: an incremental knowledge acquisition tool [Tecuci 1998]. DISCIPLE includes machine learning tools for abstracting learnt rules which makes them more generally applicable. As DISCIPLE runs, it builds and updates the meta-knowledge used for the purposes of abstraction.

2. Cohen, Chaudhri, Pease & Schrag [1999] studied how much ontologies supported the development of HPKB applications. The recent terms added to an ontology offer more support than words added previously by other authors. Such a result does not support the current efforts in building supposedly reusable ontologies.

For more information: See `http://www.teknowledge.com/HPKB/` and [Cohen, Schrag, Jones, Pease, Lin, Starr, Gunning & Burke 1998].

Figure 3: DARPA's high-performance KB (HPKB) initiative: notes

The Sisyphus projects are a series of challenge problems in which a knowledge acquisition problem is defined and tool developers are challenged to solve it with their tools. Unlike HPKB, Sisyphus was run by a loose consortium of international KE researchers on a shoestring budget. Sisyphus began in 1990 and continues to this day. Communication is mostly via email and status reports at the semi-annual Banff KA workshops. Sisyphus is mostly populated via Europeans, but some cross-over with the HPKB community exists (i.e. Musen, Gil).
A significant biases in the Sisyphus projects is towards problem solving methods (PSMs) research (exceptions: [Richards & Menzies 1998, Yost 1994]). In this regard, Sisyphus is very different to HPKB (Figure 3).
There have been four Sisyphus projects defined:

Sisyphus-I: Room Allocation [Linster 1992].

Sisyphus-II: Elevator Configuration [Schreiber & Birmingham 1996, Marcus, Stout & McDermott 1987].

Sisyphus-III: Lunar Igneous Rock Classification [Shadbolt et al. 2000].

Sisyphus-IV: Integration over the Web

While Sisyphus has unified a diverse range of researchers, and hundreds of publications have been generated, the objective evaluation results to date are inconclusive:

> ... none of the Sisyphus experiments have yielded much evaluation information (though at the time of this writing Sisyphus-III is not yet complete) [Shadbolt et al. 2000].

Nevertheless, the Sisyphus researchers remain optimistic and the project continues.

Figure 4: Notes on the Sisyphus initiatives. Extended from notes at `http://ksi.cpsc.ucalgary.ca/KAW/Sisyphus/`.

continues to this day [Nilsson 1991, Birnbaum 1991] but the complexity of reasoning about procedures (e.g. [Etherington & Reiter 1983]) drove most researchers to declarative characterizations of their frame-based knowledge (e.g. [Brachman, Gilbert & Levesque 1989]). T_0 researchers are rare these days, but some still keep the faith e.g. [Birnbaum 1991, Brooks 1991]. Few (?none) T_0 researchers are known amongst the HPKB and Sisyphus communities.

T_1**:** No explicit representation meta-knowledge when modeling KBS and a single inference procedure. Crudely expressed, in T_1, KE is just a matter of stuffing axioms into an inference engine and letting the inference engine work it all out. Successful T_1 variants provide rigid control on how new axioms are asserted, e.g. [Compton & Jansen 1990].

T_2**:** Active focus on ontology creation. Ontologies may never execute: rather they may be an analysis tool for a domain. Software engineers who develop architectures or design patterns but do not execute these abstractions directly are T_2 (e.g. [Gamma, Helm, Johnson & Vlissides 1995]).

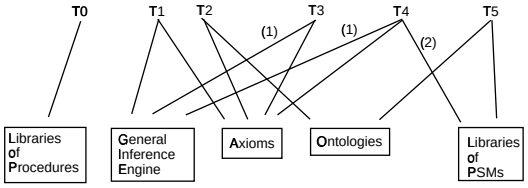

Figure 5: Different schools of knowledge engineering. (1) denotes that the single inference engine is customizable; e.g. the knowledge engineer can provide operator selection rules to customize the problem space traversal [Laird & Newell 1983]. (2) denotes that PSMs in T_4 are used only in an initial analysis stage.

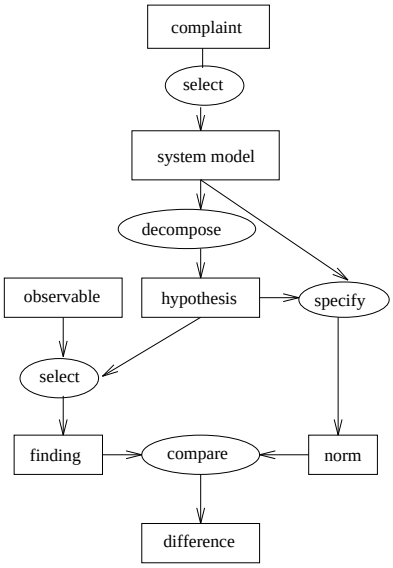

Figure 6: A PSM for diagnosis; ovals=functions, rectangles=data types.

T_3: A strong commitment to a single inference procedure, which can be customized. This inference procedure features predominantly when modeling a system; e.g. [Yost & Newell 1989, Laird & Newell 1983, Menzies & Mahidadia 1997].

T_4 : A hybrid T_3/T_5 approach in which PSMs are used to structure the analysis discussions but then converted by the knowledge engineer at design time to T_3 [Chandrasekaran et al. 1992].

T_5: Catalogues libraries of PSMs [Wielinga et al. 1992] or explores a single PSM within such a library [Marcus & McDermott 1989]. Extensive use of ontologies. At runtime, T_5 may use a general inference engine to execute their systems (e.g. older versions of PROTEGE-II [Eriksson et al. 1995] compiled down to CLIPS [NASA 1991]) but this inference engine does not feature in the design discussions. An example PSM is shown in Figure 6.

While generalizations are hard to form from the diverse international KE research community, it is roughly true that: HPKB-style KE advocates $T_i, 2 \leq i < 5$ while Sisyphus-style KE advocates T_5. Roughly speaking, in HPKB-style KE:

- Axioms and ontologies are all; e.g. [Lenat & Gutha 1990].
- PSMs are ignorable; exceptions: [Eriksson et al. 1995, Swartout & Gill 1996].

One of the reason for this difference in the two communities is the dominant research projects in different regions. The KADS project [Wielinga et al. 1992], based at the University of Amsterdam, was very influential in Europe, and most of the Sisyphus contributions came from Europe. On the other hand, nearly all of the HPKB participants come from the USA. The CYC project has a high profile in the United States [Lenat & Gutha 1990], and that project focuses on ontologies, not PSMs. Musen (personnel communication) laments this geographical distribution of the KE styles, remarking that Sisyphus-style PSM-based KE came originally from the following US research:

- Chandrasekaran's [1983] identification of "generic tasks";
- Clancey's [1985] analysis of heuristic classification;
- Kahn, Nowlan & McDermott's [1985] and Marcus & McDermott's [1989] work on method-based KA tools;
- Musen's own work in the 80's on PROTEGE-I which lead to the PROTEGE family of systems [Eriksson et al. 1995].

Musen's view is that PSMs transferred to Europe via: Steels's [1990] "Components of Expertise" paper; the subsequent work on the KREST system; and the CommonKADS project [Breuker & de Velde (eds) 1994]. Meanwhile, back in the USA, PSM research had little effect on the mainstream, as witnessed by HPKB. However, US-based PSM research remains strong at Stanford Medical Informatics [Eriksson et al. 1995]; the University of Southern California's Information Science Institute [Swartout & Gill 1996] and the computer science department at Ohio State University [Chandrasekaran et al. 1992].

Curiously, even though this clear international division exists in the KE community, comparative data on the merits of T_5 vs $\neg T_5$ are very few. These data points come from the Sisyphus project:

- In the Sisyphus-II initiative, Yost [1994] built a T_3 elevator configuration system using PSCMs while the other teams used T_5 [Schreiber & Birmingham 1996]. Yost's reported development times were much less than any of the other teams.
- In the Sisyphus-III initiative, Richards & Menzies [1998], used a T_1 approach to produce a working system. In that round of the Sisyphus-III project, the T_5 teams were still debugging their ontologies. Further, the T_1 developers had had time to explore extending their Sisyphus-III beyond the original Sisyphus-III specification (tools were implemented to handle competing viewpoints during requirements engineering).

Reuse Model	% disorders identified	% knowledge fragments identified
Straw man: invented very quickly	50	28
Mature model: decades of work	55	34
No model	75	41

Figure 7: Productivity using different models.

These data points are hardly conclusive. For example, Yost had analysed the domain extensively prior to system construction. That is, it is possible that Yost's extra experience with the domain gave him an advantage. Nevertheless, the one general conclusion we would offer is that more data points should be collected. This lack of PSM-evaluation is strange, given that the Sisyphus-style KE community has already seen results negative to the PSM paradigm. Corbridge et al. [1995] conducted a case study amongst international KE experts. Each expert was given some background knowledge to guide their analysis of a transcript of a patient talking to a doctor. Analysis time was restricted to a few hours. One group used an abstract model of diagnosis matured over many years (a variant of Figure 6); another used an abstract model invented very quickly (the "straw man"); and the rest used no model at all. The results are shown in Figure 7. The "mature model" group performed as well as the "straw man" group. Further, the "no model" group outperformed the groups using the models! Shadbolt et al.'s [2000] retrospective on this study argues that perhaps these results are a product of the learning curve required to understand the domain. In that view, the use of unfamiliar models initially impairs performance but over a longer period of time causes improvements. To the best of our knowledge, no T_5 researcher is exploring this possibility in an experiment.

Discussion

We started with the question:

> *Can we build better knowledge based systems (KBS) faster now than in the 80's?*

and our answer is,

> *We still don't know.*

Our case has been that if evaluations are defined too narrowly, then researchers cannot distinguish between competing issues. Hence, in this discussion, we have tried to broaden the focus of KE evaluation work. While we acknowledge all the problems with evaluation listed by Shadbolt et al. [2000], we disagree with their view that Sisyphus-style evaluations are the best future direction for KE evaluation research. Rather, if we take a broad view of KE research, we find a range of competing technologies ($T_0 \ldots T_5$) for building KBS. Our belief is that the future of KE evaluation will be in comparative evaluations across these essential theories as follows:

KE eval step 1: Identify a process of interest.

KE eval step 2: Create an essential theory T for that process.

KE eval step 3: Identify some competing process description, $\neg T$.

KE eval step 4: Design an study that explores core pathways in both $\neg T$ and T. Use the $\neg T$ results as baseline figures to chart the benefits (or otherwise) or the T approach. For example, the RDR T_1 technique is simple to implement [Compton & Jansen 1990, Richards & Menzies 1998] and could be used to collect baseline performance data.

KE eval step 5: Acknowledge that your study may not be definitive. Further studies or reference to previous studies may be required to make a conclusion.

The need for multiple experiments is widely recognized. An ideal experiment has (1) tight experimental controls; (2) a correspondence of the experimental situation to the real situation; and (3) some generality to situations beyond the specifics of the experiment. Experience has shown that is hard/impossible to do all three at once. Individual studies must compromise on one of these dimensions, and multiple studies are required to offer converging evidence on some issue [Sanderson et al. 1989]. Newell himself recommends multiple studies (recall his comments, above). The "next generation" KE evaluation study proposed by Menzies [1999] assumes multiple evaluations.

Concerning the Papers in this Volume

The above discussion tried to point ahead to the future of KE evaluation. The current state of the art is surveyed by the papers in this volume. Newcomers to the field of KE evaluation can gain an overview of this field by:

- Reading the excellent general introductory remarks found in [Shadbolt et al. 2000, Hori 2000];
- Reading the specific literature review on KE testing in [Caraca-Valente et al. 2000];
- Selectively exploring the rest of this volume and then the material described in Figure 1 and Figure 2.

Specific notes on each paper follow.

Shadbolt et al. [2000] offers a detailed analysis of many years of experimental research on knowledge acquisition conducted at Nottingham University. There is much to recommend this paper including:

- A detailed analysis of the problems associated with evaluation.
- A description of an exciting series of experiments aimed at key issues in knowledge acquisition.
- The careful exposition of the rationale behind the Sisyphus projects.
- An excellent literature review.

We have one minor academic quibble with this paper, which we trust Shadbolt et al. [2000] will excuse us for mentioning. The general message of Shadbolt et al. [2000] is that projects like Sisyphus are about as much as we can expect from KE evaluation. Clearly this is not our view (see the above or [Menzies 1999]).

Caraca-Valente et al. [2000] explore a important sub-area of evaluation: a cost-benefit analysis of performing further testing. Intuitively, we all know that there is some law of diminishing returns with any test procedure. In the case of safety critical systems, we accept that any amount of testing may not be enough (but see the interesting discussion in [Littlewood & Wright 1997]). However, for the average developer, the extra benefits of further testing must be carefully weighed against the cost of that extra work. Caraca-Valente et.al. define a mathematical function that lets us precisely recognizing when we are over-testing a KB. Surprisingly few tests are cost-benefit optimum for a KB. Further, the efficacy of their procedure can be improved via tuning their model using data taken from the domain. In other work, Menzies & Cukic [1999] has tried to generalise Caraca-Valente et.al.'s observation.

Menzies [2000] tries to answer a hard question: how to assess a system in the absence of an oracle. One method for oracle-less testing is a *critical success metric* (CSM). A CSM is an assessment of a running program which reflects the business concerns that prompted the creation of that program. Given *pre-disaster* knowledge, a CSM can be used while the expert system is in routine use without compromising the operation of the system. Menzies defines a general CSM experiment using pre-disaster points which can compare (e.g.) human to expert system performance. Examples of using CSMs are given from the domains of farm management and process control.

After Shadbolt et.al., Hori [2000] describes the longest running experiment reported in this volume. This paper is a detailed and learned description of a multi-year experiment to assess the merits of a domain-oriented library of problem solving methods. This paper contains many worthy comments regarding the theory of evaluation and concludes with some timely remarks on the perils of inappropriate measures of reuse. The paper also makes extensive reference to the standard software engineering metrics literature including Basili's [1992] goal-question-metric paradigm: a commonly used metrics tool. Hence, apart from the other merits of the paper, the Hori paper is an excellent bridge from knowledge engineering to software engineering.

Stroulia & Goel [2000] tackles the question of tool support for repairing broken systems. Detailed implementation techniques are discussed concerning fixing problem solving methods. The paper is an exemplary KE evaluation paper since it states an active hypothesis and then discusses experiments that explore that hypothesis.

Acknowledgements

In preparing this paper, we grateful acknowledge the help of our hard-working reviewers and the IJHCS editorial staff. This work was partially supported by NASA through cooperative agreement #NCC 2-979.

References

Angele, J., Fensel, D. & Studer, R. [1996], Domain and task modelling in mike, *in* A. S. et.al., ed., 'Domain Knowledge for Interactive System Design', Chapman & Hall.

Basili, V. R. [1992], The experimental paradigm in software engineering, *in* H. D. Rombach, V. R. Basili & R. W. Selby, eds, 'Experimental Software Engineering Issues: Critical Assessment and Future Directions, International Workshop, Germany', pp. 3–12.

Benjamins, R. [1995], 'Problem-solving methods for diagnosis and their role in knowledge acquisition', *International Journal of Expert Systems: Research & Applications* **8**(2), 93–120.

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

Brachman, R., Gilbert, V. & Levesque, H. [1989], An Essential Hybrid Reasoning System: Knowledge and Symbol Level Accounts of Krypton, *in* J. Mylopoulos & M. Brodie, eds, 'Readings in Artificial Intelligence and Databases', Morgan Kaufmann, pp. 293–300.

Breuker, J. & de Velde (eds), W. V. [1994], *The CommonKADS Library for Expertise Modelling*, IOS Press, Netherlands.

Brooks, R. [1991], 'Intelligence without representation', *Artificial Intelligence* **47**, 139–159.

Caraca-Valente, J., Gonzalez, L., Morant, J. & Pozas, J. [2000], 'Knowledge-based systems validation: When to stop running test cases', *International Journal of Human-Computer Studies* . To appear.

Chandrasekaran, B. [1983], 'Towards a Taxonomy of Problem Solving Types', *AI Magazine* pp. 9–17.

Chandrasekaran, B., Johnson, T. & Smith, J. W. [1992], 'Task structure analysis for knowledge modeling', *Communications of the ACM* **35**(9), 124–137.

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

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

Cohen, P. [1995], *Empirical Methods for Artificial Intelligence*, MIT Press.

Cohen, P., Chaudhri, V., Pease, A. & Schrag, R. [1999], Does prior knowledge facilitate the development of knowledge-based systems?, *in* 'AAAI'99'.

Cohen, P., Schrag, R., Jones, E., Pease, A., Lin, A., Starr, B., Gunning, D. & Burke, M. [1998], 'The darpa high-performance knowledge bases project', *AI Magazine* **19**(4), 25–49.

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

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'.

Crawford, J. & Baker, A. [1994], Experimental results on the application of satisfiability algorithms to scheduling problems, *in* 'AAAI '94'.

Eriksson, H., Shahar, Y., Tu, S. W., Puerta, A. R. & Musen, M. A. [1995], 'Task modeling with reusable problem-solving methods', *Artificial Intelligence* **79**(2), 293–326.

Etherington, D. & Reiter, R. [1983], On inheritance hierarchies with exceptions, *in* 'AAAI-83', pp. 104–108.

Fenton, N. E. [1991], *Software Metrics*, Chapman and Hall, London.

Fenton, N., Pfleeger, S. & Glass, R. [1994], 'Science and Substance: A Challenge to Software Engineers', *IEEE Software* pp. 86–95.

Forgy, C. [1982], 'RETE: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem', *Artificial Intelligence* pp. 17–37.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. [1995], *De-*

sign Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley.

Gaschnig, J., Klahr, P., Pople, H., Shortliffe, E. & Terry, A. [1983], Evaluation of expert systems: Issues and case studies, *in* F. Hayes-Roth, D. Waterman & D. Lenat, eds, 'Building Expert Systems', Addison-Wesley, chapter 8, pp. 241–280.

Gil, Y. & Melz, E. [1996], Explicit representations of problem-soving strategies to support knowledge acquisition, *in* 'Proceedings AAAI' 96'.

Gordon, J. & Shortliffe, E. H. [1985], 'A method for managing evidential reasoning in a hierarchical hypothesis space', *Artificial Intelligence* **26**(3), 323–357.

Gruber, T. [1993], 'A translation approach to portable ontology specifications', *Knowledge Acquisition* **5**(2), 199–220.

Hayes, C. & Parzen, M. [1997], 'Quen: An achivement test for knowledge-based systems', *IEEE Transactions of Knowledge and Data Engineering* **9**(6), 838–847.

Hori, M. [2000], 'Stability of a domain-oriented component library: An explanatory case study', *International Journal of Human Computer Studies* . (to appear).

Kahn, G., Nowlan, S. & McDermott, J. [1985], 'Strategies for knowledge acquisition', *IEEE Trsansactions on Pattern Analysis and Machine Intelligence* **Vol. PAMI-7**, 511–522.

Kowalski, R. A. [1988], 'The early years of logic programming', *Communications of the ACM* **31**(1), 38–43.

Laird, J. & Newell, A. [1983], A universal weak method: Summary of results, *in* 'IJCAI '83', pp. 771–773.

Lee, S. & O'Keefe, R. [1996], 'The effect of knowledge representation schemes on maintainability of knowledge-based systems', *IEEE Transactions on Knowledge and Data Engineering* **8**(1), 173–178.

Lenat, D. & Gutha, R. [1990], 'Cyc: A midterm report', *AI Magazine* pp. 32–59.

Linster, M. [1992], A review of sisyphus 91 and 92: Models of problem-solving knowledge, *in* N. Aussenac, G. Boy, B. Gaines, M. Linser, J.-G. Ganascia & Y. Kordratoff, eds, 'Knowledge Acquisition for Knowledge-Based Systems', Springer-Verlag, pp. 159–182.

Littlewood, B. & Wright, D. [1997], 'Some conservative stopping rules for the operational testing of safety-critical software', *IEEE Transactions on Software Engineering* **23**(11), 673–683.

Marcus, S. & McDermott, J. [1989], 'SALT: A Knowledge Acquisition Language for Propose-and-Revise Systems', *Artificial Intelligence* **39**, 1–37.

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

Marques, D., Dallemagne, G., Kliner, G., McDermott, J. & Tung, D. [1992], 'Easy Programming: Empowering People to Build Their own Applications', *IEEE Expert* pp. 16–29.

Menzies, T. [1996], On the practicality of abductive validation, *in* 'ECAI '96'. Available from `http://www.cse.unsw.edu.au/~timm/pub/docs/96abvalid.ps.gz`.

Menzies, T. [1997], 'OO patterns: Lessons from expert systems', *Software Practice & Experience* **27**(12), 1457–1478. Available from `http://www.cse.unsw.edu.au/~timm/pub/docs/97probspatt.ps.gz`.

Menzies, T. [1998], 'Evaluation issues for problem solving methods'. Banff KA workshop, 1998. Available from `http://www.cse.unsw.edu.au/~timm/pub/docs/97eval`.

Menzies, T. [1999], hQkb- the high quality knowledge base initiative (sisyphus v: Learning design assessment knowledge), *in* 'KAW'99: the 12th Workshop on Knowledge Acquisition, Modeling and Management, Voyager Inn, Banff, Alberta, Canada Oct 16-22, 1999 (submitted)'. Available from `http://www.csee.wvu.edu/~timm/docs/9905hqkb0.html`.

Menzies, T. [2000], 'Critical success metrics: Evaluation at the business-level'. International Journal of Human-Computer Studies, special issue on evaluation of KE techniques.

Menzies, T., Black, J., Fleming, J. & Dean, M. [1992], An expert system for raising pigs, *in* 'The first Conference on Practical Applications of Prolog'. Available from `http://www.cse.unsw.EDU.AU/~timm/pub/docs/ukapril92.ps.gz`.

Menzies, T. & Cukic, B. [1999], Intelligent testing can be very lazy, *in* 'Proceedings, AAAI '99 workshop on Intelligent Software Engineering, Orlando, Florida'. Available from `http://research.ivv.nasa.gov/docs/techreports/1999/NASA-IVV-99-006.pdf`.

Menzies, T. & Mahidadia, A. [1997], Ripple-down rationality: A framework for maintaining psms, *in* 'Workshop on Problem-Solving Methods for Knowledge-based Systems, IJCAI '97, August 23.'. Available from `http://www.cse.unsw.edu.au/~timm/pub/docs/97rdra.ps.gz`.

Minsky, M. [1975], A framework for representing knowledge, *in* 'The Psychology of Computer Vision'.

NASA [1991], 'CLIPS Reference Manual', Software Technology Branch, lyndon B. Johnson Space Center.

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T. & Swartout., W. R. [1991], 'Enabling technology for knowledge sharing', *AI Magazine* **12**(3), 16–36.

Newell, A. [1972], You can't play 20 questions with nature, and win, *in* W. Chase, ed., 'Visual Information Processing', New York: Academic Press, pp. 283–308.

Nilsson, N. [1991], 'Logic and artificial intelligence', *Artificial Intelligence* **47**, 31–56.

Noy, N. F. & Hafner, C. [1997], 'The state of the art in ontology design: A survey and comparative review', *AI Magazines* pp. 53–74.

Preston, P., Edwards, G. & Compton, P. [1993], A 1600 Rule Expert System Without Knowledge Engineers., *in* J. Leibowitz, ed., 'Second World Congress on Expert Systems'.

Reich, Y. [1995], 'Measuring the value of knowledge', *International Journal of Human-Computer Studies* **42**(1), 3–30.

Richards, D. & Menzies, T. [1998], Extending the sisyphus iii experiment from a knowledge engineering task to a requirements engineering task, *in* 'Banff Workshop on Knowledge Acquisition'. Available from `http://www.cse.unsw.edu.au/~timm/pub/docs/98kawre.ps.gz`.

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

Sanderson, P., Verhapge, A. & Fuld, R. [1989], 'State-space and verbal protocol methods for studying the human operator in process control', *Ergonomics* **32**(11), 1343–1372.

Schreiber, A. T. & Birmingham, W. P. [1996], 'The sisyphus-vt initiative', *International Journal of Human-Computer Studies* **44**(3/4).

Schreiber, A. T., Wielinga, B., Akkermans, J. M., Velde, W. V. D. & de Hoog, R. [1994], 'Commonkads. a comprehensive methodology for kbs development', *IEEE Expert* **9**(6), 28–37.

Selman, B., Levesque, H. & Mitchell, D. [1992], A new method for solving hard satisfiability problems, *in* 'AAAI '92', pp. 440–446.

Shadbolt, N., O'Hara, K. & Crow, L. [2000], 'The experimental evaluation of knowledge acquisition techniques and methods: History, problems and new directions', *International Journal*

of Human-Computer Studies . (to appear).

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

Stroulia, E. & Goel, A. [2000], 'Evaluating psms in redesign: The autognostic experiments', *International Journal of Human Computer Studies* . (to appear).

Swartout, B. & Gill, Y. [1996], Flexible knowledge acquisition through explicit representation of knowledge roles, *in* '1996 AAAI Spring Symposium on Acquisition, Learning, and Demonstration: Automating Tasks for Users'.

Tecuci, G. [1998], *Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies*, Academic Press.

Uschold, M. & Gruninger, M. [1996], 'Ontologies: Principles, methods, and applications', *The Knowledge Engineering Review* **11**(2), 93–136.

van Heust, G., Schreiber, A. T. & Wielinga, B. [1997], 'Using explicit ontologies in kbs development', *International Journal of Human Computer Studies* **45**, 183–292.

Vicente, K., Christoffersen, K. & Pereklita, A. [1995], 'Supporting operator problem solving through ecological interface design', *IEEE Transactions of Systems, Man, and Cybernetics* **25**(4529-545).

Waugh, S., Menzies, T. & Goss, S. [1997], Evaluating a qualitative reasoner, *in* A. Sattar, ed., 'Advanced Topics in Artificial Intelligence: 10th Australian Joint Conference on AI', Springer-Verlag.

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

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

Winograd, T. [1975], Frame representations and the declarative/procedural controversy, *in* 'Readings in Knowledge Representation', Morgan Kaufman, pp. 185–210. Also available R.J. Brachmann and H.J. Levesque (eds), Readings in Knowledge Representation, Morgan Kaufmann, Palo Alto,1985.

Yost, G. [1992], TAQL: A Problem Space Tool for Expert System Development, PhD thesis, Computer Science, Carnegie Mellon.

Yost, G. [1994], Implementing the Sisyphus-93 task using SOAR/TAQL, *in* B. Gaines & M. Musen, eds, 'Proceedings of the 8th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop', pp. 46.1–46.22.

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

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.