Int. J. Human–Computer Studies (1996) **44,** 275–280

Editorial: the Sisyphus-VT initiative

A. Th. Schreiber

University of Amsterdam, Department of Social Science Informatics (SWI), Roetersstraat 15, NL-1018 WB Amsterdam, The Netherlands. email:

W. P. Birmingham

University of Michigan, Department of Electrical Engineering and Computer Science, 1301 Beal Avenue, Ann Arbor, MI 48109–2122, USA

1. Historical background

This special issue contains results of the second ''Sisyphus'' study carried out in the knowledge acquisition community in the period 1992–1995. The Sisyphus initiative arose at the 1990 European Knowledge Acquisition Workshop in Driebergen (The Netherlands). To explain the rationale behind this initiative it is necessary to give a brief synopsis of the recent history of the knowledge-engineering field.

The traditional approach towards knowledge engineering (KE) was that of a ''mining'' activity: knowledge was ''extracted'' from available sources such as text books and experts, and subsequently ''transferred'' to the target system. This view proved to have serious limitations. Firstly, it turned out that it was difficult to maintain the resulting knowledge base. It was not clear when the knowledge base was complete, and introduction of new knowledge pieces (typically production rules) could have unpredictable effects on the problem-solving process. Secondly, the resulting systems were unable to explain their reasoning behaviour in a way that was understandable for humans. Thirdly, it was unclear how one could reuse (parts of) a system for a new application. With the growing complexity of systems, these limitations became more important.

Clancey was one of the first researchers to analyse these shortcomings based on his experiences with the NEOMYCIN system (Clancey, 1983). The limitations of so-called ''first-generation'' knowledge engineering led during the second half of the 1980s to a shift from the ''mining'' paradigm to what has become known as the ''model-based'' knowledge-engineering paradigm. Two features lie at the heart of model-based knowledge engineering:

(1) **Implementation-independent description of knowledge.** A KE methodology should provide a level of description for knowledge that acts as an intermediary between the data provided by experts and text books, and the final representations in an operational system. These intermediate knowledge descriptions should abstract from implementation details, and focus on the content and types of knowledge independent of their computational realization.

This intermediate level of description is close to what Newell (1982) called the ''knowledge level''. Therefore, the KE community has adopted the term ''knowledge-level model'' for these intermediate representations, although

1071-5819/96/3/40275 + 06$18.00/0

strictly speaking some differences exist between Newell's original notion, and some interpretations common in the KE community.

(2) **KE as a constructive activity.** Contrary to the ''mining'' view, the model-based approach views KE as a constructive activity in which a knowledge engineer builds a problem-solving model from the available expertise data. These data act as an important source of information and inspiration, but it is acknowledged that these data are hardly ever complete, and that modelling decisions have to be made by the knowledge engineer in a constructive manner, taking the goals of the target application into account.

By 1990, the research on model-based KE had led to a plethora of approaches, each with their modelling languages, methods, techniques, tools and terminology. Although it was intuitively clear that there existed a set of shared underlying concepts, the different vocabularies made comparisons difficult. The community felt that it was necessary to start a synthetic effort to create a more coherent view on model-based KE. This provided the background for the first Sisyphus study. The method chosen was to distribute a common ''data set'' (i.e. a description of an application domain), and to ask research groups to apply their methods to this data set, thus providing material useful for making detailed comparisons. The data set used in the first Sisyphus study concerned an application domain in which employees had to be assigned to office spaces. The data set included a transcript of a think-aloud protocol showing how an expert solved the problem, plus additional information about rooms and employees in a sample case. The contributions to this first study were published as a special issue of this journal (Linster, 1994).

The first study clarified many issues, in particular with respect to the scope of the differences between approaches. However, it was also felt that the office-assignment domain was not very knowledge intensive and that the application was a ''toy'' one, and that these facts had prevented a full in-depth comparison of the model-based approaches. For this reason, the community decided at the Knowledge Acquisition Workshop in 1992 in Banff (Canada) to conduct a second study in a more realistic and knowledge-intensive application domain. For this purpose, the elevator-design application was chosen. This domain had been the target of the VT application developed by Marcus, Stout and McDermott (1988). A prime reason for this choice was the availability of a meticiously documented description of the original application data, written by Gregg Yost for the purpose of his Ph.D. work at Carnegie Mellon. This special issue reports on the results of this second Sisyphus study.

2. The Sisyphus-VT study

The first ''Call for Contributions'' for Sisyphus-VT was distributed in January 1993. This resulted in eight papers, that were submitted to the Sisyphus-VT track at the Knowledge Acquisition Workshop (Banff, Canada) in February 1994. Subsequently, authors were asked to extend their papers into contributions for this issue. A one-day meeting was held prior to EKAW'94 (Brussels/Hoegaarden, Belgium) to synchronize work on the revised papers. Throughout this process discussions were taking place on a special Sisyphus-VT mailing list.

Data set. The data set of the Sisyphus-VT study consisted of two parts. Firstly, Gregg Yost made his document about the VT application knowledge available. This document contained a 40-page description of the knowledge needed for the design of a certain class of elevators (cable-operated systems driven by an overhead motor assembly). The document describes the required input and output information, types of elevator components and their parameters, numerical and logical relationships between component parameters, knowledge about solving design problems (constraint violations), and a test case. The document is very information-dense. For example, it describes some 300 component parameters. To facilitate access to the document, the PROTÉGÉ group at Stanford University converted this document during the study into a hyper-text format.†

Secondly, Tom Gruber, Greg Olsen and Jay Runkel created an ontology for VT design knowledge, as well as an associated knowledge base (''domain theory'') that contained a large part of the VT domain knowledge in the format prescribed by the ontology. The ontology and knowledge base were made available in the Ontolingua format developed within the Knowledge Sharing Effort.‡ The purpose of this exercise was to provide opportunities for experimenting with knowledge sharing and reuse. Given the size of the VT knowledge base, it was expected to be worthwhile for contributors to try to create a link with the Ontolingua theories, and thus avoid the ordeal of typing in the full domain knowledge base. The use of a predefined ontology was also emphasized in order to ensure that the study would deliver data that provided an optimal basis for comparisons. The pro's and con's of the Ontolingua theories have been heavily debated during the study, as will be clear from the papers in this issue. It should also be mentioned that the Ontolingua theories went through various revisions, including both conceptual restructuring and bug fixes in the knowledge base.

Requirements for contributions. Research groups that wanted to participate in Sisyphus-VT were asked to describe how the VT problem could be solved using their approach. As a minimum, each contribution had to describe both a running system, and a knowledge-level model of the ontology(-ies) and problem-solving method(s) underlying the system. The papers had to cover the following topics.

- The problem-solving method used. If the authors used a method based on the first Sisyphus task (room assignment), it was encouraged that a comparison between the methods be made.
- A description of the ontology used by the problem-solving methods. A minimal requirement was that the relation with the Ontolingua ontology was described. Actual reuse of the ontology was highly encouraged. All examples were expected to follow the terminology provided by the predefined ontology.
- The knowledge-acquisition methods should be fully described.
- Listings of experimental results needed to be included.

Other suggested discussion topics included the following.

- Reusability of the problem solver and knowledge-acquisition tools used.

† http://camis.stanford.edu/protege/sisyphus-2/index.html
‡ http://www-ksl.stanford.edu/knowledge-sharing/ontologies/html/configuration-design.text.html

- Comparisons to the first Sisyphus problem were encouraged. These comparisons could be formal or anecdotal.
- Reflections on the portability of the ontology.

Contributors were given guidelines for structuring their contributions. A typical contribution was expected to contain the following elements.

- **Knowledge-modelling approach.** Description of the general approach taken towards modelling problem solving.
- **Initial problem analysis.** Characterization of the VT problem in terms of the approach, e.g. a first task analysis. Characterization of the KE activities carried out in tackling the problem.
- **Ontology description(s).** Description of the way in which the domain knowledge was structured, including the relation with the Ontologua theories and reuse opportunities,
- **Problem-solving method(s).** Description of the problem solving method(s) (PSMs) that the system applies, including relations/connections with the ontology, reusability aspects, PSM selection criteria, PSM integration issues, etc.
- **Implementation aspects.** Description of how the knowledge-level model was realized computationally, including support tools. Parts of one trace of the system solving the test case needed to be included as well.
- **Evaluation criteria.** Based on the experiences in Sisyphus-I, Gaines and Linster proposed the following set of criteria for evaluating a contribution to Sisyphus-VT.
  (1) Any proposed solution should actually solve the problem, i.e. not just be a problem analysis or domain analysis.
  (2) The solution should be complete and correct in terms of specified criteria, i.e. the problem itself should be sufficiently well-defined for these criteria to be applied.
  (3) The solution should be based on technology that is as reusable as possible. The way that components, methods, ontologies and/or databases are reusable, and how they couple to other parts of the system, should be described.
  (4) The solution should be similar to that of the experts, i.e. the objective is knowledge acquisition rather than just problem-solving. (Remark: no protocol is included in the data set, but the domain description gives some information about how experts structure the VT problem.)
  (5) The computational efficiency of the solution should be evaluated. This is a side-effect of criterion 4. Human experts are not generally computationally intensive.
  (6) The basis of the approach should be made explicit, both the knowledge acquisition methodology and the problem-solving methodology.
  (7) The way in which the solution arises out of the data provided should be made explicit.
  (8) The places where knowledge is represented in the solution should be made explicit.
  (9) The sensitivity of the solution to data changes should be discussed.
  (10) The sensitivity of the solution to problem changes should be discussed.

(10) The sensitivity of the solution to problem changes should be discussed.
(11) The sensitivity of the solution to user intervention should be discussed, i.e. at what points can users intervene in the problem-solving process.
(12) All of the above criteria are recommendations and specific approaches may find them inappropriate, but the reasons for this should be stated clearly and justified.

3. Issue contents

This issue contains seven contributions that describe a solution to the VT problem. Together, these papers comprise a representative set of KE approaches. The first paper by Yost shows how the VT problem is solved using the SOAR/TAQL environment. TAQL is a language on top of SOAR which eases the specification of tasks and methods in a SOAR context. The second paper by Rothenfluh, Gennari, Eriksson, Puerta, Tu and Musen reports on the use of the PROTÉGÉ-II framework and tool set to tackle the VT application. The PROTÉGÉ work is carried out at the Stanford Medical School, but the framework is a general one and not limited to medical applications. The third contribution is from the Open University and the University of Nottingham in the UK. Motta, O'Hara, Shadbolt, Stutt and Zdrahal describe how they used the VITAL methodology and toolkit to solve the VT problem. VITAL is a KADS-based approach developed within the framework of the European ESPRIT program. The fourth paper by Schreiber and Terpstra uses the variant of KADS developed in the ESPRIT project CommonKADS. In CommonKADS there is more emphasis on domain-knowledge modelling than used to be the case in earlier versions. The fifth contribution by Runkel, Birmingham and Balkany shows how the DIDS framework and system was used to solve the VT problem. DIDS is specific for design tasks, and places much emphasis on reusable design descriptions.

The sixth paper by Poeck, Fensel, Landes and Angele uses a combination of two approaches. The knowledge-level model of VT is constructed with the MIKE approach, another descendent of KADS. This model is subsequently used as input for an implementation in an environment based on role-limiting methods, similar to the ones developed by McDermott and colleagues. Finally, the contribution of Brazier, van Langen, Treur, Wijngaards and Willems uses the DESIRE approach developed at the Free University of Amsterdam. Their approach has links with the Generic Task approach developed by Chandrasekaran and colleagues at Ohio State University.

In addition to these seven papers, two other papers have been included. These papers give readers a chance to study the VT data set used by the contributors. The paper by Yost and Rothenfluh contains the original VT document plus some additional tables that ease access to this material. The paper by Gruber, Runkel and Olsen contains the Ontolingua code of the VT design ontologies and representative excerpts from the VT domain theory (the actual knowledge base).

4. Some preliminary conclusions

The Sisyphus-VT endeavor has been useful to the research community. Aside from providing a common problem to calibrate the myriad of research approaches, the

endeavor has begun to foster a common language among researchers. It is possible now, for example, for researchers to deeply probe the work of others. Sisyphus-VT has also fostered a much needed debate about the role of ontologies, what an ontology should contain, and how ''portable'' a portable ontology truly is. Furthermore, a greater understanding of problem-solving approaches has emerged.

On the negative side, most researchers have concentrated mostly on problem solving, and have not considered knowledge acquisition. Since getting a problem solver to work is a prequiste to getting a knowledge-acquisition tool working, emphasis on problem solving is natural. We hope that the next round of Sisyphus will encourage more research on knowledge acquisition.

The Sisyphus-VT study would not have been possible without the help and support of many people. The study profited from the detailed VT document provided by Gregg Yost. The subsequent enhancements for accessing this document provided by Thomas Rothenfluh and John Gennari were extremely useful. Tom Gruber, Jay Runkel and Greg Olsen created the Ontolingua theories that gave this study an extra dimension which proved extremely fruitful. Several contributors provided useful feedback on the Ontolingua theories through bug reports and change requests for the ontology.

Marc Linster laid the foundations for this work through his successful effort to turn some loose ideas about Sisyphus expressed at EKAW'90 into a real study. Brian Gaines has supported and promoted the Sisyphus initiative right from the start. Brian Gaines, Tom Gruber, Georg Klinker, Marc Linster, Mark Musen, and Rudi Studer participated in setting up the original call for contributions. Sandra Marcus came to KAW'94, allowing us to profit during the Sisyphus-VT discussions from her extensive knowledge of the application domain. Also thanks to all the participants of the respective KA workshops on which the Sisyphus-VT issues were discussed. Those discussions were usually insightful and added to the general positive feeling about this whole enterprise.

References

Clancey, W. J. (1983). The epistemology of a rule based system—a framework for explanation. *Artificial Intelligence,* **20,** 215–251.

Linster, M. (1994). Sisyphus'91/92: Models of problem solving. *International Journal of Human–Computer Studies,* **40,** 189–192.

Marcus, S., Stout, J., & McDermott, J. (1988). VT: an expert elevator designer that uses knowledge-based backtracking. *AI Magazine,* Spring, 95–111.

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