Structuring and modules for knowledge bases: motivation for a new model

Grigoris Antoniou and Ipke Wachsmuth*

Evolving out of theoretical and practical work, the paper presents the motivation and basic ideas for the construction and use of modular knowledge bases. The approach relates to earlier work carried out by each of the two authors of the paper separately. A model is introduced that merges the two previous approaches, modules for logical knowledge bases, and ordering by generality domains, while maintaining their benefits. Central aims are reusability, the restriction of memory searching, and the management of inconsistent (competing) knowledge within one knowledge base. The model is explained using examples, and the formal semantics are discussed of structured, modular knowledge bases for knowledge representations that are based on logic programming.

Keywords: knowledge representation, knowledge sharing, modules

As artificial intelligence (AI) technology is moving towards more ambitious applications, the development of large-scale, knowledge bases has become one of the most challenging tasks (see, for example, Reference 1). We believe that in order to manage such knowledge bases *structuring* is essential.

- In large knowledge bases it is necessary to restrict the search space of deduction *by way of principle* and not simply on the basis of heuristics.
- Structured knowledge bases provide a possibility for managing inconsistencies by considering partitions that can be selectively accessed. In this case, it is possible to have alternative views on a knowledge base leading to context-dependent answers.
- Finally, structuring makes knowledge bases more easily comprehensible and maintainable, an important task in any large system where several developers and users are involved in the assimilation of increasingly many knowledge items[2].

The model we outline here is a combination of earlier work carried out independently by the two authors of this paper:

- A model for structured knowledge bases (ordered by generality domains) developed by the second author[3,4], whose basic ideas are grounded on findings from empirical research about how human knowledge is structured. The findings suggest that a major feature of human intelligence lies in focusing on a part of the knowledge that is small enough to be tractable. If a problem cannot be solved in a satisfactory way, other (perhaps competing) parts of the knowledge must be tried.
- A theory of modularity for logical knowledge bases is presented in References 5 and 6. According to this approach, motivated by work in algebraic specification[7], modules are independent entities communicating with their environment via their interfaces. One of the benefits is local verification.

NEW MODEL: INTUITION

Let us start with an example (see *Figure 1*) that demonstrates our idea of combining the approaches presented above. It describes the knowledge that one needs to determine one's behavior when one is downtown.

A structured knowledge base consists of a number of modules; these are parts of knowledge closely (semantically) related to each other and defining some specific, self-contained part of the entire knowledge. The modules are equipped with import and export interfaces that would ideally give a full description of the knowledge exported or imported (thus playing a role similar to that

Fachbereich Mathematik/Informatik, Universität Osnabrück, Albrechtstrasse 28, D-4500 Osnabrück, Germany
*Technische Fakultät, Universität Bielefeld, Universitätsstrasse 25, D-4800 Bielefeld 1, Germany
Paper received 16 August 1993. Accepted 8 October 1993

0950-7051/94/010049–03 1994 Butterworth-Heinemann Ltd

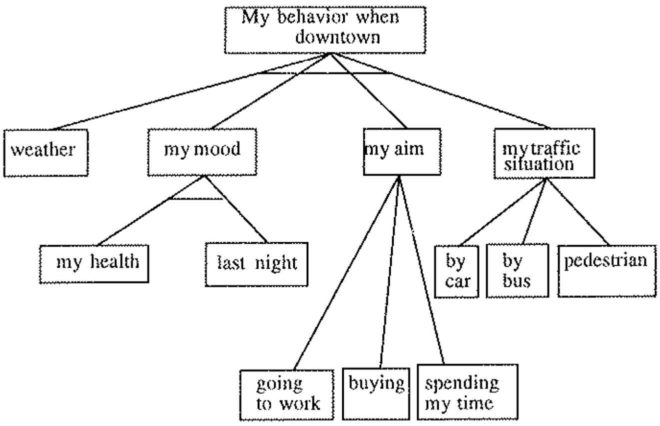

Figure 1 Example

of abstract data types in conventional computer science). Unfortunately, such a complete specification of AI systems is usually impossible. In such cases, the export (or import) interface describes the signature of exported (or imported) knowledge (like interfaces in imperative programming languages) and some *integrity* (*consistency*) *conditions* that the exported (or imported) knowledge should satisfy. The presence of formal interfaces and module semantics allows the use of formal verification methods in order to show that a module matches the requirements.

Note that, in some connections, the 'lower' modules contain knowledge about different aspects (for example the top connection), whereas in others they contain competing knowledge items that exclude each other (for example, the module *going to work* could contain in-hurry, while *spending my time* could include ¬in-hurry). We distinguish the two kinds of module connection below.

AND-connection states that the modules from which the top one can import do not contain competing knowledge, but rather information on different topics of the modeled domain. (Whereas it is often intuitively clear what competing knowledge means, it is difficult to give a general, formal definition; it is up to the knowledge engineer to decide). This means that the knowledge of all these modules (or of some of them) may be used at the same time.

Note that, according to the new model, knowledge of lower modules is not always visible to higher modules, but rather only when it is needed (according to the current focus — see below).

The *OR-connection* of modules indicates that the modules on the lower level contain competing knowledge. In this case, only one of these modules may be visible at a given time. However, note that any such module may be AND-connected to other modules in a subsequent level. Finally, note that a module needed in distinct OR-connected knowledge parts may be shared.

The meaning of a structured, modular knowledge base is defined with respect to a *current focus*. This focus defines a current view on the knowledge base and must be such that competing parts of knowledge are not visible at the same time. In the example of *Figure 1*, a focus could consist of the modules *by car*, *going to work*, and *last night*. Then, these modules and all the modules above them are visible at the moment, i.e. their knowledge can be used. Note that, for each OR-connection, at most one module can be included in a current focus. For each AND-connection, none, one, some, or all the modules involved (of the lower level) may be included. In our example, we have not included *my health* in the focus. It could be the case that I slept badly last night, and so even my good health cannot prevent my mood from being bad. The definition of possible focuses is given inductively as follows:

- $\{M\}$ is a possible focus, where M is the top module in the hierarchy.
- If F is a current focus, M' is an element of F, and M' is AND-connected to $M_1, \ldots, M_n$, then replace M' in F by an arbitrary subset of $\{M_1, \ldots, M_n\}$. The resulting set is an admissible focus.
- If F is a current focus, M' is in F, and M' is OR-connected to $M_1, \ldots, M_n$, then replace M' in F by some M_i from $M_1, \ldots, M_n$. The resulting set is an admissible focus.

All the modules that are above some element of the current focus are visible. Note that the definition of admissible focus is such that competing knowledge cannot be visible at the same time. Furthermore, it is easily verified that the focus example above is in accordance with this definition.

Obviously, it is unreasonable to demand global consistency of a structured, modular knowledge base. Instead, only knowledge items that can be active at some time need to be consistent with each other. Following Reference 3, we call this the *local consistency requirement*.

FORMAL DESCRIPTION OF MODEL

In this section, we briefly introduce the semantics of structured knowledge bases in the setting of logic programming[8,9]. The body of a module M is a logic program (possibly with negation) kb(M). The interfaces exp(M) and imp(M) of M contain the predicates that are imported (or exported). As usual in logic programming, we regard the constants and function symbols as being global (this restriction is for the sake of simplicity only). As our idea is that knowledge about some predicates are imported from other modules, we protect imported knowledge by demanding that kb(M) is conservative with respect to imported predicates (meaning that imported predicates do not occur in heads of rules in kb(M)) (see Reference 5 for more details).

The meaning of a stand-alone module is determined by the facts $p(t_1, \ldots, t_n)$ with an exported predicate p that follow from the completion of kb(M) and the imported knowledge (if there is any — this will depend on the current focus).

The combination of modules (either by OR- or AND-connection) is carried out in such a way that predicates exported by the lower modules can be imported by the top module of the connection if these predicates also appear in its import interface. Of course, more flexible methods of combination are possible, for example *signature morphisms*[10], allowing the renaming of predicates and noninjective mappings. We disregard this possibility for the sake of simplicity. The semantics of a structured knowledge base S is given relative to a current focus F. If S is a single module, then its semantics has already been described as comp(kb(M)) $\cap$ exp(M), where comp is the logical completion operation.

Let M be the top module of S, and suppose that it is related to the structured knowledge bases $S_1, \ldots, S_n$ by an AND-connection. Of these, let $S_{i_1}, \ldots S_{i_k}$ be the structured knowledge bases containing a member of the current focus. Then define Export(S), the exported knowledge of S (always with respect to the focus F) as follows:

$$[\mathrm{comp}(\mathrm{Import}(M,F) \cup \mathrm{kb}(M))] \cap \exp(M)$$

where

$$\mathrm{Import}(M,F) = [\mathrm{Export}(S_{i_1}) \cup \ldots \cup \mathrm{Export}(S_{i_k})] \cap \mathrm{Imp}(M)$$

In particular, if $M \in F$, then the exported knowledge of S is comp(kb(M)) $\cap$ exp(M). In the case of an OR-connection at the top level, the definition of Export(S) is as above, the only difference being that (by the definition of the possible foci) only one subsystem from $S_1, \ldots, S_n$ can include members of F.

CONCLUSIONS AND FUTURE WORK

We have introduced a new model for structuring knowledge bases and indicated its usefulness in practice. The model combines the advantages of two previous approaches and addresses the main requests associated with modularity (restriction of search space, maintainability and reusability) as well as the additional requirement of managing competing knowledge within one knowledge base. It is compatible with experimental findings on human intelligence while also addressing engineering problems.

One problem that we have left out of the paper is that of determining the appropriate focus, i.e. the parts of the knowledge that are relevant to the current problem. We think that this question lies at the heart of intelligent behavior. Up until now, there have only been some practical solutions for special cases (for example keyword-based access in text understanding problems), but no generally applicable theory. In a government-funded three-year research effort that has just begun at the University of Bielefeld, Germany, a modular medical knowledge base for hypertension consultation will be developed. On the basis of the experiences gained in this practical work we will further pursue the point of focus management.

REFERENCES

1 Herzog, O and Rollinger, C-R (Eds.) *Text Understanding in LILOG* Springer-Verlag (1991) (Lecture Notes in Artificial Intelligence 546)
2 Neches, R, Fikes, R, Finin, T, Gruber, T, Patil, R, Senator, P and Swartout, W R 'Enabling technology for knowledge sharing' *AI Magazine* Vol 12 No 3 (1991) pp 36–56
3 Wachsmuth, I 'Zur intelligenten Organisation von Wissensbeständen in künstlichen Systemen' *IWBS-Report 91* IBM Deutschland, Germany (1989)
4 Wachsmuth, I and Gängler, B 'Knowledge packets and knowledge packet structures' *in* Reference 1
5 Antoniou, G 'Modularity for logical knowledge bases' *Proc. 4th International Conference on Software Engineering and Knowledge Engineering* IEEE Press (1992)
6 Antoniou, G 'Logical approaches to structured knowledge bases' *Proc. Artificial Intelligence — Methodology, Systems, Applications* (1992)
7 Ehrig, H and Mahr, B *Fundamentals of Algebraic Specification — Vol 2* Springer-Verlag (1990)
8 Lloyd, J W *Foundations of Logic Programming* (2nd Ed.) Springer-Verlag (1987)
9 Sperschneider, V and Antoniou, G *Logic: A Foundation for Computer Science* Addison-Wesley (1991)
10 Ehrig, H and Mahr, B *Fundamentals of Algebraic Specification — Vol 1* Springer-Verlag (1985)