

STRUCTURING AND MODULES FOR KNOWLEDGE BASES: MOTIVATION FOR A NEW MODEL

Grigoris Antoniou¹, Ipke Wachsmuth²

¹Universität Osnabrück

Albrechtstrasse 28, D-4500 Osnabrück, Germany

E-mail: ga@informatik.uni-osnabrueck.de

²Universität Bielefeld

Universitätsstrasse 25, D-4800 Bielefeld 1, Germany

E-mail: ipke@techfak.uni-bielefeld.de

Abstract: Evolving out of theoretic and practical work, this paper presents the motivation and basic ideas for the construction and use of modular knowledge bases. The approach bears upon earlier work by the two authors done separately from each other. We introduce a model which attempts to merge two previous approaches while maintaining their benefits: Modules for logical knowledge bases, and ordered by generality domains. Central aims are reusability, restriction of memory search, and management of inconsistent (competing) knowledge within one knowledge base. We explain the model by motivational examples and discuss the formal semantics of structured, modular knowledge bases for knowledge representations that are based on logic programming.

Keywords: Knowledge representation in the large, knowledge sharing, modules

Introduction

As artificial intelligence technology is moving towards more ambitious applications, the development of large-scale knowledge bases has become one of the most challenging tasks (e.g. [5]). We believe that in order to manage such knowledge bases structuring is essential:

- o In large knowledge bases it is necessary to restrict the search space of deduction by way of principle and not simply based on heuristics.

- o 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.

- o 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 [7].

The model we outline here is a combination of earlier work by the two authors done independently from each other:

- o a model for structured knowledge bases (ordered by generality domains) developed by the second author [9, 10] 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 focussing on a part of the knowledge small enough to be tractable. In case a problem cannot be solved in a satisfactory way, other (perhaps competing) parts of the knowledge must be tried.

- o a theory of modularity for logical knowledge bases is presented in [1,2]. According to this approach, modules are independent entities communicating with their environment via their interfaces. One of the benefits is local verification.

The new model: Intuition

Let us start with an example (Figure 1) that demonstrates our idea of combining the approaches presented above. It describes the knowledge I need to determine my behavior when being 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 of abstract data types in conventional computer science). Unfortunately, such a complete specification of AI systems is usually impossible. In such cases, the export (resp. import) interface describes the signature of exported (resp. imported) knowledge (like interfaces in imperative programming languages) and some integrity (consistency) conditions the exported (resp. imported) knowledge should satisfy. The presence of formal interfaces and module semantics allows usage of formal verification methods in order to show that a module matches the requirements.

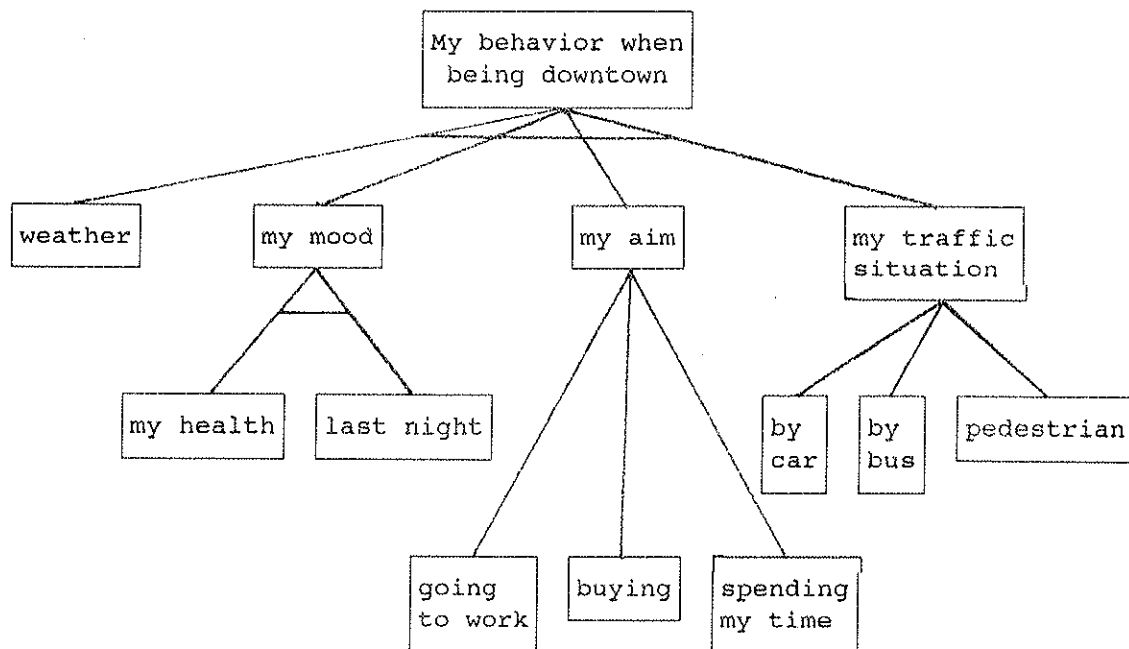


Fig.1

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

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 (Remark: 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 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. But note that any such module may be AND-connected to other modules in a subsequent level. Finally, let us 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 modules above them are visible at the moment, i.e. their knowledge may 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 involved modules (of the lower level) may be included. In our example here, we have not included my health to the focus. It could be the case that I have slept bad tonight, so even my good health cannot prevent my mood from being bad. The definition of possible focuses is given inductively as follows:

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

- o If F is a current focus, M' is in F , and M' is OR-connected to M_1, \dots, M_n , then replace M' in F by some M_i from M_1, \dots, M_n . The resulting set is a admissible focus.

All 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 respects 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 to each other. Following [9], we call this the local consistency requirement.

Formal description of the model

In this section we briefly introduce the semantics of structured knowledge bases in the setting of logic programming [6]. 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 resp. exported. As usual in logic programming, we regard the constants and function symbols as being global (this restriction is made 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 w.r.t. imported predicates (meaning that imported predicates do not occur in heads of rules in $kb(M)$); see [1] for more details.

The meaning of a module, stand alone, is determined by the facts $p(t_1, \dots, t_n)$ 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 as we shall see in a moment).

Combination of modules (either by OR- or by AND-connection) is done 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 ways of combination are possible, for example signature morphisms [3] allowing renaming of predicates and noninjective mappings. We disregard this possibility here 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 $\text{comp}(\text{kb}(M)) \text{?exp}(M)$, where comp is the logical completion operation.

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

$[\text{comp}(\text{Import}(M, F) \langle \text{union} \rangle \text{kb}(M))] \langle \text{intersection} \rangle \text{exp}(M)$, where $\text{Import}(M, F)$ is $[\text{Export}(S_{i_1}) \langle \text{union} \rangle \dots \langle \text{union} \rangle \text{Export}(S_{i_k})] \langle \text{intersection} \rangle \text{Imp}(M)$.

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

Conclusion and future work

We have introduced a new model for structuring knowledge bases and indicated its usefulness in practice. The model combines 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 we have completely left out in the present paper is that of determining the appropriate focus, i.e. the parts of knowledge relevant for the current problem. We think that this question lies at the heart of intelligent behavior. Until now there are only some practical solutions for special cases (for example keyword-based access in text understanding problems), but no generally applicable theory. In the context of a government-funded three-year research effort just begun at the University of Bielefeld, a modular medical knowledge base for hypertension consultation will be developed. Based on the experiences gained in this practical work we will further pursue the point of focus management.

References

1. Antoniou G.: Modularity for Logical Knowledge Bases, Proc.SEKE-92, IEEE Press 1992
2. Antoniou G.: Logical Approaches to Structured Knowledge Bases, Proc. AIMS-92
3. Ehrig H. and Mahr B.: Fundamentals of Algebraic Specification Vol. 1, Springer 1985
4. Ehrig H. and Mahr B.: Fundamentals of Algebraic Specification Vol. 2, Springer 1990

5. Herzog O. and Rollinger C.-R. (eds.): Text Understanding in LILOG, Lecture Notes in Artificial Intelligence 546, Springer 1991
6. Lloyd J.W.: Foundations of Logic Programming 2. ed., Springer 1987
7. Neches R., Fikes R., Finin T., Gruber T., Patil R., Senator P. and Swartout W.R.: Enabling Technology for Knowledge Sharing, AI Magazine 12,3, 36-56
8. Sperschneider V. and Antoniou G.: Logic: A Foundation for Computer Science, Addison-Wesley 1991
9. Wachsmuth I.: Zur intelligenten Organisation von Wissensbeständen in künstlichen Systemen, IWBS-Report 91, IBM Deutschland, Stuttgart/Heidelberg 1989
10. Wachsmuth I. and Gangler B.: Knowledge Packets and Knowledge Packet Structures, in [5]
11. Wirth N.: Programming in Modula-2, Springer 1983