Knowledge Packets and Knowledge Packet Structures

Ipke Wachsmuth, Barbara Gängler

1 Introduction

One of the most challenging tasks in knowledge representation for text-understanding systems is the development of large-scale knowledge bases containing semantic background knowledge. Many current knowledge-based systems restrict their applicational scope to narrow world domains, that is, they are equipped to handle domain-specific tasks by using domain-specific facts and rules. However, for the domain of text understanding this restriction appears to be unacceptable since the understanding of natural-language texts draws on a large background of diverse world knowledge and belief. Even if a knowledge-based system could be provided with a large body of world knowledge sufficiently rich to cover a broader domain of discourse, the problem of *using* this knowledge to build and utilize representations for natural language inputs might prove a serious one with respect to computational resources.

The experiences gained in the development of a LILOG text understanding system led to the conclusion that large-scale knowledge bases are unavoidable when more realistic application domains are modeled. In particular, the authors were involved in constructing the background rule knowledge base for the Düsseldorf tourist guide domain realized in the second LILOG experimental environment, LEU/2. Building upon about 600 sort declarations, this knowledge base incorporates about 300 axioms (rules) serving for different purposes. On the one hand, the information extracted from a natural language text by means of parsing and constructing a semantic representation (explicit text knowledge) needs to be elaborated and enriched by read-time inferences. On the other hand, question-time inferences are necessary in order to generate answers to queries by which the success of text understanding is demonstrated (Gängler, Wachsmuth 1991).

Some of the rules in the background knowledge base are *performance rules* which achieve the re-representation of situation-(verb-)centered semantic text representations by object-centered representations so that information about objects of interest can be more readily retrieved. Further performance rules were incorporated to give rise to a more flexible choice of wordings in queries. For example, it should be equally possible to ask at which date the construction of a church was "finished" or when it was "completed." Other rules are *elaborative rules* which make implicit text information explicit by enriching it through commonsense or specific domain knowledge. While in some cases longer chains of inference are necessary to achieve a coherent text representation, elaborative rules often bridge very short gaps. For example, when the text mentions that a particular church has a crooked spire, then the system should be able to infer that this church in fact has a spire (which is not true for all churches).

Since performance and elaborative rules in many cases perform tiny inference steps, the background knowledge base must include a quite enormous number of rules even when the scenario is relatively restricted. It was observed that the size of search spaces to be managed by the system results not so much from the depth of proofs to be performed, but results rather from the amount of memory search necessary when deeper text representations are constructed or when a query is answered. Even though an attempt was made to enhance the efficiency of the system by tailoring the rules, costly memory search often could not be avoided. More significantly, the more rules are incorporated, i.e. the more "knowledgeable" the system becomes, the larger becomes the body of knowledge elements to be screened.

The question arises whether there is a means for excluding parts of the axiomatic knowledge for certain inference tasks *by way of principle*. The experiences in furnishing domain-specific rules for the Düsseldorf domain showed that in many cases it was clear for which specific subaspect of the domain the particular rule was invented. In turn, it was clear in many cases which domain segment was relevant for finding an answer to a given query. Hence the set of possible rules of inference could be restricted if the relevant domain segment could be identified when processing a piece of text or a query. As a prerequisite for such a selective memory search procedure, the rule inventory needs to be structured in a modular fashion. It should be possible to build up and use portions of a knowledge base that pertain to subareas or subaspects of the general discourse domain. As far as possible these portions should be independent units which are organized in a way making access possible according to actual needs. That is, besides being concerned about the content of a background knowledge base, its *structure* needs particular attention as well as the *conditions for using* its knowledge.

This paper presents the general ideas which gave rise to our proposing modules of knowledge called *knowledge packets* and for specifying *knowledge packet structures* which establish usability conditions. To avoid any confusion it is noted that "module" is meant to refer to a component of a knowledge base, and it should not be confused with the notion of specialized systems dedicated to sensory and linguistic input analysis and to linguistic and motor output as in (Fodor 1983) and (Garfield 1987). Further it should be understood that "module" does not refer to a single propositional statement (fact or rule) but rather to a set of statements that "belong together" in a certain respect. The basic idea of knowledge packets was first introduced in (Wachsmuth 1987). It is grounded on findings from empirical research which suggest that the domain knowledge accumulated by humans is grouped in such a way that subbodies perceived to be relevant in a given situation can be separated out. These findings were elaborated toward an abstract specification for a modular knowledge representation scheme that provides a clear and formal basis for handling problems of consistency and accessibility of domain-specific knowledge (Wachsmuth 1989b).

In the following section we discuss a number of demands to be recognized for the conception of modular knowledge bases. In the third section a brief summary of an empirical study is given from which basic ideas of human knowledge organization can be drawn. As the main part of this paper, the fourth section presents principles of domain-oriented knowledge structuring (the "DOKS principles") which were elaborated from the empirical findings. The core ideas of how these principles can be related to modular knowledge base development and some implications are discussed in the final section.

2 Modular Knowledge Bases

The need for explicit consideration of large and diverse knowledge bases and the question of how to modularize them is finding increasing attention (Wachsmuth 1989a; Prerau et al. 1990; Wachsmuth, Meyer-Fujara 1990; Pletat 1991). The question of how to grasp a module concept for a knowledge representation language is rather straightforward, though a number of technical questions remain to be dealt with (cf. Pletat, in this volume). Principles from the area of formal software specification suggest that each knowledge base module consist of an export and an import interface and a module body. A mathematically precise definition of the syntax and semantics of L_{LILOG} which includes the basis for such module definitions is given in (Pletat 1991). As will be motivated below, the semantics of these module definitions should imply that the modules of a modular knowledge base are basically autonomous knowledge bases which may even deliver different answers to the same logical query, depending on the actual context or view.

While categories of classical software engineering do seem relevant in constructing a large knowledge-based software system, fundamental questions need to be answered that concern the question of how knowledge (not programs) can be perceived in a modular fashion. There are many possible ways in which particular modules may be interrelated in terms of exporting and importing knowledge among each other, and the question has not been solved on principle in which way a knowledge base in a given application domain is to be structured in modules. The modularity of a domain knowledge base cannot be defined solely in a way that is convenient for software development. The question is whether principles can be found, on the knowledge level, which give rise to a modular conception of a body of knowledge and which can guide the construction of modular knowledge bases. Such principles should account for quite a diverse set of demands as listed below:

- **(Alternative) views on a knowledge base:** It should be possible to consider partitions of a background knowledge base that can be asserted selectively and by this support context-dependent answers (e.g., Hendrix 1979);

- **Elaboration tolerance:** When a knowledge base is elaborated through adding facts, formal reasoning gets slower because more axioms need to be screened. In contrast, human reasoning does not seem to undergo significant decrease in speed by the accumulation of more knowledge. John McCarthy made the point that, ideally, a reasoning process should not be altered or grow in length much when facts that should not affect that process are added to a knowledge base (cf. Lifschitz 1989);

- **Toleration of inconsistencies:** Deductive problem solving requires knowledge bases to be logically consistent. However, humans can successfully deal with their knowledge even though inconsistencies are possible and probably unavoidable (Minsky 1981);

- **Restriction of consistency proofs:** The question has been raised whether for broad knowledge bases the requirement of global consistency can be weakened such that consistency proofs are restricted to portions of a knowledge base that potentially interact during a problem-solving process (Lenat, Feigenbaum 1987);

- **Locality of reasoning:** When formalizing commonsense reasoning by including default information, McCarthy recommends that as much of the reasoning as possible should involve small numbers of facts, i.e., be kept "local" (cf. Lifschitz 1989);

- **Reduction of memory search:** With domain knowledge bases becoming richer, the problem of retrieving relevant facts from an excessive variety can hardly be ignored, for it involves the need to prevent the search from a possible combinatorial explosion (Minsky 1981);

- **Restricted reason maintenance:** When the validity of a conclusion can be traced to a limited set of facts, the expense of reason maintenance could possibly be reduced by recording dependencies with respect to that limited set instead of to the particular facts (to the authors' knowledge no such proposal has yet been made);

- **Transparency and maintainability:** An important aim is to make large knowledge bases more easily comprehensible and maintainable, in particular, when several developers are involved or when maintenance is to be carried out by users (e.g., Prerau et al. 1990). The point is especially relevant when different knowledge sources are incorporated in a knowledge base.

Thus, beside increasing efficiency by controlling inferences, there are a number of further reasons why a modular knowledge base design seems desirable. While the above points are not in all cases independent from one another, they illustrate different aspects to be kept in mind when thinking about design principles for modular knowledge bases.

3 The Empirical Basis

The ideas presented in this paper are grounded on findings from empirical research which gave rise to a basic understanding of how the knowledge that human beings possess is structured. As human beings have proven to take good advantage of quite an impressive body of knowledge, it seems well-motivated to observe findings from cognitive research. For example, Newell and Simon (1972) have used protocol analysis in order to elucidate general heuristics of intelligent problem solving. Rosch (1978) and Dahlgren (1985) have done empirical analyses of cognitive categories that can help to design ontologies for natural language understanding. The rationale for the research exploited here is that the observation of human intelligent behavior can help to find cognitive principles of knowledge organization. The study and its findings are described in detail in (Wachsmuth 1989b). Here we just give a rough sketch of what was done.

In a one-year clinical teaching experiment on the acquisition of mathematical knowledge (Post et al. 1985), a large body of interview data was acquired and analyzed to observe the genesis of domain-specific knowledge structures. During the whole period, the gradual development of selected students' ability to answer questions and deal with problems involving fractions was recorded. No propositional "products" such as rules were taught to the students. Instead, there was ample occasion for them to make observations and gain experience about how concepts in the chosen domain are interrelated. In the course of the study, students showed a strong tendency to abstract observations about the interrelation of concepts in conditional statements (rules). These were accumulated to the repertoire of things students know about, and know how to perform with, the conceptual entities in the field. Such rules were considered as representing belief particles making up the competence of individual subjects, as a potential for generating action (cp. Newell 1982).

For selected topics, the repertoire of rules subjects possessed by the end of the teaching experiment was identified from the interview data. As was evident from subjects' verbal explanations, they would recognize the applicability of a rule by matching rule conditions with task information. With respect to the questions addressed in this paper, the following issues were of interest: How is such a repertoire of domain-specific knowledge utilized in applied situations? Are all rules always attempted? Is full use of all resources always made? If there are any restrictions, what are their features and characteristics? Can they serve to extract general principles comprising a model of a modular representation of domain-specific knowledge? Of course, there is no way to observe structure in human knowledge other than through inferring it from subjects' behavior. To this end, several complex problem-solving tasks were presented toward the end of the teaching experiment. For success with these tasks, a coordinated use of rules from the repertoire the subjects had acquired was necessary.

In summary, evaluation of behavioral data from task-based interviews gave a strong indication that performance differences across subjects depended not only on the soundness of their rules but also on the way their rules appeared to be accessible. The crucial observation is that such pieces of domain-specific knowledge are not assembled as an unstructured collection but that larger structures evolve that reflect the context in which these rules are abstracted. Rules appear to be grouped in a way enabling subjects to separate out subbodies of domain-specific knowledge they perceive as relevant for use in a given task

situation. By way of extrapolation, this observation could contribute to understanding the general problem-solving ability of human beings, namely, from their ability to access appropriate subbodies of knowledge based on clues from a task situation. It may also explain how one can act on the basis of an enormous repertoire of knowledge without getting confused by having to face all of it most of the time.

The findings of the study suggest that a critical feature of human intelligence lies in a dynamic partitioning of the total knowledge into "visible" and "invisible" parts such that the visible part is normally small enough to be tractable. Further findings show that the presence of certain concept words in a situation seems to trigger access to domain knowledge and further vocabulary associated with this knowledge. In turn, domain- and situation-specific word meanings seem to be mediated by association of concept words with portions of domain-specific knowledge.

Without going into details, the next section elaborates general principles characterizing those observations that seem relevant for structuring domain-specific knowledge. These principles aim at devising a way that allows large amounts of domain-dependent knowledge to be used by a knowledge-based system while keeping the system manageable. Each of the nine principles is motivated by certain empirical observations. Altogether, the principles are understood as an abstract specification rather than a symbol-level description for a representation scheme.

4 Domain-Oriented Knowledge Structuring

The central point in our proposal is that knowledge be structured in a way oriented at the domain. More precisely, admitting that much and important intellectual capability can be credited to domain-specific knowledge (Feigenbaum 1977), domain specificity is made the explicit target of structuring. This refers to different degrees of specificity as well as to different aspects in which a specific body of knowledge may branch to extend into specific subbodies. The principles of domain-oriented knowledge structuring, for short, the *DOKS principles* discussed in this section, are of a wider scope than solely for the area of natural language systems. However, for the reasons mentioned in the introduction, they are particularly relevant for natural language understanding and processing, and throughout this section we have chosen examples from this application area for illustration.

Adopting Newell's slogan equation, REPRESENTATION = KNOWLEDGE+ACCESS (Newell 1982) we will have to concern ourselves with (1) the organization of a body of knowledge "in a form that can be used to make selections of actions in the service of goals" and (2) the way particular parts of a structured body of knowledge are accessed such that they "can be used by the larger system that represents the knowledge about goals, actions etc." (all quotes see Newell 1982, p.114). We assume the knowledge base (KB) of a knowledge-based system (KBS) to be constituted by a set of identifiable statements each of which interrelates domain-specific concepts and asserts something held for true in a modeled world. Thus, we consider a collection of entities like logical sentences or production rules or any means-ends relations as givens which hereafter will be referred to by the term *knowledge elements*. A knowledge element could also be a structured term (e.g., a sort declaration with features and roles) describing an entity or a class of entities to which statements in a KB refer.

Overall, the DOKS principles specify a representational scheme for a KBS *abstractly*, without any commitment to particular programs. Additional structuring features of a syntactic nature might be brought in by a particular implementation, e.g., when representing a KB as a connection graph of first-order logical sentences (Kowalski 1975) or when using a restriction strategy like set-of-support for efficient theorem

proving (Wos et al. 1984). As pointed out by Levesque and Brachman (1986) in elaborating on Newell's (1982) recommendations, such decisions concern an entirely separate issue that has been widely investigated in automated theorem proving.

The first three principles pertain to the way the knowledge elements are *organized* in a static knowledge base. This arrangement is viewed as permanent until an explicit change of the KB takes place. The other principles concern the way in which structured knowledge is *accessed* and made available to the knowledge-based system.

4.1 Organization of Knowledge

Principle of packing knowledge elements. Collections of knowledge elements that pertain to a specific domain of knowledge are comprised in a *packet*. We say the packet *owns* these knowledge elements. A packet may properly contain further packets of knowledge elements that constitute identifiable subbodies of more specific knowledge within the outer packet.

Principle of competitive knowledge. Collections of knowledge elements that concern alternative methods or views in a given domain of knowledge are packed separately within the surrounding packet. Such packets are referred to as *competitive*.

Principle of local consistency. The collection of knowledge elements in one packet must not permit conclusions that are contradictory (or actions that are incompatible). A packet P may only contain contradictory (or incompatible) knowledge elements if they are packed separately within P. A collection of knowledge elements satisfying this principle is called *locally consistent*.

The principle of packing knowledge elements is visualized in Figure 1. Within the most general packet P1, the packet P3 (or likewise P2) comprises more specific knowledge contained in P1 (thus, owned by both P3 and P1), with P4 being still more specific than P3. Consider for example P3 containing general knowledge about restaurants such as attributed features, opening-hours or date-of-origin. P4 might then contain specific knowledge about a special kind of restaurant, say, taverns. In general, the decision about what knowledge is considered more specific is a separate issue that involves domain modeling heuristics. A criterion for a knowledge element k to be "more specific" than those in a collection of knowledge elements $k_1,..., k_n$ could be that a reasonable body of problems can be dealt with without using k. "More specific" knowledge might also concern the question of how the more general knowledge around it is adapted to certain contexts.

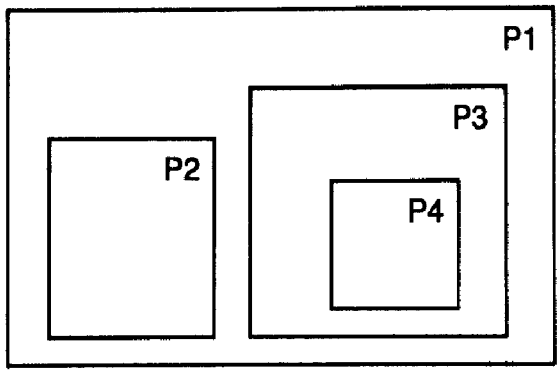

Figure 1

The intent of the principle of competitive knowledge is to make only one alternative, at a given time, available for use by the KBS as will be provided for by further principles describing static access conditions. In Figure 1, P2 and P3 (but not P3 and P4) depict competitive knowledge packets. For example, P2 may contain knowledge about sights and objects of interest whereas P3 could contain some information about restaurants.

The principle of local consistency reflects the fact that a method devised to guarantee global consistency can prove computationally intractable. The intent is to restrict consistency checks to packed subcollections of knowledge elements to be performed at any incremental augmentation of a KB. It would also allow "alternative worlds" to be modeled within a KB. According to this principle, P2 and P3 in Figure 1 may include inconsistent knowledge elements among them, but not so P3 and P4.

Imagine that P3 contains some knowledge about traffic, including information about streetcars as moving objects on which people are conveyed to a certain schedule, etc. In contrast, P2 might contain knowledge about objects of interest, including information about a streetcar as a *non-moving* object of historical and technical interest that could be seen in a museum. Conflicting implications would be possible depending on whether a streetcar is considered in Packet P2 or P3. The principle of local consistency tolerates such global inconsistencies as long as they are captured in separate knowledge packets.

The three principles of knowledge organization describe how a collection of knowledge elements can be structured by way of set containment. Though such a structure cannot be directly observed from the empirical findings, the structural principles are compatible with what was gleaned from the empirical data. We now proceed to describe how knowledge in a KB satisfying the above structuring principles is accessed. There will be static and dynamic access conditions.

4.2 Static Access Conditions

Principle of eligibility of knowledge elements. The knowledge elements owned by a packet P are conjointly eligible for use by the knowledge-based system when their packet P, or a packet within P, is tagged ACCESSED, *but only as far as they are not also owned by a packet contained within the one tagged* ACCESSED. We say a knowledge element (or a set of knowledge elements) eligible for use is VISIBLE. No knowledge elements packed separately from the packet tagged ACCESSED are eligible.

Principle of single access to packed knowledge. Only one packet at any given time may be tagged ACCESSED.

Principle of reachability of knowledge. When a knowledge packet P tagged ACCESSED owns knowledge elements that are also owned by a packet Q within P, then the set of knowledge elements in Q (or for simplicity, the packet Q) is REACHABLE. A collection of knowledge elements packed separately from the one tagged ACCESSED is NOT REACHABLE.

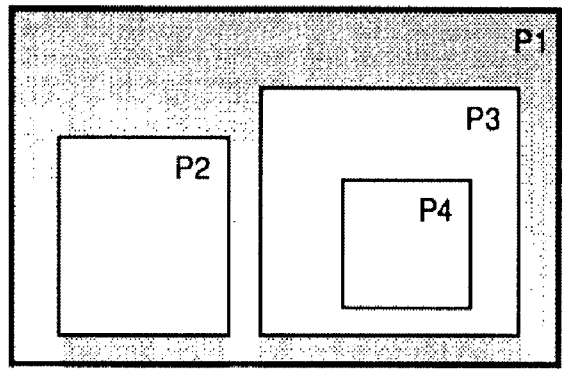

Figure 2a

In Figure 2a, the bold shading of P1 is used to depict that P1 is currently tagged ACCESSED. That is, the collection of knowledge elements that are in P1, but not in P2 or P3, are VISIBLE. In Figure 2b, P3 is tagged ACCESSED, that is, the knowledge elements in P3 and in the surrounding packet P1 are VISIBLE. The knowledge elements in P2 and P4 are not eligible for use by the KBS in the access condition shown.

For illustration, consider P1 containing some general knowledge stating that institutions are localized in buildings. In the access condition shown in Figure 2a only such general information about institutions is VISIBLE. Now think of P3 containing knowledge about restaurants. When P3 is tagged ACCESSED (as in Figure 2b) knowledge about restaurants in P3 is VISIBLE as well as more general knowledge about institutions in the surrounding packet P1.

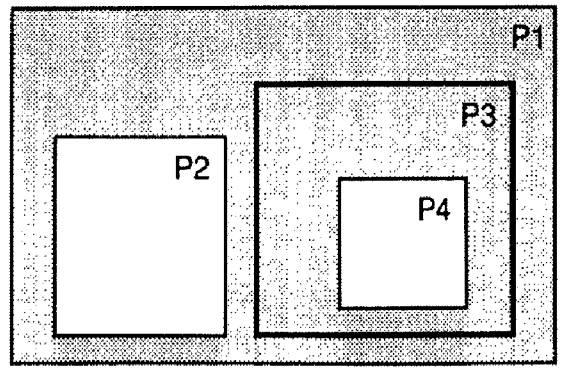

Figure 2b

The principle of single access seems restrictive but it is only posited here to keep things straight for the basic model. The meaning of "multi-access" needs separate discussion which is postponed to the end of this paper. Multiple access would allow reasoning with competitive knowledge at one time which would make it possible to obtain (or detect!) contradictory statements among VISIBLE knowledge.

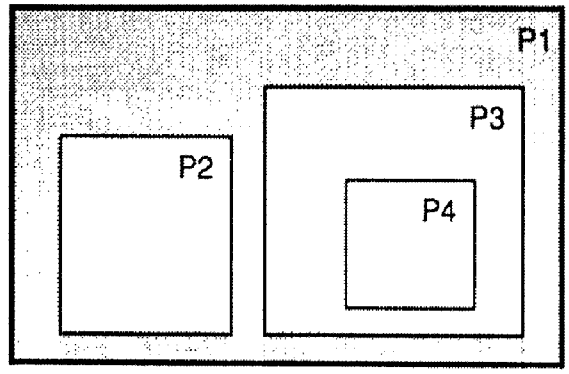

Figure 3a

In Figure 3a, P1 is tagged ACCESSED and thus P2, P3, and P4 are REACHABLE. In Figure 3b, P3 is tagged ACCESSED, hence only P4 is REACHABLE while P2 is not (and neither it is VISIBLE). If P2, for example, contains knowledge about museums, P3 about restaurants, and P4 about Japanese restaurants, then all this information is REACHABLE in the access condition shown in Figure 3a. In Figure 3b, restaurant information (P3) is tagged ACCESSED, hence only information about Japanese restaurants is REACHABLE. The principle of reachability of knowledge attributes a special role to knowledge elements packed within the packet tagged ACCESSED which is reflected in a principle describing a dynamic access condition (structure-dependent access) given further below.

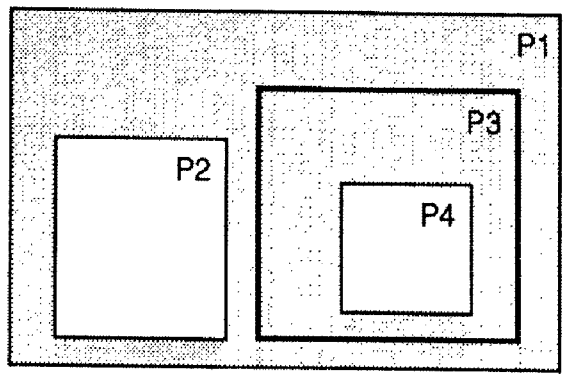

Figure 3b

Overall, static access conditions are characterized in these three principles by splitting the notion of accessed knowledge into VISIBLE and REACHABLE knowledge.

4.3 Dynamic Access Conditions

Dynamic access conditions concern the way the movement of the ACCESSED tag in the structured KB is controlled by some strategy. There may be global control having to do with domain-independent problem-solving strategies, agenda setting, goal selection, etc. These may bring about a KBS's ultimate decisions to act (which may include the need to use certain knowledge from the KB), whereas a KB, by itself, can only

support different potential ways of acting (possibly competitive ones). However, the way a KB is structured and what sorts of knowledge elements are packed together does embody some sort of local control in that it constrains the succession in which access to knowledge is attempted.

In general, the empirical observations called upon here say much about human cognitive functioning in terms of how one is able to act on the basis of an enormous repertoire of knowledge elements without getting confused. The findings suggest that a critical feature of human intelligence lies in a dynamic partitioning of the total knowledge into "visible" and "invisible" parts such that the visible part is normally small enough to be computationally tractable. To prepare the ground for exploiting this feature in a KBS, an attempt is made here to capture some observations on principles of dynamic access conditions.

Principle of structure-dependent access. When dealing with a task situation on the basis of knowledge currently VISIBLE turns out unsuccessful, the ACCESSED tag is moved to *one* of those knowledge packets REACHABLE next.

Principle of keyword-dependent access. A means to tag a packet of domain-specific knowledge ACCESSED is the finding of certain concept words (or combinations of concept words) directly associated with knowledge elements in this packet. We refer to such words as *keywords*.

Principle of persistence of access conditions. Upon completion of a goal, the current ACCESSED tag persists as a start-off condition for the partitioning state of the knowledge base when the next goal is issued.

The principle of structure-dependent access reflects the empirical observation that "zooming in" or "focusing" within a packet of domain-specific knowledge is given preference over "wandering." It suggests that, given that a certain domain was found relevant, more specific knowledge (if available) in that domain is accessed next. Thereby, while the VISIBLE part of a KB gets larger, the REACHABLE part gets progressively smaller (see Figures 3a,b). For motivation, imagine that a query about some specialty in a restaurant is issued. If no suitable information is found in the restaurant packet, it seems reasonable to look into the Japanese-restaurant packet. It may come to the point that the resources of a packet are exhausted with no knowledge found applicable but competitive knowledge still available. The question what is to happen then is left open for treatment in a particular implementation. If desired, some form of backtracking would allow a KB to be traversed totally. In this case, the advantage of a well-structured KB over an unstructured one might lie in a higher chance to find the "right" piece of knowledge soon.

The principle of keyword-dependent access suggests the inclusion of vocabulary terms (concept signifiers) among knowledge elements associated with those concepts within a packet. This reflects the empirical observation that rapid access to certain knowledge is often triggered by wordings or technical terms in the context or in a question or a problem statement conveyed. For example, when in a piece of touring guide text the words "museum" and "ceramics collection" are mentioned, one can be pretty definite in many cases that it is about objects of interest (and not about Japanese restaurants). It might even be the case that this interaction, or association, between term words and knowledge elements constitutes a primary means of access that may also be involved in choice making among competitive knowledge packets.

Again, the principle of persistence of access conditions is suggested by empirical observations. As further task-descriptive data (or queries, or sentences in a discourse) are issued to an individual, the recent history (previous discourse) may influence the way these data are processed, at least when there is no break in coherency. This feature reflects what one might call a "mind set."

5 Discussion

The proposal presented in this paper characterizes domain-specific knowledge as an organized repertoire of knowledge elements. The notion of structure is based on cognitive principles and is independent of particular representation languages. Static and dynamic access conditions give rise to a finite set of knowledge base partitions each of which can be considered as an autonomous knowledge base which characterizes a certain segment of the modeled domain. While the completeness and consistency of a so-structured knowledge base cannot be guaranteed in general, the notion of local consistency restricts consistency proofs to the domain of VISIBLE knowledge and might prove tractable and sufficient most of the time. Since knowledge pertaining to different (sub)domains is kept separate by packing, global inconsistencies seem less likely. Even when inconsistencies exist they might not come to bear. However, a mechanism should be provided for which gives notice of the discovery of inconsistencies at run time so that corrective action can be taken.

The principles are not claimed to be comprehensive. There might be additional ones yet to be formulated. In turn, only some of the principles might be realized in an actual implementation, depending on the particular problem domain or purpose of a KBS. It should be understood that the principles are intended for improving the manageability of *large* knowledge bases containing *diverse* knowledge. A conclusion near at hand is that knowledge packets might be encapsulated as modules from the aspect of a "responsibility assignment" (Parnas 1972).

One of the experiences in software engineering is that in order to obtain a transparent module structure, the use conditions among modules should be hierarchic to the greatest extent possible (Nagl 1990). If a knowledge base is to be conceived in a hierarchical fashion then the question arises under which aspect a hierarchical structure should be established. The DOKS principles presented in the fourth section take *specificity* as an aspect of hierarchic subordination. A lot of the background knowledge in a knowledge base is domain-specific, and it is often possible to distinguish different degrees of specificity. The more specific a piece of knowledge is, the more restricted is its use with respect to the number of situations it can come to bear on. That is, more general knowledge should be exported more "generously" than more specific knowledge, and the more general knowledge should not import very specific knowledge which could be of very local impact and even be unqualified to be applied. On the other hand, very specific knowledge should import a lot of general knowledge that covers a broad range of situations including the ones that the very specific knowledge is good for.

While the point of view in the preceding sections was directed toward increasingly specific knowledge, one can easily imagine larger knowledge bases to be built by integrating existing knowledge bases from different origin. Each knowledge base can be regarded as a large knowledge packet that may have an internal knowledge packet structure. Where different knowledge sources were used to construct self-contained knowledge bases, these could be viewed as separate packets which are made competitive knowledge packets and by this become integrated in a larger knowledge packet structure. In this way, ever larger bodies of diverse background knowledge can be assembled to cover growing domains of discourse. It has been the subject of debate whether domain-specific inference rules should be part of the LILOG inference engine or should be kept in the domain knowledge base. It should be one aspect of further research to clarify which kind of rules should be globally available (i.e., part of the inference engine) and which rules should be kept separate in knowledge packets as repositories for local reasoning.

Several issues have been excluded in this paper. The technical questions and implications of an *overlap of knowledge packets* have been discussed extensively in (Wachsmuth 1989b). More research is needed to see when this feature should actually be used in structuring a knowledge base. While knowledge packet structures as presented here have a tree-structured access graph (concerning visibility and reachability of knowledge packets), general knowledge packet structures may have access graphs that allow competitive knowledge packets to become VISIBLE. There are two main issues that need to be considered: the sharing of partitions of a knowledge base in certain access states, and the requirements arising from local consistency. If the VISIBLE knowledge of competitive knowledge packets needs to be included for handling specific situations, it may be motivated to subordinate a new and more specific knowledge packet coined for this situation to these competitive packets. Then at this point the consistency of the comparatively large partition of the whole knowledge base that can become eligible for inferences needs to be scrutinized for consistency.

The notion of *multi-access* mentioned in discussing the principle of single-access has to be further researched. It might be motivated to make competitive knowledge packets simultaneously available for inferencing when their knowledge needs to be drawn on in one task. However, simply unifying the corresponding partitions of the knowledge base would conflict with the conditions of local consistency. Rather, it seems sensible to accumulate the results of inferences drawn on different KB partitions in a KB workspace and record their origin (i.e., the focus under which they were derived). Then the conclusions generated from different foci can be used by the system while further inferences that draw on background knowledge still submit to the principle of single access. Early experimental implementations have taken first steps to make use of these ideas (Wachsmuth 1985, Gust 1986) but they are not fully worked out yet. Apparently, reason maintenance would have to be related to the different partitions that are used at run time.

While the principles characterizing knowledge organization and static access conditions are addressed to the idea of modular knowledge bases in general, particularly interesting ideas for dynamic access control arise for natural language systems. With respect to task-oriented dialogue systems, Grosz and Sidner (1986) have formalized similar ideas to those presented above by means of a "focus space stack." They define a focus space to capture "an abstraction of the participants' focus of attention as their discourse unfolds" by including information about objects, properties, relations, and intentions that are most salient. The focus space of the task under consideration is pushed onto a stack and remains there until the task is completed. That is, it is VISIBLE regardless of whether or not other focus spaces of subtasks are considered at intermediate inference steps.

Furthermore, components of linguistic analysis and a focus space stack could take mutual advantage of each other. The focus space of the discourse segment under consideration may constrain the search for possible referents of definite noun phrases and pronouns and help to resolve ellipsis (cf. Scha, Bruce, Polanyi 1987). In the interpretation of an incoming definite noun phrase (or pronoun or ellipsis) possible referents are first searched in the most recent focus space before going back to less recent ones. In turn, information about the attentional state of the unfolding discourse can be gained from linguistic structure (e.g., functional words and word order). The principles of keyword-dependent access and of the persistence of access conditions are particularly keyed to the use of knowledge-based systems in natural language interaction. Further insight into selective knowledge use might be gained by more subtle analysis of the general interaction of linguistic structure and knowledge access. It would be an interesting research question to analyse the interaction of linguistic structure and domain-focusing and exploit it for elaborating dynamic access control in modular knowledge bases.

References

Dahlgren, K. (1985): The cognitive structure of social categories. Cognitive Science 9(3), 379-398

Feigenbaum, E.A. (1977): The art of artificial intelligence: themes and case studies of knowledge engineering. Proc. 5th Int. Joint Conf. on Artificial Intelligence (IJCAI-77), Cambridge, MA

Fodor, J.A. (1983): The Modularity of Mind. MIT Press, Cambridge, MA

Gängler, B., Wachsmuth, I. (1991): Antwortgenerierung, flexible Wortwahl und elaborative Inferenzen - ein Regelinventar für LEU/2. In: Klose, G., Lang, E., Pirlein, T. (eds.): Die Ontologie und Axiomatik der Wissensbasis von LEU/2. IWBS-Report 171, IBM Deutschland, Stuttgart/ Heidelberg

Garfield, J.L. (ed.) (1987): Modularity in Knowledge Representation and Natural Language Understanding. MIT Press, Cambridge, MA

Grosz, B., Sidner, C. (1986): Attention, intentions and the structure of discourse. Computational Linguistics 12, 175-204

Gust, H. (1986): Strukturiertes Wissen als Grundlage für Sprachverstehensprozesse. LDV-Forum 4(2), 9-14

Hendrix, G.G. (1979): Encoding knowledge in partitioned networks. In: Findler, N.V. (ed.): Associative Networks - Representation and Use of Knowledge by Computers. Academic Press, New York, pp. 51-92

Kowalski, R. (1975): A proof procedure using connection graphs. Journal of the ACM 22(4), 572-595

Lenat, D.B., Feigenbaum, E.A. (1987): On the thresholds of knowledge. Proc. 10th Int. Joint Conf. on Artificial Intelligence (IJCAI-87), Milano, pp. 1173-1182

Levesque, H.J., Brachman, R.J. (1986): Knowledge level interfaces to information systems. In: Brodie, M.L., Mylopoulus, J. (eds.): On Knowledge Base Management Systems. Springer-Verlag, New York, Berlin, Heidelberg, pp. 13-54

Lifschitz, V. (1989): Benchmark problems for formal nonmonotonic reasoning, version 2.00. In: Reinfrank, M., deKleer, J., Ginsberg, M.L., Sandevall, E. (eds.): Non-Monotonic Reasoning. Proc. 2nd Int. Workshop, Springer-Verlag, Berlin, Heidelberg, New York, pp. 202-219

Minsky, M. (1981): A framework for representing knowledge. In: Haugeland, J. (ed.): Mind Design, MIT Press, Cambridge, MA, pp. 95-128

Nagl, M. (1990): Softwaretechnik: Methodisches Programmieren im Großen. Springer-Verlag, Berlin, Heidelberg, New York

Newell, A., Simon, H.A. (1972): Human Problem Solving. Prentice Hall, Englewood Cliffs, NJ

Newell, A. (1982): The knowledge level. Artificial Intelligence 18(1), 1-20

Parnas, D.L. (1972): On the criteria to be used in decomposing systems into modules. Communications of the ACM 15(12), 1053-1058

Pletat, U. (1991): Modularizing Knowledge in L_{LILOG}. IWBS-Report 173, IBM Deutschland, Stuttgart/ Heidelberg

Post, T.R., Behr, M.J., Lesh, R., Wachsmuth, I. (1985): Selected results from the rational number project. Proc. 9th Int. Conf. for the Psychology of Mathematics Education, Utrecht, pp. 342-351

Prerau, D.S., Gunderson, A.S., Reinke, R.E., Adler, M.R. (1990): Maintainability techniques in developing large expert systems. IEEE Expert June 1990, 71-79

Rosch, E. (1978): Principles of categorization. In: Rosch, E., Lloyd, B.B. (eds.): Cognition and Categorization. Erlbaum, Hillsdale, NJ

Scha, R.J.H., Bruce, W.C., Polanyi, C. (1987): Discourse understanding. In: Shapiro, S.C., Eckroth, D., Vallasi, G.A. (eds.): Encyclopedia of Artificial Intelligence, Vol. 1. John Wiley and Sons, New York, pp. 233-245

Wachsmuth, I. (1985): LAKOS - Ein Modell der Wissensrepräsentation zur Erklärung kognitiven Verhaltens. In: Mandl, H., Fischer, P.M. (eds.): Lernen im Dialog mit dem Computer. Urban and Schwarzenberg, München

Wachsmuth, I. (1987): On Structuring Domain-Specific Knowledge. LILOG-Report 12, IBM Deutschland, Stuttgart

Wachsmuth, I. (1989a): Modularisierung wissensbasierter Systeme - Rahmenentwurf für ein Forschungsprogramm im Bereich Wissensbasierte Systeme/Künstliche Intelligenz. University of Bielefeld (MOSYS-Report 1), Bielefeld

Wachsmuth, I. (1989b): Zur intelligenten Organisation von Wissensbeständen in künstlichen Systemen. IWBS-Report 91, IBM Deutschland, Stuttgart/Heidelberg

Wachsmuth, I., Meyer-Fujara, J. (1990): Addressing the retrieval problem in large knowledge bases. Proc. 3rd Conf. Computational Intelligence (CI-90), Milano (Summary). Full paper: University of Bielefeld (MOSYS-Report 3), Bielefeld

Wos, L., Overbeek, R., Lusk, E., Boyle, J. (1984): Automated Reasoning - Introduction and Applications. Prentice Hall, Englewood Cliffs, NJ