Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Document
D-92-08

DFKI Workshop on Taxonomic Reasoning

Saarbrücken, February 26, 1992

Proceedings

Jochen Heinsohn, Bernhard Hollunder (Eds.)

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern und Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Daimler Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Krupp-Atlas, Mannesmann-Kienzle, Philips, Sema Group Systems, Siemens and Siemens-Nixdorf. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- ☐ Intelligent Engineering Systems
- ☐ Intelligent User Interfaces
- ☐ Intelligent Communication Networks
- ☐ Intelligent Cooperative Systems.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Prof. Dr. Gerhard Barth
Director

DFKI Workshop on Taxonomic Reasoning Proceedings

Jochen Heinsohn, Bernhard Hollunder (Eds.)

DFKI-D-92-08

DFKI Workshop
on
Taxonomic Reasoning

DFKI, Saarbrücken
February 26, 1992

Organization: Jochen Heinsohn
Bernhard Hollunder

Contents

Preface

The internal DFKI workshop on "Taxonomic Reasoning" was held in Saarbrücken on February 26, 1992. More than 30 participants (most of them are members of DFKI) attended the workshop and nine talks were given.

With respect to the specific intent of each presentation—language extension, application, or experience report—the workshop was separated into four sections.

- First, WIP and WINO reported from their activities regarding Terminological Logics (TL). The first talk gave a general introduction to TL, the second an overview of the research activities in WINO, and the following two described language extensions and specific applications currently investigated in WIP.
- The second section contained two presentations of the DISCO and ASL projects. Both talks concentrated on specific problems in natural language processing which may possibly be solved or at least better be handled in the framework of TL.
- The third section started with an overview of the role taxonomic reasoning plays in the HYDRA project. ALV and ARC-TEC reported from their ideas and experience, respectively, with employing TL. It was argued that at the moment TL are not the appropriate formalism for profitably being employed in the project ALV in the framework of document analysis. On the other hand, TL already play a major role in ARC-TEC, i.e., the knowledge representation component is realized by the terminological system TAXON.
- The last section consisted of the final discussion where every DFKI project group was represented by at least one member.

In summarizing the results of the workshop we restrict ourselves to the most important aspects that will influence the future work.

It turned out that the project groups ARC-TEC, DISCO, WINO, and WIP already employ or intend to use TL in applications. For most applications, however, it seems that classical TL are too weak, i.e, language extensions are required and developed to solve specific problems in real world modeling (e.g., incremental consistency check, belief revision, plans, and uncertainty).

A main topic of the discussion was related to the development of terminological systems. At DFKI two such systems are under implementation. The TAXON system, developed by ARC-TEC, serves as the knowledge representation component in a mechanical engineering application. TAXON's inference mechanism basically consists of sound and complete algorithms which mostly have been developed jointly by ARC-TEC and WINO. The system is well-tailored to engineering applications. Consequently, TAXON provides facilities which are not included in classical terminological systems (e.g., the integration of concrete domains, an extended rule calculus), but does not include some standard language constructs (e.g., number restrictions).

The implementation of the other system, called KRIS, was mainly motivated by theoretical results on sound and complete inference algorithms for TL. Although KRIS currently has certain deficiencies (e.g., an inefficient implementation), the system is already employed as the knowledge representation tool in the project WIP and has been extended with respect to the application.

It is planned to further optimize and improve KRIS and TAXON jointly in the projects WINO, WIP, and ARC-TEC. One final goal is to distribute KRIS as a kind of "public domain system" (at least within DFKI) such that every project group has an easy access to KRIS. The future integration of TL into other systems should be facilitated with the help of precisely defined and described interfaces.

The maintenance of the software (the KRIS system in particular) developed at DFKI, has been discussed. There was an agreement between all participants that it seems to be useful to establish positions for software maintenance at DFKI.

Another result of the discussions was that, for the ALV project, TL in general (and the KRIS system in particular) are too complex and not suitable for being fruitfully employed in the domain of document analysis. During the discussions several case studies and possibilities were evaluated.

Another topic was related to the coordination of future research with respect to taxonomic reasoning—not necessarily restricted to the DFKI activities. In order to have more information about the activities regarding TL at DFKI, the Universities of Saarbrücken and Kaiserslautern, and the Max-Planck Institute for Computer Science in Saarbrücken, it is intended to generate a list containing the currently investigated problems and topics (including diploma or doctoral theses). In addition, a newsgroup for TL is going to be established in order to have a medium for fast information exchange and retrieval as well as a problem corner.

It is planned to organize a further DFKI workshop on taxonomic reasoning in about one year. Meanwhile, it is intended to have informal meetings—as already existing between WINO and WIP (e.g., regarding improvements of KRIS)—in order to discuss related problems and trigger the exchange of know-how concerning TL. Promising collaborations have been triggered during the workshop. ARC-TEC and WIP will cooperate concerning the representation of actions in TL. An extension of KRIS may be used as the knowledge representation kernel for DISCO.

JH, BH

List of Participants

Jürgen Allgayer (SFB)
Franz Baader
Susanne Biundo
Reiner Bleisinger
Harold Boley
Stephan Busemann
Hans-Jürgen Bürckert
Enrico Franconi (IRST)
Philipp Hanschke
Matthias Hecking
Jochen Heinsohn
Bernhard Hollunder
Hans-Ulrich Krieger
Daniel Kudenko
Armin Laux
Wolfgang Maaß
Michael Malburg
Bernhard Nebel
John Nerbonne
Joachim Niehren
Werner Nutt
Hans-Jürgen Profitlich
Carola Reddig (SFB)
Thomas Reinartz
Frank Schneiderlöchner
Jörg Siekmann
Renate Schmidt (MPI)
Gert Smolka
Ralf Treinen
Wolfgang Wahlster
Jörg Würtz
Detlef Zimmermann

Workshop Program

DFKI-Workshop über taxonomisches Schließen
— Entwicklungen, Anwendungen, Erfahrungen —
Mittwoch, 26. Februar 1992
Saarbrücken, Bananensaal des DFKI

09:30 Jochen Heinsohn, Bernhard Hollunder (WIP-WR, WINO) *Eröffnung*

09:35 Bernhard Nebel (WIP-WR, 15min)
Terminologische Logiken - Eine Übersicht
Hans-Jürgen Bürckert (WINO, 25min)
Aktuelle und geplante Forschung im Bereich terminologischer Logiken im Projekt WINO
Hans-Jürgen Profitlich (WIP-WR, 15min)
Repräsentation von Aktionen in terminologischen Logiken
Jochen Heinsohn (WIP-WR, 15min)
Repräsentation von Unsicherheit in terminologischen Logiken

11:15 **Kaffeepause**

11:45 Stephan Busemann (DISCO, 30min)
Über Anforderung an Wissensrepräsentation und -modellierung bei der Wortwahl
John Nerbonne (DISCO, ASL, 30min)
Über sortenbasierte Desambiguierung und über die Verwendung von Inferenzen in KL-ONE für die semantische Verarbeitung

13:00 **Mittagspause**

14:00 Gert Smolka (HYDRA, 25min)
Logikprogrammierung und terminologische Inferenz
Michael Malburg, Reiner Bleisinger (ALV, 30min)
Dokumentanalyse:
Möglichkeiten zum Einsatz von taxonomischem Schließen?
Philipp Hanschke (ARC-TEC, 30min)
Anforderungen intelligenter Ingenieursysteme an eine TL-Wissensrepräsentationskomponente

Franz Schmalhofer, Thomas Reinartz (ARC-TEC, 10min) (canceled)
Definition von Problemklassen einer komplexen Anwendungsdomäne in einer terminologischen Sprache

15:30 **Kaffeepause**

16:00 **Abschlußdiskussion**
Ziel dieser Diskussion soll es sein, Gemeinsamkeiten und Unterschiede bei der Anwendung von terminologischen Logiken in verschiedenen DFKI-Projekten herauszuarbeiten und zukünftige Arbeiten auf diesem Gebiet abzustimmen.

Terminological Logics & Representation Systems

Bernhard Nebel
Project WIP

1 Introduction

Terminological logics form a family of representation formalisms that support the representation of and reasoning with domain-specific *terminologies.* The roots of this family of representation formalisms are Brachmann's [4] *structured inheritance networks.* Since then, a large number of different representation systems based on this representation paradigm have been built and the semantic and computational foundations of this kind of representation formalisms have been investigated thoroughly.

In particular the investigation of the theoretical foundations (for a survey of recent results see [12]) of terminological logics has been very actively perused in recent years. Meanwhile, inference algorithms for almost all interesting terminological logics are known, where most of them are based on a technique developed by Schmidt-Schauß and Smolka [16] and refined by Hollunder, Nutt, and Schmidt-Schauß [11]. These algorithms are optimal w.r.t. the problem complexity of the respective inference problems.

After having "straightened out" the theoretical area, it now seems to be necessary to focus more on practical aspects such as the design of efficient and useful terminological representation systems and the extension and terminological logics by other forms of representation formalisms and reasoning.

2 Integration and Extension

The basic formalism of a terminological logic alone (also often called "TBox") is not sufficient to support applications effectively. For this reason, usually a "world description" or "ABox" is added to the basic formalism. Such a combination of a TBox and ABox is very similar to an object-based database system, and, as a matter of fact, such TBox/ABox systems can be used as intelligent information retrieval systems [14, 3, 5, 6].

Besides using terminological representation systems as intelligent information retrieval systems, these representation systems are often employed as subsystems in AI systems, such as design systems [19], configuration systems [13], or natural language systems [18, 2]. However, in this context terminological reasoning is usually intertwinned with other forms of representation and reasoning, which is not supported by terminological representation systems. The common solution in this situation is to integrate the different forms of representation and reasoning by "programming".

For often occurring combinations, it is, of course, preferable to provide generic solutions. This reduces the work to build an AI system and the risk of providing unsound solutions. So, for instance, often defeasible reasoning or probabilistic reasoning [8] shall be integrated with terminological reasoning. While straight-forward solutions can be easily provided, most of them turn out to be problematical leading to incorrect or unintuitive behavior. Similarly, the integration of time [17, 15] and action [7, 9] into terminological logics are often mentioned topics of interest where straight-forward solutions can be easily specified. A tight integration of both forms of reasoning is usually not easy, however, neither from a theoretical nor implementational point of view. Most probably, no general-purpose solutions will be available for these problems if we restrict ourselves to practically feasible methods, i.e., tractable or almost tractable algorithms.

Summarizing, the area of integrating terminological logics with other forms of representation and reasoning seems to be interesting and challenging from a practical and theoretical point of view.

3 Efficiency of Terminological Representation Systems

While the theoretical research in terminological logics has led to a state where we know a lot about the structure of the computational processes underlying terminological reasoning, the art of building fast and usable systems has largely remained a matter of personal knowledge.

A systematic comparison of six different terminological representation systems [10] revealed that not only the size but also the structure of real knowledge bases is responsible for the runtime. Further, the systems show a wide variation in runtime performance on real as well as on constructed examples (see Figure 1).

One of the possible conclusions one could draw from this experiment is that systems that support complete inferences for powerful languages (KRIS is such a system) are necessarily considerably slower than systems that are incomplete (such as LOOM) or systems that support a much less expressive language (such as CLASSIC).

Before drawing such a conclusion, one should consider the motivations for building the systems, however. While CLASSIC and LOOM were built to be used in applications, KRIS was intended to be an experimental testbed for different inference algorithm with no emphasis on efficiency. This does not mean that KRIS *must* be slower. However, it may be an indication that it is possible to speed up the system.

In fact, first experiences with possible optimizations and the empirical analysis of different *classification* algorithms indicate that the efficiency of a system is not necessarily dominated by the computational costs of the subsumption algorithm. Further, there do not seem to be principal reasons that prohibit a speedup of KRIS for knowledge bases on which CLASSIC and LOOM are fast. It is an open problem, however, whether conventional

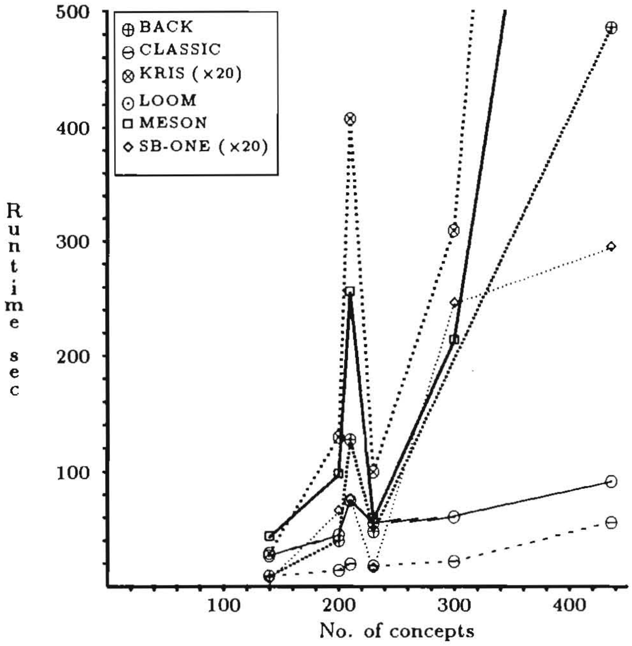


Figure 1: Runtime performance on real KBs

optimization techniques as used in CLASSIC and LOOM do also apply to more powerful languages, or whether new techniques have to be developed.

Such considerations of the "practical" efficiency are, of course, necessary when a system is intended to be used in applications. In fact, the implementation of the RAT system on top of KRIS forced us to look into these efficiency issues. Additionally, even from a scientific point of view it is an interesting topic since almost nothing is published about efficient implementation techniques and algorithms that are fast on typical knowledge bases. First experiments, for instance, indicate that advanced algorithms that are fast on randomly generated partial orders tend to have problems with knowledge bases occurring in practice. Further, our experiments indicated that advanced optimization techniques only have effects on *some* real knowledge bases, namely, knowledge bases that were not built by KR researchers but by linguists.

4 Conclusion

In the last few years, a solid theoretical foundation for terminological logics has been provided. Now, it seems to be the time to enhance the utility of terminological representation systems. In this paper, two important topics were sketched, namely, extension and integration of other forms of reasoning and the systematic analysis of existing systems and the development of efficient algorithms and implementation techniques. There are, of course, more such topics, such as analyzing and classifying application requirements for terminological representation systems and empirical and analytical investigations of knowledge bases that are used in applications, just to name some. In general, the analysis of *applications* will probably become more important then the analysis of *formalisms* in the area of terminological logics.

References

[1] *Proceedings of the 8th National Conference of the American Association for Artificial Intelligence*, Boston, MA, Aug. 1990. MIT Press.

[2] J. Allgayer and C. Reddig. Processing descriptions containing words and gestures—a system architecture. In C.-R. Rollinger and W. Horn, editors, *GWAI-86 und 2. Österreichische Artificial-Intelligence-Tagung*, pages 119–130, Ottenstein, Austria, Sept. 1986. Springer-Verlag.

[3] H. W. Beck, S. K. Gala, and S. B. Navathe. Classification as a query processing technique in the CANDIDE semantic data model. In *Proceedings of the International Data Engineering Conference, IEEE*, pages 572–581, Los Angeles, CA, Feb. 1989.

[4] R. J. Brachman. *A Structural Paradigm for Representing Knowledge.* PhD thesis, Havard University, 1977.

[5] M. Damiani, S. Bottarelli, M. Migliorati, and C. Peltason. Terminological Information Management in ADKMS. In *ESPRIT '90 Conference Proceedings*, Dordrecht, Holland, 1990. Kluwer.

[6] P. T. Devanbu, R. J. Brachman, P. G. Selfridge, and B. W. Ballard. LaSSIE: a knowledge-based software information system. *Communications of the ACM*, 34(5):35–49, May 1991.

[7] P. T. Devanbu and D. J. Litman. Plan-based terminological reasoning. In J. A. Allen, R. Fikes, and E. Sandewall, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 2nd International Conference*, pages 128–138, Cambridge, MA, Apr. 1991. Morgan Kaufmann.

[8] J. Heinsohn. A hybrid approach for modeling uncertainty in terminological logics. In R. Kruse and P. Siegel, editors, *Symbolic and Quantitative Approaches to Uncertainty, Proceedings of the European Conference ECSQAU*, Lecture Notes in Computer Science 548, pages 198–205. Springer-Verlag, Berlin, Heidelberg, New York, 1991.

[9] J. Heinsohn, D. Kudenko, H.-J. Profitlich, and B. Nebel. Integration of action representation in terminological logics. In C. Peltason, K. von Luck, and C. Kindermann, editors, *Terminological Logic Users Workshop - Proceedings*, page 117, Berlin, Germany, Dec. 1991. Published as KIT Report 95.

[10] J. Heinsohn, D. Kudenko, H.-J. Profitlich, and B. Nebel. An empirical analysis of terminological representation systems. In *Proceedings of the 10th National Conference of the American Association for Artificial Intelligence*, San Jose, CA, July 1992. MIT Press. To appear.

[11] B. Hollunder, W. Nutt, and M. Schmidt-Schauß. Subsumption algorithms for concept description languages. In B. Neumann, editor, *Proceedings of the 10th European Conference on Artificial Intelligence*, pages 348–353, Vienna, Austria, Aug. 1990. Pitman.

[12] B. Nebel and G. Smolka. Attributive description formalisms ... and the rest of the world. In O. Herzog and C.-R. Rollinger, editors, *Text Understanding in LILOG*, volume 546 of *Lecture Notes in Artificial Intelligence*, pages 439–452. Springer-Verlag, Berlin, Heidelberg, New York, 1991.

[13] B. Owsnicki-Klewe. Configuration as a consistency maintenance task. In W. Hoeppner, editor, *Künstliche Intelligenz. GWAI-88, 12. Jahrestagung*, pages 77–87, Eringerfeld, Germany, Sept. 1988. Springer-Verlag.

[14] P. F. Patel-Schneider, R. J. Brachman, and H. J. Levesque. ARGON: Knowledge representation meets information retrieval. In *Proceedings of the 1st Conference on Artificial Intelligence Applications*, pages 280–286, Denver, Col., 1984.

[15] K. Schild. A tense-logical extension of terminological logics. KIT Report 92, Department of Computer Science, Technische Universität Berlin, Berlin, Germany, Sept. 1991.

[16] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.

[17] A. Schmiedel. A temporal terminological logic. In AAAI-90 [1], pages 640–645.

[18] B. L. Webber and R. J. Bobrow. Knowledge representation for syntactic/semantic processing. In *Proceedings of the 1st National Conference of the American Association for Artificial Intelligence*, pages 316–323, Stanford, CA, Aug. 1980.

[19] B. C. Williams. Interaction-based invention: Designing novel devices from first principles. In AAAI-90 [1], pages 349–356.

WINO—Logical Foundations of (Terminological) Knowledge Representation and Processing: Current and Future Research

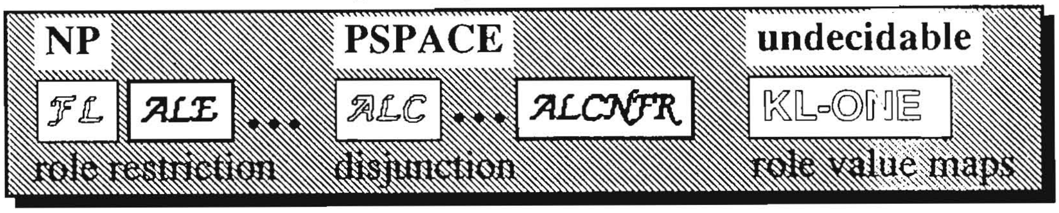
Hans-Jürgen Bürckert, Franz Baader,
Bernhard Hollunder, Armin Laux, Werner Nutt

The research aim of the WINO project was the investigation of logic-based knowledge representation (KR) formalisms, and of suitable inferences for these formalisms. Until now, we have focused on terminological knowledge representation formalisms, so-called terminological logics, which can be seen as descendants of the original system KL-ONE. When WINO started in May 1989, there already existed various implementations of terminological KR systems, but their inference components were mainly based on incomplete algorithms, i.e., on algorithms which could not deduce all the facts implied by the knowledge base. One reason for the use of such algorithms was that sound and complete algorithms were only known for rather trivial languages. Another was that first results on the worst case complexity of typical inference problems for terminological systems indicated that one could not expect to have polynomial algorithms for larger languages.

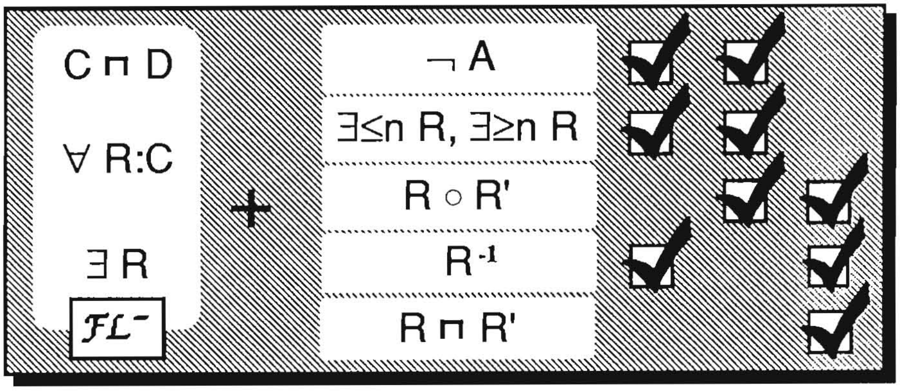
During the last three years of research we have investigated these typical inference problems for terminological logics more closely. The languages we have considered comprise concept forming operators such as conjunction, disjunction, negation, value/existential restrictions, number restrictions, agreement and disagreement of functional roles as well as certain role forming operators such as conjunction of roles or transitive closure of roles. As a main result of our investigations we have obtained sound and complete algorithms for inferences such as subsumption and consistency checking for a great variety of different languages (see e.g., [6, 7, 8, 10, 11, 1, 2, 9]). These algorithms are based on a tableaux-like calculus that yields a rather general and intuitive method for the integration of other constructs (e.g. qualifying number restriction, role chains, inverse roles). This calculus, which is a generalization of an approach of Manfred Schmidt-Schauß and Gert Smolka [12], not only yields inference algorithms. It has also enabled us to classify various terminological logics w.r.t. the complexity of typical inference problems, and it helps to revealing the different sources of complexity.

We could show that most terminological logics are intractable, i.e., subsumption of concepts is usually an NP-hard problem. Only a few and rather small sublanguages have polynomial complexity (see figure below).

Most concept languages are intractable



Only a few languages are polynomial



Based on our theoretical work on inference algorithms, we have specified and implemented the demonstrator system KRIS (Knowledge Representation and Inference System) [5], which is a terminological KR system in the tradition of KL-ONE. This system provides the user algorithms for terminological reasoning (e.g. satisfiability and subsumption of concepts) and assertional reasoning (e.g. consistency checking, instantiation) for several very expressive concept languages. KRIS differs from other terminological systems in that all its reasoning facilities are realized by complete algorithms. Together with the DFKI project WIP, KRIS has been evaluated with respect to its runtime performance on some large and realistic knowledge bases. It turned out that the first, unoptimized version of KRIS was orders of magnitude slower than some other terminological systems tested by WIP. However, this is not necessarily a consequence of the use of complete algorithms. First optimizations of the system are very promising. They seem to indicate that an optimized version of KRIS can have a runtime behaviour which is similar to that of the other systems, at least on the test data used by WIP.

In collaboration with other DFKI project groups (WIP and ARC-TEC) several application driven extensions of terminological languages have been proposed and investigated during the WINO project. For example, we have considered cyclic concept definitions [1], role forming constructs for union, composition, and transitive closure of roles [2], and functional roles mapping into concrete domains, thus allowing for the integration of e.g. real numbers [4]. For these language extensions inference algorithms have been developed, which are partially going to be realized in systems implemented by other groups.

Another important extension of the purely terminological formalism was the constrained based integration of terminological logics into full predicate logics (which yields the so-called concept logics) [3].

For the future—in our new project TACOS (for TAxonomic and COmmon Sense reasoning)—we intend to investigate non-classical extensions of terminological logics in order to incorporate

- temporal and spatial information (cf. concepts like perishable goods),
- belief dependent information (e.g. presumably duty-free goods),
- prototypical or default reasoning (e.g. furniture trucks usually transport furniture, but may be used also otherwise),
- uncertain and vague knowledge (cf. concepts like small forwarding agencies).

We hope that we can provide not only a logical foundation of such extensions, but can also adapt our inference procedures in order to deal with these kinds of commonsense knowledge in an efficient and logically complete manner. The results should be realized in a demonstator system (KRISTOS—Knowledge Representation and Inference System for Taxonomical and cOmmonSense reasoning) based on the optimized version of KRIS.

References

[1] F. Baader. Terminological cycles in KL-ONE-based knowledge representation languages. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 621–626, Boston, Ma., 1990.

[2] F. Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, 1991.

[3] F. Baader, H.-J. Bürckert, B. Hollunder, W. Nutt, and J. H. Siekmann. Concept logics. In *Proceedings of the Symposium on Computational Logics*, Brüssel, Belgium, 1990.

[4] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, 1991.

[5] F. Baader and B. Hollunder. $\mathcal{KRIS}$: $\mathcal{K}$nowledge $\mathcal{R}$epresentation and $\mathcal{I}$nference $\mathcal{S}$ystem. *SIGART Bulletin*, 2(3):8–14, 1991.

[6] F. Donini, B. Hollunder, M. Lenzerini, A. M. Spaccamela, D. Nardi, and W. Nutt. The Complexity of Existential Quantification in Concept Languages. Research Report RR-91-02, DFKI Kaiserslautern, 1991.

[7] F. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. In *Proceedings of the 2nd International Conference of Knowledge representation and Reasoning*, Cambridge, Mass., 1991.

[8] F. Donini, M. Lenzerini, D. Nardi, and W. Nutt. Tractable concept languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, 1991.

[9] B. Hollunder. Hybrid inferences in KL-ONE-based knowledge representation systems. In *14th German Workshop on Artificial Intelligence*, volume 251 of *Informatik-Fachberichte*, pages 38–47, Ebingerfeld, Germany, 1990. Springer.

[10] B. Hollunder and F. Baader. Qualifying number restrictions in concept languages. In *Proceedings of the 2nd International Conference of Knowledge Representation and Reasoning*, Cambridge, Mass., 1991.

[11] B. Hollunder, W. Nutt, and M. Schmidt-Schauß. Subsumption algorithms for concept description languages. In *Proceedings of the 9th European Conference on Artificial Intelligence*, pages 348–353, Stockholm, Sweden, 1990.

[12] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Journal of Artificial Intelligence*, 47, 1991.

RAT – Representation of Actions in Terminological Logics

J. Heinsohn, D. Kudenko, B. Nebel, and H.-J. Profitlich

DFKI – Project WIP
Stuhlsatzenhausweg 3
W-6600 Saarbrücken, Germany
email: <*last name*>@dfki.uni-sb.de

Abstract

The system described in this paper is used in the WIP project of DFKI as a tool to represent the domain knowledge. The current prototype of WIP generates multimodal explanations and instructions for assembling, using, maintaining or repairing physical devices. We describe an approach to extend a terminological formalism to cope with some temporal/causal relationships arising when reasoning about actions and plans.

1 Introduction

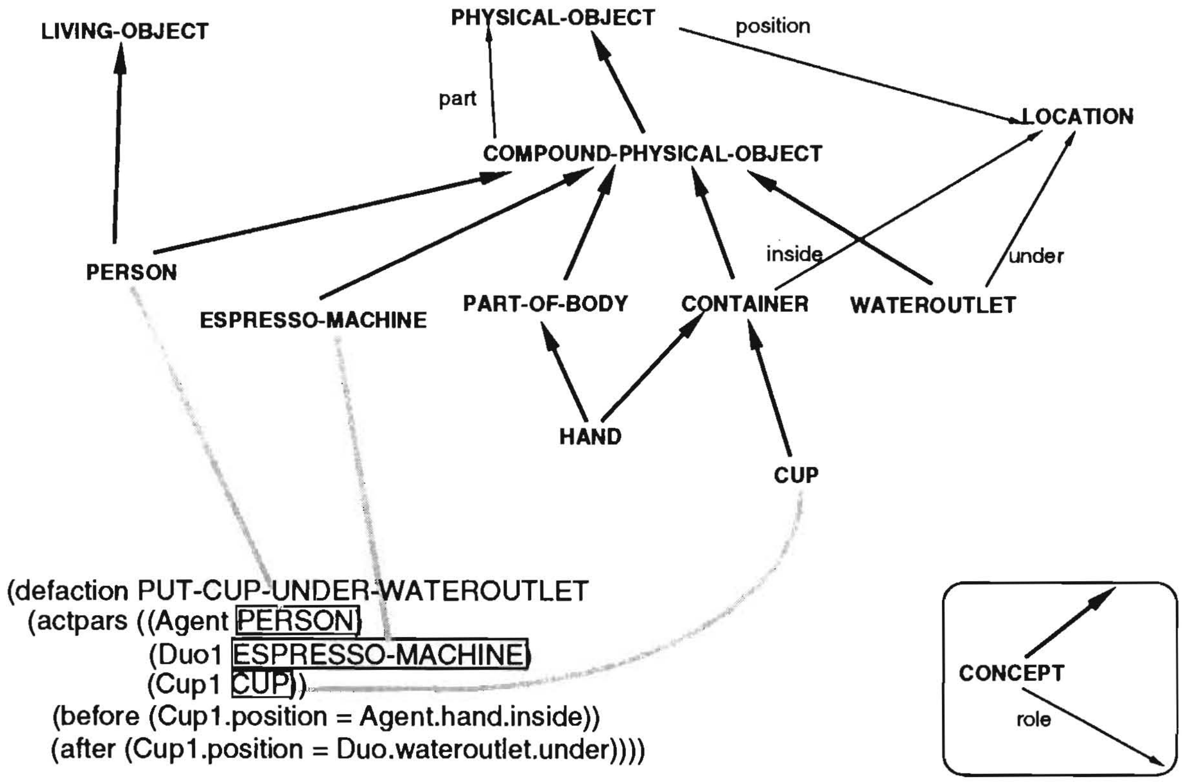
Terminological representation systems have proved to be adequate formalisms for the representation of ontologies in various applications [4]. Beside their abilities of managing concept and instance descriptions, however, they do not provide any meaningful way to represent temporal or causal relationships. The motivation for RAT was the need of a tool to represent and reason about actions and plans. It was designed as an extension of the terminological logic KRIS [1] with close links between action and concept representation.

The RAT system does not only cope with actions and plans but fully integrates the conceptual part of the knowledge representation. In order to have a uniform interface to both, a TELL/ASK-language has been defined that allows the retrieval of any information without having to know in which part the information is represented.

2 Representing domain plans with RAT

In RAT actions are defined by the change of the world state they cause. We distinguish between ***atomic actions***, which are non-decomposable and defined by a pre- and postcondition and ***plan schemata***, which represent sequences of actions with possible constraints on the involved objects. The pre- and postconditions of atomic actions are sets of state descriptions, each defined by a feature agreement or value restriction and denoting a set of world states. The precondition of the atomic action `PUT-CUP-UNDER-WATEROUTLET` in the first example below says that the position of the cup is equal to the inside-region of the hand of the agent. **`position`**, **`has-hand`**, and **`inside-region`** have to be defined as features in the taxonomy (see Figure 1).

The precondition of an action must be satisfied by the current world state to allow the execution of the action. By interpreting the set of state descriptions as a concept description, this can be proved by checking if the subsumption relation holds between the precondition and the current world state. The postconditions are asserted to be valid after the successful execution by interpreting their restrictions on the world state as assignments. Each object that is mentioned in the conditions and is therefore relevant to the action must be declared as *action parameter* by restricting its type to a concept. In the example, `Cup1` is a parameter with its type restricted to `CUP`, which is the name of a concept in the taxonomy.



```
(defaction PUT-CUP-UNDER-WATEROUTLET
  (actpars ((Agent PERSON)
            (Duo1 ESPRESSO-MACHINE)
            (Cup1 CUP))
   (before (Cup1.position = Agent.hand.inside))
   (after (Cup1.position = Duo.wateroutlet.under))))
```

Figure 1: Relations between actions and terminology

Examples for atomic actions in RAT:

```
(defaction PUT-CUP-UNDER-WATEROUTLET
  (actpars ((Cup1 CUP)
            (Agent PERSON)
            (Duo ESPRESSOMACHINE))
   (before (Cup1.position = Agent.has-hand.inside-region))
   (after  (Cup1.position = Duo.has-wateroutlet.under-region))))

(defaction TURN-SWITCH-TO-ESPRESSO
  (actpars ((EM1 ESPRESSOMACHINE)
            (Agent PERSON))
   (before (EM1.state: (and OFF READY))
           (EM1.has-on/off-switch.position: OFF-POSITION))
   (after  (EM1.state: ON)
           (EM1.has-on/off-switch.position: ESPRESSO-POSITION))))
```

A sequence of actions that shall be executed in a fixed order can be summarized in an action sequence or plan schema. In addition to the list of subactions there can be equational constraints between the action parameters of subactions.

Example for a plan schema:

```
(defplan MAKE-ESPRESSO
  (actpars ((EM ESPRESSOMACHINE)
            (Agent USER) ...)
   (sequence (A1 ...)
              ...
             (A5 PUT-CUP-UNDER-WATEROUTLET)
             (A6 TURN-SWITCH-TO-ESPRESSO)
              ...)
   (constraints (equal EM     (A5 Duo) (A6 EM1) ...)
                (equal Agent  (A5 Agent) (A6 Agent) ...)
                 ...))
```

As we want to be able to talk about additional aspects of actions than necessary for the RAT system, each action is also defined as a concept in the terminological part of RAT with its action parameters as attributes. There are two connections between the representation of an action in RAT and the corresponding action concept: first, action parameters that correspond to deep cases must be named accordingly and second, 'defaction' statements may contain references to the predefined action taxonomy[1]:

```
(defaction PUT-CUP-UNDER-WATEROUTLET
    (isa MOTION-WITH-POSITION-CHANGE) ...)
```

---

[1] Not shown in the examples above.

The first approach in extending a terminological logic to cope with plans has been done by Devanbu and Litman [3]. The main focus of their system CLASP, which is build upon the CLASSIC representation system [2], lies in the recognition and management of plan individuals as instances of general plan descriptions and the classification of plans.

Motivated by the varying purposes CLASP and RAT are built for, there are some differences in the design of the systems. The formalism of CLASP, for example, provides a greater variety of operators for composing actions to plans (e.g., loops or conditionals), whereas RAT only allows linear sequences (in the current state). One of the main advantages of RAT, however, is the ability to represent complex state descriptions vs. non-decomposable descriptions in CLASP. Beside that, RAT provides a number of inferential services, which are essential for its application in WIP, e.g., the computation of intermediate world states and the stepwise simulation of plans.

3 Execution of Domain Plans

One of the main inference services provided by RAT is the simulated execution of instantiated domain plans and the closely related specification of intermediate world states. Considering that WIP can also be used as a standalone system and that RAT takes charge of the whole management of domain plans for WIP, RAT is also capable to perform the instantiation of plans itself (Note that the input for the WIP system normally consists of already instantiated plans). The instantiation of actions or plans requires that the given world state satisfies their applicability conditions, i.e., their preconditions. As a plan schema only consists of a sequence of subactions with constraints over the envolved objects, its global preconditions have to be determined by temporal projection of the pre- and postconditions of the subactions (see Figure 2).

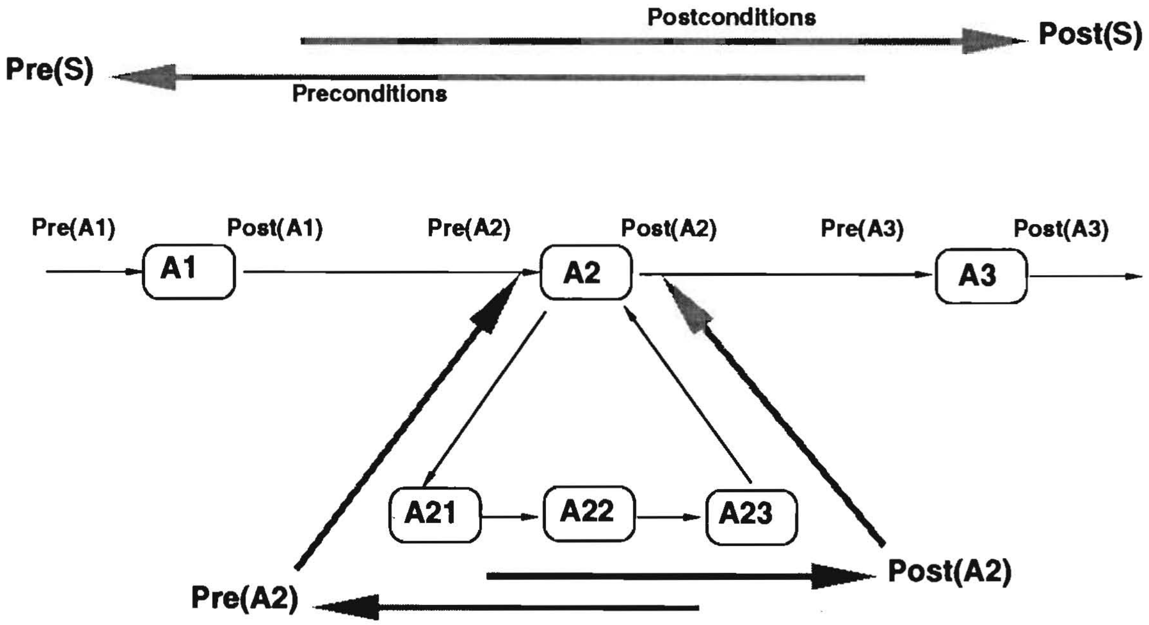


Figure 2: Propagation of Pre- and Postconditions

The temporal projection algorithm takes a plan schema P (i.e., a sequence of subactions A_1 to A_n) as input and computes the global pre- and postconditions of P. If any of the subactions is itself a schema, the function is applied recursively to this subaction. As side effects, the consistency of all intermediate states is proved and the ADD- and DELETE-lists of relations are cached for each subaction.

Since an action may occur in different sequences with varying constraints, the names for action parameters cannot be determined in advance (Exception: deep cases). Furthermore, this freedom in naming parameters has the advantage that while defining an action the names of action parameters can be chosen without considering other actions. To be able to compare state descriptions of different subactions a unification of the names of all involved action parameters has to be done first according to the constraints of the plan schema in which the subactions appear.

Temporal Projection Algorithm:
Some definitions: Let c = (P.F1...Fn $\doteq$ X), `first(c)` = P.F1..Fn, `second(c)` = X, S a set of c's. Then c `member-of` S holds, iff $\exists$ c' $\in$ S such that `first(c')` or `second(c')` is prefix of `first(c)`; S $\ominus$ T = S $\setminus$ {c | $\exists$ c' $\in$ T: first(c') is prefix of first(c)}; S $\oplus$ T = (S $\ominus$ T) $\cup$ T.

```
Function PROPAGATE-CONDITIONS (P);
$Pre_1$ := $Pre(A_1)$;
$PostPre_1$ := $Pre(A_1) \ominus Post(A_1)$;
$PostPost_1$ := $Post(A_1)$;
FOR i := 2 TO n DO
 NewPre := {c | c $\in$ $Pre(A_i)$ and NOT (c member-of $PostPost_{i-1}$)};
 Mentioned := $Pre(A_i)$ without NewPre;
 IF $\exists$d$\in$Mentioned:  NOT (d subsumes $PostPost_{i-1}$)
    THEN Return "not executable";
 $Pre_i$ := union of $Pre_{i-1}$ and NewPre;
 Check if $Pre_i$ is consistent;
 $PostPre_i$ := (union of $PostPre_{i-1}$ and NewPre) $\ominus$ $Post(A_i)$;
 $PostPost_i$ := $PostPost_{i-1}$ $\oplus$ $Post(A_i)$;
 Check if $Post_i$ = (union of $PostPre_i$ and $PostPost_i$) is consistent
OD;
Pre(P) := $Pre_n$;
Post(P) := union of $PostPre_n$ and $PostPost_n$.
```

The intention of the operator $\ominus$ is to forget all information about a feature chain if a prefix of this chain is added. For example, if the current world state contains information about the temperature of the contents of a cup (`CUP.contents.temperature: HIGH`) and an action changes the contents (`CUP.contents = Liquid1`), the information about the temperature may no longer be valid.

The algorithm above maintains three lists: `Pre`, `PostPre`, and `PostPost`. `Pre` contains the global precondition of P. The global postconditions are splitted into `PostPre`, conditions that hold because they are part of a local precondition and are not changed, and `PostPost`, conditions that are asserted by local postconditions. $\texttt{Pre}_k$ represents the most general state of the world that allows the execution of all subactions up to $\texttt{A}_k$. The union of $\texttt{PostPre}_k$ and $\texttt{PostPost}_k$ describes the world state after the execution of $\texttt{A}_k$.

To understand the necessity for splitting the postconditions, regard the following examples. Let $\texttt{A}_i$ and $\texttt{A}_j$ be subactions of a plan, $i < j$, `c` and `c'` conditions, which are mentioned only in $\texttt{A}_i$ and $\texttt{A}_j$, respectively, and which are incomparable, but not disjoint (e.g., `c` says temperature is greater than 30 degrees, `c'` says temperature is less than 50 degrees.). Let the precondition of $\texttt{A}_j$ state that `c'` has to hold.

First, let us regard the case when $\texttt{A}_i$ requires the truth of `c` and does not change this fact. Then `c` is added to `PostPre` and the global precondition `Pre` contains `c` and `c'` which means that the sequence is still executable if the conjunction of them holds (i.e., temperature is between 30 and 50 degrees).

In the second case, let `c` be set by the postcondition of $\texttt{A}_i$. Then `c` is part of `PostPost` and $\texttt{A}_j$ is not executable because, according to the assumptions, `c'` does not subsume the current world state (which contains `c`) and there is no action between $\texttt{A}_i$ and $\texttt{A}_j$ that establishes the conjunction of `c` and `c'`. By seperating these two kinds of postconditions we can differentiate between the two cases.

After having proved the applicability of the plan by checking if its gobal precondition subsumes the current world state we instantiate all subactions of the schema (this is done to be able to talk about them but does not mean that they are already executed).

In the normal case WIP's back end system is responsible for the delivery of instantiated plan schemata, but, nevertheless, it could use RAT itself to perform this task.

Any intermediate state during the execution of the plan can be easily computed by starting from the given world state and exploiting the ADD- and DELETE-lists which have been cached as a side effect of the above algorithm. All references to action parameters occuring in these lists are replaced by references to existing objects in the current world state. The correspondences between them can be explicitly given as parameters of the instantiation procedure.

4 Outlook

The current state of implementation includes the handling of actions and linear action sequences. The ABox part is not yet fully tested because of the unsatisfying performance of the KRIS-ABox. Further steps in the development of RAT will be the inclusion of some forms of conditionals and other kinds of non-linear plans.

References

[1] Franz Baader and B. Hollunder. Kris: Knowledge representation and inference system. *SIGART Bulletin*, 2(3):8–14, 1991.

[2] Ronald J. Brachman, D.L. McGuinness, P.F. Patel-Schneider, L.A. Resnick, and A. Borgida. Living with classic: When and how to use a kl-one-like language. In

J. Sowa, editor, *Principles of Semantic Networks: Explorations in the representation of knowledge.* Morgan Kaufmann, San Mateo, Cal., 1990.

[3] Prem Devanbu and D. Litman. Plan-based terminological reasoning. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 128–138, Cambridge, Mass., 1991.

[4] Peter F. Patel-Schneider, B. Owsnicki-Klewe, N. Guarino A. Kobsa, R. MacGregora, W.S. Mark, D. McGuinness, B. Nebel, A. Schmiedel, and J. Yen. Term subsumption languages in knowledge representation. *The AI Magazine*, 11(2):16–23, 1990.

$\mathcal{ALCP}$ – Representation of Uncertainty in Terminological Logics*

Jochen Heinsohn
DFKI – Project WIP
Stuhlsatzenhausweg 3
D–6600 Saarbrücken 11, Germany
e-mail: heinsohn@dfki.uni-sb.de

Abstract

This paper proposes the language $\mathcal{ALCP}$, a probabilistic extension of terminological logics. While our earlier investigations mainly focus on primitive concept specializations, this paper concentrates on more complex concept expressions and the associated refined probabilistic requirements. We show that several problematic cases identified by Amarger et al. and arising in a purely probabilistic framework can be best handled in a hybrid framework as proposed here.

1 Introduction

In terminological logics [8] the terminological formalism is used to represent a hierarchy of terms (*concepts*) that are partially ordered by a subsumption relation: Concept B is *subsumed by* concept A, if, and only if, the set of B's real world objects is necessarily a subset of A's world objects. The algorithm called *classifier* inserts new generic concepts at the most specific place in the terminological hierarchy according to the subsumption relation. Since, on one hand, the idea of terminological representation is essentially based on the possibility of *categorically describing* concepts, the classifier can be employed to draw correct inferences. Such a terminological framework however excludes the possibility to handle non-categorical concept descriptions involving, e.g., "usually true" properties. On the other hand, purely numerical approaches for handling uncertainty (which are suited to handle non-categorical assertions [7]) in general are unable to consider terminological knowledge and do not provide algorithms comparable to the classifier for maintaining the consistency of the terminology.

We propose the language $\mathcal{ALCP}$ that allows one to handle the problems discussed above and that pursues our earlier investigations [3, 6]. First, we briefly introduce the

---

*This work has been carried out in the WIP project which is supported by the German Ministry for Research and Technology BMFT under contract ITW 8901 8. I would like to thank Bernhard Nebel and Hans-Jürgen Profitlich for helpful comments on earlier versions of this paper.

propositionally complete terminological language $\mathcal{ALC}$. In Section 3 we extend $\mathcal{ALC}$ by defining syntax and semantics of *probabilistic implication*, a construct aimed at considering non-categorical knowledge sources and based on a statistical interpretation. On the basis of terminological and probabilistic knowledge, certain consistency requirements have to be met. While [3] mainly focuses on primitive concept specializations, [5] concentrates on more complex concept expressions and the associated refined probabilistic requirements. We show that several problematic cases identified by Amarger et al. (see Section 8 in [1], e.g.) and arising in a purely probabilistic framework can be best handled in a hybrid framework as proposed here. For proofs and details the reader is referred to [4] where also the related work is discussed.

2 Terminological Languages

The basic elements of the terminological language $\mathcal{ALC}$ ([9], see also [2] for an implementation) are concepts and roles denoting subsets of the domain of interest and binary relations over this domain, respectively. Assume that $\top$ ("top", denoting the entire domain) is a concept symbol, that A denotes a concept symbol, and R denotes a role. Then the concepts (denoted by letters C and D) of the language $\mathcal{ALC}$ are built according to the abstract syntax rule

$$C, D \longrightarrow A \mid \forall R : C \mid \exists R : C \mid C \sqcap D \mid C \sqcup D \mid \neg C$$

A formal semantics of $\mathcal{ALC}$ can be based on a translation into set theoretical expressions with $\mathcal{D}$ being the domain of discourse. For that purpose, we can define a mapping $\mathcal{E}$ that maps every concept description to a subset of $\mathcal{D}$ and every role to a subset of $\mathcal{D} \times \mathcal{D}$ [3]. Concept descriptions are used to state necessary, or necessary and sufficient conditions by means of specializations "$\sqsubseteq$" or definitions "$\doteq$", respectively. Assuming symbol A and concept description C, then "$A \sqsubseteq C$" means the inequality $\mathcal{E}[\![A]\!] \subseteq \mathcal{E}[\![C]\!]$, and "$A \doteq C$" means the equation $\mathcal{E}[\![A]\!] = \mathcal{E}[\![C]\!]$. A set of well formed concept definitions and specializations forms a *terminology*, if every concept symbol appears at most once on the left hand side and there are no terminological cycles. A concept C_1 is said to be *subsumed by* a concept C_2 in a terminology $\mathcal{T}$, iff the inequality $\mathcal{E}[\![C_1]\!] \subseteq \mathcal{E}[\![C_2]\!]$ holds for all extension functions satisfying the equations introduced in $\mathcal{T}$. Basic reasoning services of terminological languages are checking subsumption, e.g., A subsumes C that is defined by $C \doteq A \sqcap B$, and checking equivalence, e.g., $A \doteq C \sqcup D$, $B \doteq \neg(\neg C \sqcap \neg D)$ imply $A \equiv B$.

3 The Probabilistic Extension

Assume the symbol $\mathbf{D}$ for the set of concept descriptions. As language construct that takes into account non-categorical knowledge, we introduce the notion of *conditional probabilistic implication* (p-implication) based on the interpretation as relative cardinalities.

Definition 3.1 *An extension function $\mathcal{E}$ over $\mathbf{D}$ satisfies a p-implication $C_1 \overset{[p_l,p_u]}{\longrightarrow} C_2$, written $\models_{\mathcal{E}} C_1 \overset{[p_l,p_u]}{\longrightarrow} C_2$, iff there exists a value $p \in [p_l, p_u]$ such that $p = \frac{|\mathcal{E}[C_1 \sqcap C_2]|}{|\mathcal{E}[C_1]|}$ holds for concepts $C_1, C_2 \in \mathbf{D}$.*

For simplicity, we assume $p = p_l = p_u$ and write $C_1 \xrightarrow{p} C_2$ instead of $C_1 \overset{[p_l,p_u]}{\longrightarrow} C_2$ for the rest of the paper. Note that because of the set theoretic semantics $B \sqsubseteq A \Rightarrow B \xrightarrow{1} A$ holds.

When representing p-implications, their consistency has to be maintained. First of all we examine the possible relationships between extensions of simple concepts that are introduced by means of the specialization operator "$\sqsubseteq$". The respective consistency requirements can be formulated as follows:

Theorem 3.1 *Assuming concepts A, B, C, p-implications $A \xrightarrow{p} C$, $A \xrightarrow{q} B$, $q \neq 0$, $B \xrightarrow{q'} A$, and $B \xrightarrow{r} C$, then this knowledge is (statistically) inconsistent, if inequality $\frac{q'}{q} \cdot \max(0, q + p - 1) \leq r \leq \min(1, 1 - q' + p \cdot \frac{q'}{q})$ is violated.*

Theorem 3.2 *Assuming concepts A, B, C, p-implications $A \xrightarrow{p} C$, $C \xrightarrow{p'} A$, $A \xrightarrow{q} B$, $B \xrightarrow{q'} A$, and $B \xrightarrow{r} C$, $p', q \neq 0$, then this knowledge is (statistically) inconsistent, if inequality $\frac{q'}{q} \cdot \max(0, q + p - 1) \leq r \leq \min(1, \frac{p}{p'} \cdot \frac{q'}{q}, \frac{p}{p'} \cdot \frac{q'}{q} \cdot (1 - p' + q \cdot \frac{p'}{p}))$ is violated.*

The above theorems cope with concepts introduced by means of simple terminological axioms involving only the specialization operation "$\sqsubseteq$". The associated local consistency requirements, however, have to be strengthened if non-trivial concept expressions are involved. Some simple relations can be formulated as follows (see [4] for other cases):

Theorem 3.3 (Concept Negation and Conjunction)

$$A \xrightarrow{p} B \Leftrightarrow A \xrightarrow{1-p} \neg B\,, \qquad A \xrightarrow{p} \neg A \Rightarrow p = 0\,, \qquad A \xrightarrow{p} C \Leftrightarrow A \xrightarrow{p} A \sqcap C$$

A problem that may arise in a purely probabilistic model is related to the complete handling of logical expressions. For example, Amarger et al. (see Figure 4 in [1]) introduce a network that, as described in [5], in our hybrid framework may be represented by the two terminological knowledge bases given by

$$\text{Concept specialization:} \quad A \sqsubseteq \top,\ B \sqsubseteq \top,\ C \sqsubseteq A,\ C \sqsubseteq B, \text{ and} \tag{1}$$

$$\text{Concept definition:} \quad A \sqsubseteq \top,\ B \sqsubseteq \top,\ D \doteq A \sqcap B \tag{2}$$

It is obvious that because of the set theoretic semantics in case (1) the extension of C remains completely indeterminate except to inequality $\mathcal{E}[\![C]\!] \subseteq \mathcal{E}[\![A]\!] \cap \mathcal{E}[\![B]\!]$. This fact is the main reason for the complexity of Theorems 3.1 and 3.2. More concrete, in case (2) $\mathcal{E}[\![D]\!] = \mathcal{E}[\![A]\!] \cap \mathcal{E}[\![B]\!]$ holds, i.e., concept D can be *proved* to be the *most general concept* that is subsumed by both A and B. Consequently, integrating the two terminological knowledge bases (see Figure 1(a) and (b)[1]) the classifier finds subsumption $C \sqsubseteq D$ (see

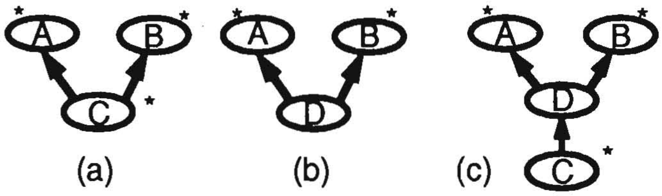


Figure 1: (a) Concept specialization, (b) definition, and (c) integration of both

Figure 1(c)) which further implies the 1-implication $C \xrightarrow{1} D$. This result does not depend on existent p-implications as long as they do not contradict the subsumption relations.

In the following, we show that the specializations introduced in (1) can be simply substituted by 1-implications without changing the obtained result. For that it is necessary to examine the notions of concept conjunction, disjunction, and negation in a more complex probabilistic framework. Note that following proposition applies to a *global*, i.e. non-triangular, case where five concepts are involved and that it can be easily generalized to the situation where $\top$ is substituted by an arbitrary concept C:

Proposition 3.1 (Additivity) *(i) From* $\top \xrightarrow{q_1} A$, $\top \xrightarrow{q_2} B$, *and* $\top \xrightarrow{p} (A \sqcap B)$ *one obtains on the basis of local (triangular) computations the p-implication* $\top \xrightarrow{s} (A \sqcup B)$ *with probability range*

$$\begin{cases} s = q_1 + q_2 - p & \textit{if } p \textit{ is known} \\ \max(q_1, q_2) \leq s \leq \min(1, q_1 + q_2) & \textit{otherwise} \end{cases}$$

(ii) From $\top \xrightarrow{q_1} A$, $\top \xrightarrow{q_2} B$, *and* $\top \xrightarrow{p} (A \sqcup B)$ *one obtains on the basis of local (triangular) computations the p-implication* $\top \xrightarrow{s} (A \sqcap B)$ *with probability range*

$$\begin{cases} s = q_1 + q_2 - p & \textit{if } p \textit{ is known} \\ \max(0, q_1 + q_2 - 1) \leq s \leq \min(q_1, q_2) & \textit{otherwise} \end{cases}$$

In particular, substituting (1) by $C \xrightarrow{1} A$ and $C \xrightarrow{1} B$, and integrating it with (2), from Proposition 3.1(ii) and unknown p we obtain $1 \leq s \leq 1$, i.e., $C \xrightarrow{1} D$. Above inequalities can be simply generalized to ranges as considered in Definition 3.1.

By examining *triangular cases* as in Theorems 3.1 and 3.2, local inconsistencies are discovered early and can be taken into account just in the *current context* of the three concepts involved. Further, not as yet known p-implications can be generated and the associated probability ranges can be stepwise refined. However, testing local consistency requirements *only* for those concepts that are introduced *explicitly* is no guarantee for *global* probabilistic consistency. Such situations require the generation of new concept expressions for which the correct subsumptions are offered by the terminological logic. For instance, testing the additivity axiom formulated in Proposition 3.1 indeed is a *global* test that can be handled on the basis of only triangular cases.

---

[1] Undefined concepts are marked by a star, subsumption relations are denoted by a bold arrow

4 Conclusions

We have proposed the hybrid language $\mathcal{ALCP}$ that generalizes "classical" terminological logics and takes into account uncertain knowledge arising when certain concept properties are, e.g., usually but not categorically true. This theoretical approach has several advantages: $\mathcal{ALCP}$ preserves and extends the features of terminological logics. In particular, it covers terminological knowledge (e.g., complete and partial concept definitions) and offers reasoning services such as subsumption computation (i.e., the "classifier"). On the basis of the new construct ***probabilistic implication*** we are able to consider ***universal*** knowledge that may appear in the form of non-categorical or non-definitional concept properties, generalized quantifiers, or exceptions. Being based on conditional probabilities, consistency can be checked in the current context of three concepts involved, i.e., in the framework of "triangular cases". The associated algorithms provide propagation and refinement of ranges, consider both concept specializations and definitions, and take into account the current terminology.

As sketched above, the natural relations and interactions between the terminological and probabilistic knowledge help to avoid many problems (such as recovering bounds related to additivity) arising in a purely probabilistic framework.

References

[1] S. Amarger, D. Dubois, and H. Prade. Constraint propagation with imprecise conditional probabilities. In *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, pages 26–34, UCLA, Los Angeles, CA, July 1991.

[2] F. Baader and B. Hollunder. KRIS: knowledge representation and inference system. *SIGART Bulletin*, 2(3):8–14, 1991.

[3] J. Heinsohn. A hybrid approach for modeling uncertainty in terminological logics. In R. Kruse and P. Siegel, editors, *Symbolic and Quantitative Approaches to Uncertainty, Proceedings of the European Conference ECSQAU*, Lecture Notes in Computer Science 548, pages 198–205. Springer, Berlin, Germany, 1991.

[4] J. Heinsohn. A hybrid approach for modeling uncertainty in terminological logics. DFKI Report RR-91-24, German Research Center for Artificial Intelligence, Saarbrücken, Germany, Aug. 1991. (22 pages).

[5] J. Heinsohn. ALCP - an integrated framework to terminological and noncategorical knowledge. In *Proceedings of the 4th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'92)*, Palma de Mallorca, Spain, July 6–10 1992. to appear.

[6] J. Heinsohn and B. Owsnicki-Klewe. Probabilistic inheritance and reasoning in hybrid knowledge representation systems. In W. Hoeppner, editor, *Proceedings of the*

12th German Workshop on Artificial Intelligence (GWAI-88), pages 51–60. Springer, Berlin, Germany, 1988.

[7] R. Kruse, E. Schwecke, and J. Heinsohn. *Uncertainty and Vagueness in Knowledge Based Systems: Numerical Methods.* Series Artificial Intelligence. Springer, Berlin, Germany, 1991. (491 pages).

[8] P. F. Patel-Schneider, B. Owsnicki-Klewe, A. Kobsa, N. Guarino, R. MacGregor, W. S. Mark, D. McGuinness, B. Nebel, A. Schmiedel, and J. Yen. Term subsumption languages in knowledge representation. *The AI Magazine*, 11(2):16–23, 1990.

[9] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1), 1991.

Logic Programming and Terminological Inference

Gert Smolka
Project HYDRA

It is a major objective of the Hydra Project to advance towards a framework that is well-suited for knowledge representation. In Hydra's framework special-purpose representations and the accompanying deduction can be integrated as constraint systems. We will concentrate on integrating terminological constraints, but we will also explore the general potential of constraint-based knowledge representation and reasoning in Hydra's framework.

One important task in modeling an application domain in Artificial Intelligence systems is to fix the vocabulary intended to describe the domain—the *terminology*—and to define interrelationships between the atomic parts of the terminology. Representation systems supporting this task are KL-ONE [1] and its descendants [2, 14, 11, 10, 7], which grew out of research in semantic networks and frame systems. These systems allow to describe unary relations, called *concepts*, and binary relations, called *roles*, by means of a restricted class of first-order formulae, called *terminological axioms*. A typical example is:

$$\neg\exists X\ (\mathsf{male}(X) \wedge \mathsf{female}(X))$$

$$\begin{aligned}
\mathsf{woman}(X) &\leftrightarrow \mathsf{human}(X) \wedge \mathsf{female}(X)\\
\mathsf{parent}(X) &\leftrightarrow \mathsf{human}(X) \wedge \exists Y\ (\mathsf{child}(X,Y))\\
\mathsf{sonlessParent}(X) &\leftrightarrow \mathsf{parent}(X) \wedge \forall Y\ (\mathsf{child}(X,Y) \rightarrow \mathsf{female}(Y))\\
\mathsf{grandmother}(X) &\leftrightarrow \mathsf{woman}(X) \wedge \exists Y\ (\mathsf{child}(X,Y) \wedge \mathsf{parent}(Y)).
\end{aligned}$$

A collection of terminological axioms defines a constraint system that has as models the models of the axioms. Given the axioms from above, we can now write constraints like

$$\mathsf{grandmother}(\mathsf{Mary}) \wedge \mathsf{child}(\mathsf{Mary},\mathsf{Tim}) \wedge \mathsf{male}(\mathsf{Tim}),$$

where Mary and Tim are variables.

Given a terminological constraint system, we can define new relations using the principle of definite construction. For instance,

$$\begin{aligned}
\mathsf{son}(X,Y) &\leftrightarrow \mathsf{child}(X,Y) \wedge \mathsf{male}(Y)\\
\mathsf{uncle}(X,Y) &\leftrightarrow \exists U,Z\ (\mathsf{son}(Z,X) \wedge \mathsf{child}(Z,U) \wedge X \neq U \wedge \mathsf{child}(U,Y)).
\end{aligned}$$

Obviously, the concepts **woman**, **parent**, **sonlessParent** and **grandmother** could have been defined by definite construction as well. (To define **sonlessParent** by definite construction, we have to admit the universally quantified constraint $\forall \mathsf{Y}\ (\mathsf{child}(\mathsf{X},\mathsf{Y}) \rightarrow \mathsf{female}(\mathsf{Y}))$.) In fact, for our example it suffices that the underlying terminological constraint system provides the primitive concepts **male**, **female**, and **human**, the primitive role **child**, and the assumption that **male** and **female** are disjoint.

It is an important objective of the Hydra project to investigate the combination of terminological constraint systems with definite construction. We don't know of any published research on this very promising combination. Important questions that need to be attacked are:

- what knowledge is formulated best at the terminological level, and what knowledge is formulated best at the definite level?
- which terminological constraints allow for incremental constraint solving algorithms?
- should complete or incomplete constraint solving algorithms be used?
- how can terminological constraints be combined with other constraints, in particular, the constraints of TFS? (For instance, one may have lists whose elements are parents).

Terminological constraint systems are different from most other constraint systems in that they have many models. For definite construction it makes no difference whether the underlying constraint system has one or many models. However, interesting questions arise with a deep combination of terminological constraints with the constraints of TFS: a tree whose leaves are elements of a terminological model does neither belong to the model of TFS nor to a terminological model.

It is important to note that the combination of terminological constraint systems with definite construction is different from the approach followed in the Krypton system [2, 3, 4], where full function-free Predicate Logic is combined with a terminological system. While Krypton was an attempt to combine general theorem proving with terminological reasoning, our goal is to combine logic programming with terminological reasoning. It is our belief that the rigid structure of definite construction will lead to a feasible combination (which Krypton is not).

If the deep combination of terminological constraints with TFS succeeds, we will arrive at a much richer system than the current stand-alone terminological systems. Even without a deep combination, the computational service coming with definite construction will be much more powerful than the so-called assertional reasoning provided by current systems. Still, our system will rely only on a few but general principles: all that is needed to obtain definite reduction with residuation and guarded rules is a sufficiently powerful constraint solver for terminological constraints.

The basic computational service provided by current terminological systems is *subsumption checking*: given two concepts p and q, does the implication

$$\forall x\ (p(x) \rightarrow q(x))$$

hold in all models of the terminological axioms? Starting with a seminal paper by Brachman and Levesque [5], the computational structure and complexity of this service has been studied carefully for a variety of terminological descriptions of different expressivity [8, 12, 13, 9, 6]. The quintessence of this work is that subsumption checking is computationally intractable even for very weak terminological descriptions.

In our framework we need constraint solving rather than subsumption checking. Moreover, many concepts that are defined with terminological axioms in current systems can be defined with definite construction in our approach. Still, constraint solving for the terminological constraints we need is probably intractable. Should this be the case, our goal will be a tractable incomplete constraint simplifier. It is one of the advantages of our framework that an incomplete constraint simplifier does not affect the soundness of the system.

References

[1] R. J. Brachman and J. G. Schmolze. An overview of the KL-ONE knowledge representation system. **Cognitive Science**, 9(2):171–216, Apr. 1985.

[2] R. J. Brachman, V. Pigman Gilbert, and H. J. Levesque. An essential hybrid reasoning system: Knowledge and symbol level accounts in KRYPTON. In **Proceedings of the 9th International Joint Conference on Artificial Intelligence**, pages 532–539, Los Angeles, CA, Aug. 1985.

[3] R. J. Brachman, R. E. Fikes, and H. J. Levesque. KRYPTON: A functional approach to knowledge representation. **IEEE Computer**, 16(10):67–73, Oct. 1983.

[4] R. J. Brachman and H. J. Levesque. Tales from the far side of KRYPTON. In L. Kerschberg, editor, **Expert Database Systems—Proceedings of the 1st International Conference**, pages 3–43. Benjamin/Cummings, Menlo Park, CA, 1987.

[5] R. J. Brachman and H. J. Levesque. The tractability of subsumption in frame-based description languages. In **Proceedings of the 4th National Conference of the American Association for Artificial Intelligence**, pages 34–37, Austin, TX, Aug. 1984.

[6] F. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. In **Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning**, Cambridge, MA, 1991.

[7] R. MacGregor and R. Bates. The Loom knowledge representation language. Technical Report ISI/RS-87-188, University of Southern California, Information Science Institute, Marina del Rey, CA, 1987.

[8] B. Nebel. Computational complexity of terminological reasoning in BACK. **Artificial Intelligence**, 34(3):371–383, Apr. 1988.

[9] B. Nebel. Terminological reasoning is inherently intractable. IWBS Report 82, IWBS, IBM Deutschland, Stuttgart, West Germany, Aug. 1989. To appear in Artificial Intelligence.

[10] B. Nebel and K. von Luck. Hybrid reasoning in BACK. In Z. W. Ras and L. Saitta, editors, **Methodologies for Intelligent Systems**, volume 3, pages 260–269. North-Holland, Amsterdam, Holland, 1988.

[11] P. F. Patel-Schneider. Small can be beautiful in knowledge representation. In **Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems**, pages 11–16, Denver, Colorado, 1984.

[12] M. Schmidt-Schauß. Subsumption in KL-ONE is undecidable. In R. J. Brachman, H. J. Levesque, and R. Reiter, editors, **Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning**, pages 421–431, Toronto, Canada, May 1989.

[13] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. **Artificial Intelligence**, 47, 1991.

[14] M. B. Vilain. The restricted language architecture of a hybrid representation system. In **Proceedings of the 9th International Joint Conference on Artificial Intelligence**, pages 547–551, Los Angeles, CA, Aug. 1985.

Lexical Choice and Knowledge Representation

Stephan Busemann
Deutsches Forschungszentrum für Künstliche Intelligenz
Project DISCO
Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11
busemann@dfki.uni-sb.de

March 12, 1992

1 Introduction

Recently the problem of choosing communicatively adequate lexemes has attracted much interest in the NL generation community. in general, the task amounts to deciding for a given representation of an intended meaning, which words will most appropriately convey that meaning to the addressee.

Whether lexical choice must be exact in the sense that all and only the intended meaning is verbalized, depends on the respective communication situation. In a multimodal discourse, where language is supplemented by gestures or graphics, the linguistic device need not convey everything to the partner. In written discourse without a predefined context, as in DISCO, exact verbalization seems much more in order.

In all theories of lexical choice, the *convergence problem* has to be solved: there is always a decision for *exactly one* lexical item.

We may distinguish the following subtasks of lexical choice:

Definite reference, proforms: Events and objects must often be described using words that allow for an unambiguous identification of the referent. The problem subdivides in finding appropriate words for the referents and in describing the relations between them, as deictic and intrinsic readings of "The ball is infront of the car" suggest.

Social jugdement: Some words carry social jugdements with them. German *Putzfrau* and *Raumpflegerin* mean both *cleaning woman*, but only the latter is now used officially.[1] The former has a pejorative connotation. See [6].

Collocations: There are different kinds of cooccurrence restrictions between lexemes. Some words cannot be used together with others, some tend to be used together with others and some yield a different meaning when used with certain others (idioms).

---

[1] *Raumpflegerin* reminds at *Krankenpflegerin* (*nurse*).

Choice of open class words: Given a conceptual representation of the intended meaning, an appropriate word for each concept must be identified.

In this paper, we assume that for lexical selection the following kinds of knowledge are necessary:

- the concepts of the meaning representation language
- lexical entries (lemmata and/or phrasal items including semantic and syntactic information, among other things)
- knowledge about the reader (including the reader's goals ans beliefs)
- knowledge about the linguistic, situational, and social context

We will show that lexical choice requires a domain model based on linguistic considerations, and that standard KL-ONE techniques are insufficient for parts of the task at hand.

2 Techniques for the Choice of Open-Class Words

We briefly sketch the techniques employed for the choice of open-class words, as described in [13]. In many generation systems, conceptual knowledge is used: however, the criteria for modelling this knowledge largely differ. From a theoretical point of view, one can represent conceptual knowledge as a theorie of mental categories. In implemented generation systems, however, a model is usually oriented towards the special purpose of the respective system. Obviously the concrete task of choosing open-class items depends on the structure of the underlying knowledge base.[2]

Direct replacement: This still often used technique presupposes a one-to-one relationship between concepts and lexical items. For a given concept (e.g. `TRUCK`) the lexeme *truck* is uniquely determined. This approach does not really deal with lexical choice.

Structural replacement: Partial structures of the meaning representation are identified that match with lexical entities, and the former are replaced by the latter. The procedure terminates if all elements of the meaning representation are replaced by lexical material (cf. [10]).

Classification: The unique lexeme is searched that expresses closest the meaning of the concept to be verbalized. A well-known early approach used decision trees [4] where possible verablizations of a concept (e.g. *eat, drink, breathe* for `INGEST`) are represented as leaf nodes. More recent work uses classification in KL-ONE based systems.

---

[2] Experiences with the modelling task for the LILOG-System are described in [8].

Structural replacement and classification can be combined by using classification techniques during pattern matching. The disadvantage of the techniques presented is that they are restricted to considering the propositional content of the meaning representation. It is difficult to account for the assumed knowledge of the reader, the goals of the author, contextual knowledge and the maxims of conversation [5]. Hence, the choice of the most specific lexeme may lead to a correct but inadequate response (1b2) if *bachelor* is more specific than *man*.

(1) a. Is Kim a woman?
b1. No, Kim is a man.
b2. No, Kim is a bachelor.

3 Demands placed on the model

3.1 Noun choice

Definite reference and pronominalization requires taxonomic reasoning, as the following examples show.

(2) We'll take the big truck and the Mercedes. Both vehicles are available.

(3) The big truck is smashed. The engine doesn't work.

(4) Grind the carots and potatos, cook them to a paste and put them into a prewarmed bowl.

(2) involves generalization (choosing a word whose meaning subsumes those of *truck* and *Mercedes*). It is neither the most specific one (e.g. `MOTOR-VEHICLE`) nor a very general one (e.g. `THING`). Rather, it is the most specific *basic-level class* [14]. To solve this selection problem by classification, a domain model should include a class of basic-level objects.

In (3) the part-of relation holding between `TRUCK` and `ENGINE` allows for a definite description. The representation of the parts of a typical truck is required, but not of additional parts of some special truck in order to avoid the generation of sentences like (5).

(5) ? The small truck of Mr Evens is smashed. The rudder came off.

Example (4) involves pronominalization even though the referents undergo a change of state, as is determined by the verbs *grind* and *cook to a paste*. Obviously a changed aggregate state must not be referred to by the verb, as the following examples suggest.

(6) Melt the ice and put it into a bowl.

(7) * Melt the ice and pour it into a bowl.

The interrelation between event representation and changes of state has been neglected so far, but see [15].

3.2 Adjective choice

Consider the generation of dimensional adjectives such as *high, long, deep.* On the basis of the defining features of e.g. a pole alone it is not possible to verbalize its maximal axis. Whether (8) or (9) is appropriate depends on the pole's position given by the context; only if the pole is upright, (8) is possible.

(8) The pole is 10 m high.

(9) The pole is 10 m long.

[9] suggest a two-staged propositional semantic representation that relates language-independant conceptual entities to lexical entities. Among other things, conceptual features of spatial objects (`POLE` has a maximal axis) are combined with lexical constraints of dimensional adjectives (*high* requires, in contrast to *long*, a vertical orientation of an object's maximal axis).

The selection of dimensional adjectives also depends on the speaker's spatial location. [9] claim that their theory purports to any dimensional expression, thus including prepositions, nouns, and verbs as well.

For this approach to be used in a lexical choice system, the model of spatial knowledge must be geared towards distinguishing the two levels carefully.

4 Inalienable Possessives

Inalienable possessives allow in German definite descriptions without explicit prior introduction of the referent (10).[3] While in (11) the owner of the ladder is verbalized as a genitive attribute, in (10) it is expressed as a dative NP. Exchanging the syntactic construction would lead to inacceptability in both cases.

(10) Hans trat Martha auf den Fuss. [Hans stepped Peter on the foot.]

(11) Hans trat auf Marthas Leiter. [Hans stepped on Martha's ladder.]

There are, however, no clear boundaries between what must go with a dative or with a genitive. (12) seems to be acceptable with both constructions.

(12) Hans trat Martha gegen das Auto. [Hans stepped Martha against the car.]

. A domain model should exhibit the information which objects count as inalienable possessives.

---

[3] The brackets contain an interlinear translation.

5 Demands placed on the model and the formalism

5.1 Verb choice

The event of buying a car (meaning represented as (13)) can be described by lexically converse verbs (14–17). In these examples, all participants in the buying event (buyer, seller, goods, money) can be verbalized, but not all are obligatory. If a participant is not verbalized, its existence in the event being described can be deduced.. As [3] notes, the verbs bring different participants into *perspective*. Verb choice must thus take into consideration which discourse referents are in perspective. This is determined by dynamic context knowledge. [7] demonstrates how lexical and conceptual knowledge represented in one and the same formalism (Ace, extending KODIAK [16]) can be interrelated for verb choice.

```
(13) CommercialExchange(goods: car123, buyer: Peter,
                        seller: John, money: $800)
```

(14) John bought a car from Peter for \$800.

(15) Peter sold a car to John for \$800.

(16) John paid \$800 to Peter for a car.

(17) Peter received \$800 for a car from John.

The domain model must hence be capable of expressing entities in perspective as well as different *views* of some event [7]. A view is a verbalization under certain conditions. For instance, if the seller and the goods are in perspective, the view of (13) as a *selling* is most appropriate. Views relate lexical items to concepts. There is no obvious way how view can be represented in standard KL-ONE dialects [2].

5.2 Noun choice revisited

If the knowledge of dialogue partners is to be considered during lexical choice, we have a domain model[4] and various user-dependant models of how concepts may be verbalized. The system FN [13] helps a user with the decision whether a given object should be used in an action. Should, for instance, a certain flight be used during a journey? FN wants to express that flight ABC3465 lands in La Guardia. It can do this by explicitly saying (18). Alternatively FN could say (19) because FN knows that shuttles typically land in La Guardia. It will depend on the user's knowledge about shuttle flights whether FN will choose the (preferred) shorter version.

(18) Take flight ABC3465 at 11, which lands in La Guardia.

(19) Take the 11 o'clock shuttle.

Since dialogue participants may have different knowledge about shuttles, a lexical choice system should be capable of anticipating (and avoiding) false communicative implicatures

---

[4] Let us assume that this model be shared by all dialogue participants.

[5] (e.g. by choosing *shuttle* if the partner thinks there will a meal be served, as during other flights, but it won't).

These and related issues are implemented using an "overlay" for the domain model that contains all the user-dependent knowledge [13, 12]. This overlay can be exchanged if a different partner is addressed. Implementing such an overlay will extend the standard KL-ONE formalism.

6 Conclusion

By looking at lexical choice in generation we have demonstrated that a number of demands should be placed on a domain model by developers of NL front ends. Application-oriented models usually do not exhibit sufficient knowledge to fulfill these demands. A step in the right direction (but not yet a solution to the problems mentionned) would be the use of an Upper Model [1], which structures the most general part of the knowledge base according to linguistic criteria.

Moreover we have shown how conceptual and lexical knowledge may depend on each other. Two ways of relating these kinds of knowledge can be pursued. One of them keeps the lexicon separate from conceptual knowledge. This requires the definition of complex interfaces and reasoning processes (a choice may fail due to constraints in either the lexicon or the conceptual knowledge). Views would be part of the interface. The alternative is to explore work in lexical semantics (e.g. [11]) which suggests to incorporate taxonomic distinctions into the lexicon. This requires, in practice, the reconstruction of much of what has been done in KL-ONE style knowledge representation work. A concept of views would then involve an extension to the formalism.

It is a yet open question which of the information needed for lexical choice should be provided by the lexicon and which by the domain model.

References

[1] John A. Bateman. Upper modeling: organizing knowledge for natural language processing. In *Proceedings of the 5th International Workshop on Language Generation*, Pittsburgh, PA., 1990.

[2] R. J. Brachman and J. G. Schmolze. An overview of the kl-one knowledge representation system. *Cognitive Science*, 9:171–216, 1985.

[3] C. J. Fillmore. The case for case reopened. In P. Cole and J. M. Sadock, editors, *Syntax and Semantics Vol. 8. Grammatical Relations.* Academic Press, New York, 1977.

[4] N. M. Goldman. Conceptual generation. In R. C. Schank, editor, *Conceptual Information Processing.* North Holland, Amsterdam, 1975.

[5] H.P. Grice. Logic and conversation. In P. Cole and J.L. Morgan, editors, *Speech Acts*, pages 41–58. London, 1975.

[6] E. H. Hovy. Unresolved issues in paragraph planning. In R. Dale, C. Mellish, and M. Zock, editors, *Current Research in Natural Language Generation*, chapter 2, pages 17–46. Academic, London, 1990.

[7] P.S. Jacobs. King: a knowledge-intensive natural language generator. In G. Kempen, editor, *Natural Language Generation*, pages 219–230. Martinus Nijhoff Publishers, Dordrecht, Boston, Lancaster, 1987. NATO ASI Series E Nr. 135.

[8] Gudrun Klose, Ewald Lang, and Thomas Pirlein. Die ontologie und axiomatik der wissensbasis von leu/2. Iwbs report 171, IBM Deutschland, Stuttgart, Mai 1991.

[9] E. Lang, K.-U. Carstensen, and Geoff Simmons. Modelling spatial knowledge on a linguistic basis. IWBS Report 147, IBM Deutschland, Institut f"ur wissensbasierte Systeme, Postfach 80 08 80, 7000 Stuttgart 80, December 1990.

[10] J. Nogier and Michael Zock. Lexical choice as pattern matching. In T. Nagle, J. Nagle, L. Gerholz, and P. Elklund, editors, *Current Directions in Conceptual Structures Research*. Springer, New York, 1991.

[11] James Pustejovsky. The generative lexicon. *Computational Linguistics*, 17(4):409–4413, 1992.

[12] Ehud Reiter and Chris Mellish. Using classification to generate text. In *Proc. Conf. of the 30th Annual Meeting of the ACL*, Newark, 1992.

[13] Ehud B. Reiter. Generating descriptions that exploit a user's domain knowledge. In Robert Dale, Chris Mellish, and Michael Zock, editors, *Current Research in Natural Language Generation*, pages 257–286. Academic Press, 1990.

[14] Eleanor Rosch. Principles of categorization. In E. Rosch and B. Lloyd, editors, *Cognition and Categorization*, pages 27–48. Lawrence Erlbaum, Hillsdale, 1978.

[15] Bonnie L. Webber. Accommodating context change. In *Proc. Conf. of the 30th Annual Meeting of the ACL*, Newark, 1992.

[16] R. Wilensky. Kodiak–a knowledge representation language. In *Proc. 6th Annual Conf. of the Congitive Science Society*, Boulder, Colorado, 1984.

Natural Language Disambiguation and Taxonomic Reasoning

John Nerbonne *
Deutsches Forschungszentrum für Künstliche Intelligenz
DISCO and ASL Projects
Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, Germany
nerbonne@dfki.uni-sb.de

March 6, 1992

Précis: Among the expertises relevant for successful natural language understanding are grammar, semantics and background knowledge, all of which must be represented in order to decode messages from text (or speech). In this extended abstract we examine disambiguation—the choice of one of several hypotheses about the meaning of an input string—as an example of the current cooperation of formalisms. It is quite tempting to provide for a cooperation of background knowledge (represented in taxonomic logic) with grammar representations—and this is an example of heterogeneous formalisms in cooperation. The integration of taxonomic and grammatical information has the advantage of allowing a very close integration of background information with grammar—and therefore an early elimination of some analysis hypotheses, but we argue here that the approach also has problems which could be remedied by a closer integration with semantics. The problems are (i) the range of taxonomic reasoning possible in grammar formalisms, and the apparent usefulness of a more sophisticated reasoning in disambiguation; (ii) the need to incorporate information from discourse content in disambiguation; and (iii) the need to distinguish presupposed from asserted information, which calls for mechanisms familiar in meaning representations but otherwise unneeded in grammar formalisms.

Background

Background for some of the remarks below is a proposal for integrating semantics and grammar formalisms found in Shieber 1986, Pollard and Sag 1987, Fenstad et al. 1987,

---

*Thanks to Rolf Backofen for discussion of the ideas presented here. This work was supported by a research grant, ITW 9002 0, from the German Bundesministerium für Forschung und Technologie to the DFKI DISCO project.

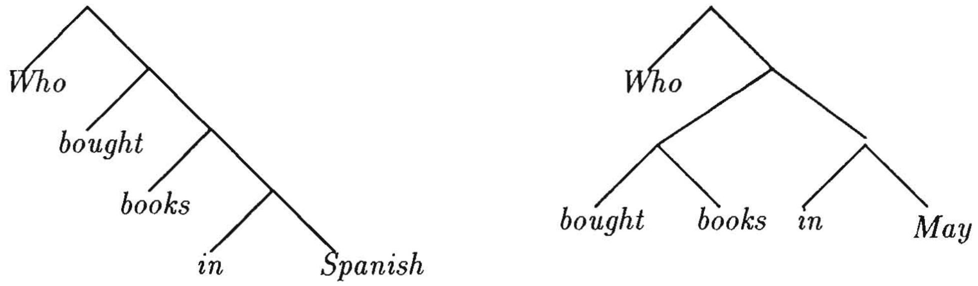


Figure 1: Disambiguation may even require the recognition of distinct constituent structures. Note that the proper names *Spanish* and *May* are not syntactically distinct, nor do that belong to distinct logical types—each denotes an entity. But they do denote objects of different SORTS, as this term is used in sortal logic, since May is a time and Spanish a language. The link to knowledge representation, especially of the kind encoded in KL-ONE, is justified by the emphasis on sorts—taxonomies, or conceptual hierarchies—which distinguishes knowledge representation schemes such as KL-ONE.

and Moore 1989, and elaborated in Nerbonne 1992a, Nerbonne 1992b, and Nerbonne 1992c. The heart of the proposal is a scheme for using feature formalisms as formalized metalanguages for semantic representation languages. In Nerbonne 1992a it is argued that this division between object language and metalanguage specifications is necessary for the characterization of linguistic ambiguity and that it provides a foundation for disambiguation.

Disambiguation

DISAMBIGUATION is the process of determining (i) which of potentially many meanings was intended in an utterance, but also (ii), with respect to a particular application, which facet is relevant to an NL interaction. The former is a response to the ambiguity of natural language, while the latter exists even where no genuine ambiguity does. We illustrate these in turn below. Disambiguation occupies computational linguists more than theoretical linguists, and is extremely important in applications in which there many be uncertainty about input—e.g., speech. Like parsing itself, at least some disambiguation seems to be automatic, so that untrained speakers are not aware of needing to disambiguate structures. The example below, graphed in Figure 1, suggests how unobtrusive the process is:

(1) a. Who bought books in Spanish?
b. Who bought books in May?

This sort of example is convenient because it shows how pervasive the effects of disambiguation may be—reaching even into the parsing component. It is simultaneous misleading if it suggests that genuine disambiguation tasks need to be accompanied by such striking consequences. For even if disambiguation MAY be accompanied by striking consequences in application independent ways, the need for disambiguation arises in NLP interface efforts in ways that need have no purely linguistic ramification whatsoever. In particular, NL interfaces need to be sensitive to application distinctions which do not correspond to natural language ambiguities.

Consider the DISCO application, that of consulting with multiple agents who plan shipping. Here the phrase *Schmidts Ladung* 'Schmidt's freight' certainly denotes freight which stands in some relation to Schmidt. For example, we may imagine the freight contracts in the application as organized into a small database, where the freight contract is the basic tuple.

Order Nr.	Contractor	Agent	Destination	?
457	Schmidt	...	...	...
574	...	Schmidt	...	...
745	...	...	Schmidt	...
475	...	...	...	Schmidt

Thus the phrase *Schmidts Ladung* could designate freight which Schmidt contracted to have shipped, freight for which he is the freight agent, freight being sent to him, and perhaps even freight which stands in yet another relation to him (as owner, inspector, as packer, etc.). Now it is unlikely that the relation expressed by the German possessive construction (genitive + $\bar{\text{N}}$) is ambiguous, and it is unthinkable that the construction is ambiguous to just this degree and in just this fashion.

Taxonomic reasoning of the KL-ONE variety (Brachman and Schmolze 1985, Baader and Hollunder 1990) may fruitfully be applied both to the resolution of linguistic ambiguity and to resolution of application-specific distinctions (Bobrow 1979). We consider these in turn. At the heart of taxonomic reasoning is the imposition of a sort hierarchy on the domain, illustrated in Figure 2 for the case where we found linguistic ambiguity.

In addition to the provision of a sort hierarchy, sortal disambiguation requires a characterization of which sorts are appropriate for which (argument positions of) relations. We would then allow that *in* translate into (at least) two relations, one temporally relating eventualities to times, and the other relating documents to media (but not to times). Schematically:

Lexical Item	relation	Argument1-Sort	Argument2-Sort
in	temporal-loc	Eventuality	Time
	use-medium	Document	Medium

Finally, we must enforce the sortal compatibility restrictions. For many applications, it is desirable to enforce these as early as possible, so that unnecessary processing is avoided. Of course, this mechanism takes us outside KL-ONE proper, but the require information—

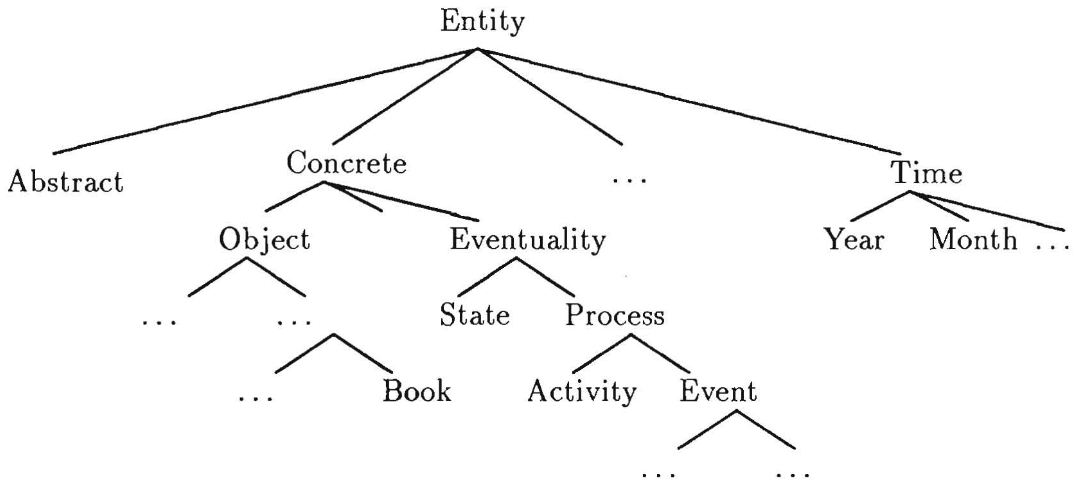


Figure 2: A sort hierarchy which distinguishes enough classes in the domain to illustrate sortal disambiguation for the sentences in (1) (in the text). The hierarchy here is better portrayed not as a tree, but as a directed graph—in which sorts inherit on more than one path back to the root. For example, states and activities might best be viewed as subsorts of both eventualities and nondiscretes, which would include physical substances (water, flour) as well. The move to non-tree-structured hierarchies does not affect the points here however, about the interaction of grammar, semantics and knowledge representations.

compatibility is efficiently provided in KL-ONE-like systems. As the example in Figure 1 suggests, an enforcement of sortal compatibility as early as parse time would be useful (and recall that, e.g., speech applications will rely on disambiguation to prune unlikely hypotheses). This raises the question of how well these constraints can be integrated into other processing—which of course depends on whether they can be expressed in the formalisms of other modules. Here then is a concrete instance of the question of how one relates knowledge representation to grammatical and semantic formalisms.

Earlier work in DISCO demonstrates that semantics can be formulated in an indirect fashion in feature formalisms, so we shall show here that the same is true of knowledge representation—at least within bounds. Cf. Moens et al. 1989 for an earlier proposal along the same lines. That is, once we've taken the step to representing the semantics of *in* in a typed feature description language:

$$\begin{bmatrix} \text{PRED} & \textbf{temporal-location} \\ \text{THEME} & \\ \text{LOCATION} & \end{bmatrix}$$

Then we can also represent the sortal information, relying on unification to enforce sortal compatibility, and thus integrating sortal disambiguation with the unification used in parsing. The following feature structure description represents the ambiguous lexical item *in*:

$$\begin{bmatrix} \text{FORM} & \text{in} \\ \text{SEMANTICS} & \left\{ \begin{matrix} \begin{bmatrix} \text{PRED} & \textbf{temporal-location} \\ \text{THEME} & \begin{bmatrix} \textit{Eventuality} \end{bmatrix} \\ \text{LOCATION} & \begin{bmatrix} \textit{Time} \end{bmatrix} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \text{PRED} & \textbf{expressive-means} \\ \text{THEME} & \begin{bmatrix} \textit{Document} \end{bmatrix} \\ \text{LOCATION} & \begin{bmatrix} \textit{Medium} \end{bmatrix} \end{bmatrix} \end{matrix} \right\} \end{bmatrix}$$

The representation for the word *May*, whose semantics is of the sort Month, and therefore also of the sort Time, can successfully unify with the (location argument of) the first alternative semantics for *in*, but not the last, for which an argument of the sort Medium is expected. Thus the PP *in May* is a seeks to attach where its first argument will be of the sort Eventuality—and this can be a VP attachment, since VP's denote eventualities, but not an NP with the head noun *books*, since this denotes objects of an incompatible sort.

Although we shall not present the details of the treatment of the resolution of application-specific distinctions, it should be clear that the same techniques apply. In the example *Schmidts Ladung*, the relation between Schmidt and the freight is potentially disambiguated by information about whether Schmidt is a shipper, a customer, or the recipient of a customer's shipment. Nor shall we attempt on the basis of this example to argue that sortal restrictions must come from the domain and NOT from the lexicon—the dilemma seems spurious, since the lexicon must in some way be accommodated to the domain for serious applications anyway. Cf. Iida et al. 1989 on the relation between lexicon and disambiguation in complex applications.

Emerging Issues in Disambiguation

Thus the feature formalism allows the integration of constraints from knowledge representation as well. There are several qualifications needed, however. First, the feature formalism cannot faithfully represent all of the sorts of richer KL-ONE-like languages, in particular not those which allow quantified sort defintions, e.g., definitions such as:

$$\text{Parent}(x) \leftrightarrow \exists y.\text{Child-Of}(y, x)$$

Some KL-ONE derivatives allow these without relinquishing decidability, but they are not foreseen in feature formalisms. On the other hand I am unsure of how important this sort of example is—i.e., how frequently one must appeal to sorts of this complexity.

A second qualification is that this sort of treatment will not allow the enforcement of constraints which derive from inferences based on earlier utterances—in order to accomplish this a genuine integration into the semantics representation language would be required. We have in mind the kind of inference possible when information about an individual accumulates during the course of a conversation, but which may be demonstrated even in a single sentence:

(2) Sam talked for two hours in the library and read books for one.

The interesting phrase is the *for one*, and the interesting question is how we account for its VP attachment. Of course, the sortal explanation is available—we simply postulate that *for* denotes a relation between eventualities and durations, but that it denotes no relation between books and durations. But this information cannot be available on the basis of the lexical item one—it must be inferred on the basis of previous content (and the anaphoric link).

One can, as always, attempt alternative explanations, but the ones which immediately come to mind are unconvincing. One could postulate that the choice of attachment site depends on a parallelism to the first clause, but that is not necessary (a). Or one could hypothesize that the VP atachment is strongly preferred. But the $\bar{\mathrm{N}}$-structures of the form $\bar{\mathrm{N}}+$ PP-for are quite possible (b):

(3) a. Two hours elapsed. Sam read books for one, and daydreamed for the other.
b. Sam looked for gifts for his kids. He saw <u>books for one</u>
and <u>T-shirts for the other</u>.

Thus we conclude that a proper account of disambiguation should go beyond the encoding in feature structures illustrated above, and that a more thoroughgoing integration of semantic representation and disambiguating mechanisms is ultimately required. The presentation above stills shows how a great deal of disambiguating information can be integrated into the feature systems and thus arbitrarily deep into a modern NLP system, even if it turns out to be incomplete.

A third qualification about the usefulness of feature-based disambiguation concerns a fundamental pitfall of sortal disambiguation, i.e., that it needs to distinguish between asserted and presupposed sortal information. This is quite clear in the case of application-specific distinctions, and arguably necessary for linguistic ambiguities as well. We examine the case of application-specific distinctions first. We argued above that *Schmidt's freight* might be understood on the basis of a variety of application-specific relations, including 'freight-shipper', 'freight-contractor', etc. In deciding which of these is relevant, it is legitimate to examine the sort to which Schmidt belongs (shipper, contractor, etc.). But notice that in a phrase such as *Schmidt's freight* the relation between Schmidt and the freight is presupposed, not asserted. It would clearly be wrong to apply disambiguation techniques to cases where the relation is asserted or questioned, but not presupposed, e.g., in *Is this freight Schmidt's?* or *If Schmidt sends freight, then his freight will arrive* *today*. (In the latter case, one can imagine blithely disambiguating *his freight* to the 'freight-recipient' relation on the basis of Schmidt's being listed only as recipient—but this would clearly lead to errors.) The case of genuine linguistic ambiguity is similar, but arguably different, in that sortal mismatches remain peculiar enough even in assertion to warrant perhaps being categorized as ill-formed. This tack would regard the following as ill-formed:

> The book is for one hour.
> Is the book for one hour?
> The book cannot be for one hour.
> The book is in May.
> Is the book in May?
> It would be impossible for the book to be in May.

While the examples are undoubtedly peculiar, it still seems wrong to regard them as ill-formed as opposed to unusually formulated or simply concerned with unusual circumstances. Presented with a sentence such as one of these, it would seem that the appropriate reaction would be to try to interpret it metaphorically or, if possible, to clarify it with a user. This conclusion suggests that both sorts of disambiguation—that of resoving genuine ambiguities and that of resolving application-specific distinctions—benefit from the distinction between assertion and presupposition, which therefore ought to be part of a comprehensive disambiguation scheme. We would propose an integration into a logic for presupposition of the sort proposed for dynamic logic (Beaver 1992) or discourse representation theory (van der Sandt and Geurts 1992).

Summary and Conclusions

The purpose of the present abstract has been the investigation of the use of taxonomic information for the purpose of disambiguation. We have argued that taxonomic information is clearly very useful in disambiguation, and that, while it can be incorporated into feature formalisms (allowing an integration of disambiguation with feature unification), still there are several indications that a more thorough-going integration with semantic representation would be sensible.

References

Baader, F., and B. Hollunder. 1990. KRIS: Knowledge Representation and Inference System—System Description. Technical memo, Deutsches Forschungszentrum für Künstliche Intelligenz, Kaiserslautern.

Beaver, D. 1992. Presuppoposition in Update Semantics and 3-Valued Logic. In *Proceedings of the 8th Amsterdam Colloquium*, ed. P. Dekker and J. van der Does, xxx–yyy.

Bobrow, R. 1979. Semantic Interpretation in PSI-KLONE. Annual Report 4274, Bolt, Beranek and Newman, Cambridge, MA.

Brachman, R. J., and J. G. Schmolze. 1985. An overview of the KL-ONE knowledge representation system. *Cognitive Science* 9:171–216.

Fenstad, J. E., P.-K. Halvorsen, T. Langholm, and J. van Benthem. 1987. *Situations, Language, and Logic.* Dordrecht: Reidel.

Iida, M., J. Nerbonne, D. Proudian, and D. Roberts. 1989. Accommodating Complex Applications. In *Proceedings of the First International Workshop on Lexical Acquisition, IJCAI 1989*, xxx–yyy. Menlo Park. Morgan Kaufman.

Moens, M., J. Calder, E. Klein, M. Reape, and H. Zeevat. 1989. Expressing Generalizations in Unification-Based Grammar Formalisms. In *Proceedings of the 4th Meeting of the European Chapter of the Association for Computational Linguistics*, 174–181.

Moore, R. C. 1989. Unification-Based Semantic Interpretation. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, 33–41.

Nerbonne, J. 1992a. Constraint-Based Semantics. In *Proceedings of the 8th Amsterdam Colloquium*, ed. P. Dekker and J. van der Does, xxx–yyy.

Nerbonne, J. 1992b. A Feature-Based Syntax/Semantics Interface. In *Proceedings of the Second International Conference on the Mathematics of Language*, ed. A. Manaster-Ramer and W. Zadrozny, xxx–yyy. Annals of Mathematics and Artificial Intelligence.

Nerbonne, J. 1992c. Representing Grammar, Meaning and Knowledge. In *Proceedings of the Berlin Workshop on Natural Language Processing and Knowledge Representation*, ed. S. Preuss and B. Schmitz. Technische Universität Berlin. KIT FAST.

Pollard, C., and I. Sag. 1987. *An Information-Based Theory of Syntax and Semantics, Vol.I.* Stanford: CSLI.

Shieber, S. 1986. *An Introduction to Unification-Based Approaches to Grammar.* Stanford University: Center for the Study of Language and Information.

van der Sandt, R., and B. Geurts. 1992. Presupposition, Anaphora and Lexical Content. unpublished ms.

Document Analysis: Application Domain for Taxonomic Reasoning?

Michael Malburg, Rainer Bleisinger
Project ALV
E-Mail: {malburg, bleising} @dfki.uni-kl.de

1 Document Analysis in ALV

Document analysis can be seen as an automatic transformation of a paper document into an electronic representation. In the ALV project a model-based document analysis is proposed whereby business letters are treated exemplary.
To enable this analysis various knowledge sources have to be represented in a manner adequate to the problem and supporting the analysis. This knowledge includes a *document model* for a structural description of paper documents. The *layout structure* and the *logical structure* are used in taking pattern from the international standard ODA [5]. The layout structure specifies the hierarchical subdivision of a document page bitmap into areas such as blocks, lines, and connected components. The logical structure defines the hierarchical ordered objects assigned with a human perceptible meaning, e.g. recipient, subject, or date of a letter. Additionally, a classification and description of *message types*, such as order and offer, and a *domain specific world* model is planned which contains addresses of customers as well as products and employees of an individual company.
Within document analysis in ALV four main phases are performed starting after optical scanning of a given document sheet. The first phase, *layout extraction* performs a low-level image processing and segments a letter image with respect to the underlying *layout structure* specified in the document model. In the second phase, called *logical labeling*, layout objects, i.e. rectangular areas of the scanned sheet, are associated with a logical object. For example, the text block in the upper left corner is tagged with the label *recipient*. Up to now, no content based analysis has taken place. But now, the third step in analysis undertakes a *text recognition* whereby the hitherto existing text image is transformed into a stream of characters. Based on these results a *text analysis*, the fourth and last phase, can be started. This step is planned to include several analysis tasks in order to extract important information from the given letter. For example, the identification of the recipient and the pragmatic intention (or the subject) contained in the letter's body shall be obtained. For a detailed description of the ALV system see [3].
By a superficial sight, one can see that different classification techniques are used during the particular document analysis phases. Thus, from a ve point of view, taxonomical reasoning seems to be an adequate support in document analysis, as its superior tasks are

classification, realization and retrieval. For further information on such systems see e. g. [2] describing the system KRIS developed at DFKI.

This paper discusses the hypothetically use of a terminological (concept) language for the representation of the model and especially the usefullness of taxonomic reasoning on this knowledge in document analysis. In particular, the evaluation of KL-ONE like systems for this application is treated.

2 Terminological Document Modeling?

At first, the representation of the ALV document model in a taxonomical logic system (TL-system) is discussed. The layout structure itself is a typical part-of hierarchy describing the composition of words to lines, lines to blocks and so on. Thus, a representation in a KL-ONE like language is certainly possible, but alternatively and far simpler can be done in an object oriented language. Moreover, to each of the layout objects a set of attributes is attached that include geometric features such as position, dimensions, and typographical data. Fortunately, an object oriented language enables all useful numerical operations, whereas most TL- systems do not support the effective use of geometric/numerical features which are fundamental in the layout structure. One approach for integrating such "concrete domains" in TL systems is described by [1].

Additionally, the computation of the subsumption hierarchy for the layout structure P if there were any P is not useful. The layout structure is fix and does not change during analysis, and furthermore, it probably does not change at any time.

Similar statements hold for the logical structure of our model. The logical structure describes the intentional, only content oriented, structure of a document, that means a division into such parts as *salutation*, *recipient* and *signature*. This is, like the layout structure, mainly a part-of hierarchy of logical units, e.g. the *recipient* is constituted of *name, company, address, I* A subsumption hierarchy is not an important component of this structure, although it surely can be forced upon it. Like the layout structure, the logical structure is essentially a fix structure.

3 Document Analysis as Taxonomic Reasoning?

As we have shown above, a representation of the ALV document model in a taxonomic language is possible, but causes some problems. Still, we have not found any argument **for** such a representation. We now take a look on document analysis in hope of finding applications for taxonomic reasoning.

Therefore, in the following we make the assumption that the document model is formulated as a T-Box in a KL-ONE alike. The problems occurring in doing so are neglected generously. Thus, the analysis itself is destined to fill the A-Box.

The most interesting step during layout extraction is that of segmentation. According to the layout structure a hierarchical part-of description of the document is constructed.

This is done by specialized, partially heuristic algorithms that make use of the underlying information associated with the layout structure.

For incorporating taxonomic reasoning, several algorithms are usually offered in terminological systems. Let us exemplarily take a look onto the concistency check algorithm. In our case, the data yielded by segmentation is assumed to be represented in an A-Box, and constructed with respect to the underlying T-Box. Therefore, a concisteny check of the A-Box w.r.t. the T-Box does not make many sense in our case.

Besides, such a representation would involve that each connected component, e.g. the dot of letter RiS, is represented as a concept element of the A-Box - an unsuitable data structure for thousands of pixel clusters.

The instantiation of the logical structure is duty of the logical labeling phase. Only by graphical means, i.e. especially without any text recognition, an assignment of logical meaning to certain layout objects is performed. Thus, the layout block in the upper left is identified as recipient.

Employment of the consistency check for the results of logical labeling fails with the same reasons as for the results of segmentation mentioned above, because the analysis uses T-Box knowledge for building the A-Box.

So far, we have discussed the assumed incorporation of taxonomic reasoning in ALV for checking the *results of analysis.* Now we concentrate on hypothetically support by taxonomic reasoning of the *analysis phases* themselves, namely the process of logical labeling, and finally, the text recognition phase will be described sketchily.

Central knowledge source for logical labeling is a structure called geometric tree. Therein, the arrangement of a letter's logical objects is represented by geometric properties. For example, these are positions, absolute or relative to other objects, and dimensions. For efficient classification of a letter with regard to these features, the geometric tree is used as kind of decision tree. Therefore, it can be mapped to a subsumption hierarchy, and - in principle - be handled by a terminological subsumption algorithm. Suppose, the specific layout structure of the letter at hand is represented in the A-Box, the task of logical labeling may be performed as realization.

But still there remain some differences. The main one concerns the production of multiple results augmented with measures of belief. Moreover, the computation itself is guided by measures of belief for each hypothesis of the input. The integration of uncertainty modeling and terminological logics is part of several research activities (cf. [4] and the related abstract in this volume) but the guidance of realization algorithms by these "probabilities" in general is still an open question. For efficient realization, not only the computation of probabilities but their consideration during realization is important.

One component of the subsequent analysis phase of text recognition, the *character generation* or *recognition*, performs the classification of character images. For a given layout object of type character, i.e. an area of the letter that is assumed to be a character, an extraction of certain features is carried through. These features are the size of the surrounding rectangle (absolute or relative to the line expansion), the degree of blackening, projection to x- and y-axis and thus, ascender and descender etc. As we had an detailed discussion of logical labeling, and by a superficial view, this is a classification problem similar to that of logical labeling, we conclude our consideration with a simple rffisumffi. A formulation in a terminological language for this kind of classification is quite possible, but taxonomical reasoning is not enough and far too much for the requirements of text recognition. The representation of the subsumption hierarchy itself is much easier done by a decision tree and the classification process needs additionally probabilities for hypotheses weighting.

4 Conclusion

As we have seen, an incorporation of terminological reasoning mechanisms, especially with regard to existing systems, into the ALV system is questionable. The results of this negative evaluation fact can be reduced to a few central disagreements.

First, the integration of external, non-terminological knowledge is very important for document analysis. This concerns mainly numerical, but also operational knowledge that directs the analysis itself. By seeing a terminological component as a basis that has to be extented to idiosyncratic needs, this means no refusal, but rises the question why.

Second, there is a big need for modeling uncertainty in connection with terminological constructs enforced by the central philosophy of the ALV- system. At any step of analysis, as we have shown in the context of logical labeling, there are a lot of hypotheses on the results. In the real world we can not say in general that a concept A belongs (as a subconcept or instance) to concept B, but we have expectations on this belonging, i.e. a probability. Uncertain information is the central data to be handled in document analysis.

Third and last, such probabilities have to be respected during analysis, i.e. in particular, an assumed realization (or classification etc.) algorithm for terminologics should be guided by such probabilities, because the whole number of possibilities is to big to deal with. Therefore, a control of taxonomic reasoning, as it should be incorporated into document analysis, considering the probabilities of subconceptualization is an unalterable requirement.

Beside these three statements concerning the question "is it possible?" there remains one question "does it make sense?" that implicitly is answered, too. We have not found a reasonable application of taxonomic reasoning in document analysis as far as we described it here. But there may be applications in that analysis phase which has been omitted in our above description, namely in text analysis. For example, a text skimmer, one evaluation topic of our current activities, disposes of a probably wide spanned terminology. This terminology consists of concepts for message types, like order, offer or invoice, that naturally may be represented in a taxonomical language. Besides, this was the originating domain for KL-ONE alikes and thus, it makes no wonder. Another application would be found in the domain specific world model of ALV wherein for example correspondence partners and the internal structure of a company will be represented.

All in all, a taxonomical reasoning system can not be seen as a general (classification) problem solver that is useful for any application.

References

[1] F. Baader & P. Hanschke. A Schmeme for Integrating Concrete Domains into Concept Languages. DFKI Research Report RR-91-10.

[2] F. Baader & B. Hollunder. KRIS: Knowledge Representation and Inference System. DFKI Technical Memo TM-90-03, November 1990.

[3] A. Dengel, R. Bleisinger, R. Hoch, F. Hönes, F. Fein, M. Malburg. PODA: The Paper Interface to ODA. DFKI Research Report RR-92- 02, February 1992.

[4] J. Heinsohn. A Hybrid Approach for Modeling Uncertainty in Terminological Logics. DFKI Research Report RR-91-24, August 1991.

[5] ISO 8613, Information Processing Q Text and Office Systems Q Office Document Architecture (ODA) and Interchange Format, vol. I-III, parts 1-8, 1988.

Terminological Knowledge Representation for Intelligent Engineering Systems

Philipp Hanschke
Project ARC-TEC
German Research Center for AI (DFKI)
`hanschke@dfki.uni-kl.de`

The ARC-TEC project constitutes an AI approach towards the CIM idea. Along with conceptual solutions, it provides a coherent sequence of software tools for the acquisition, representation, and compilation of technical knowledge [4]. A prototypical production planning application in the domain of lathe workpieces is used to demonstrate the ideas developed (e.g., [6]). Reasoning in this application follows a scheme (Figure 1) that is inspired by William J. Clancey's *heuristic classification*: The input to the system is a CAD drawing describing the workpiece in terms of primitive surfaces and basic technological data. The *abstraction* phase generates a schematic description of the workpiece in terms of *(CAD/CAM) features* [10]. Such features often correspond to characteristic parts of the workpiece with respect to how these parts (or the whole lathe) may be manufactured. Since the features describing a particular workpiece form a DAG[1], the first phase can be seen as a parsing process. The second phase, associates *skeletal (production) plans* to the features (i.e. to the nodes in the feature DAG). Finally, the third phase, *refines* and merges the skeletal plans to a complete numerical control (NC) program.

This abstract tries to extract from this sample application some points on the suitability and the limitations of KL-ONE-like terminological formalisms that appear also to apply in broader contexts such as intelligent engineering systems.

At first glance it seems to be reasonable to use the abstraction power of terminological formalisms to manage the abstraction phase of Figure 1: Define the features in the Tbox of a concept language, represent a particular CAD diagram in the Abox, and obtain the feature description of the workpiece by Abox services such as realization (i.e., classification of objects wrt. the terminology). But we encountered certain problems that enforce some important extensions of terminological formalisms.

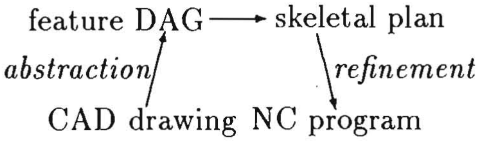


Figure 1: Planning Scheme

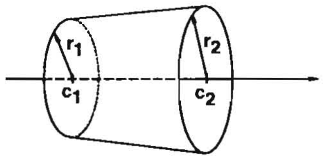


Figure 2: A Truncated Cone

---

[1] I.e., a directed acyclic graph used for representing 'syntax' trees with shared subtrees

Concrete Domains and Role Interaction

Concept terms are inductively constructed from simpler *abstract* terms by means of concept-forming operators ($\sqcap$, $\sqcup$, $\neg$, $\exists R.C$, $\forall R.C$, etc). Most of these constructs do not allow to specify any *interaction of role or attribute*[2] *fillers* in a concept term. In the lathe-work application truncated cones (Figure 2) and their specializations are the most important primitive surfaces. They can be characterized by certain four-tuples of rational numbers. So one would like to write something like

$$\begin{aligned}\mathsf{Truncone} &= \mathsf{truncone\text{-}condition}(\mathsf{r}_1, \mathsf{r}_2, \mathsf{c}_1, \mathsf{c}_2)\\ \mathsf{Cylinder} &= \mathsf{Truncone} \sqcap (\mathsf{r}_1 = \mathsf{r}_2)\end{aligned}$$

Here, the **truncone-condition** could be a four-place predicate over real numbers. It should require that the attribute fillers of r_1, r_2, c_1, and c_2 together characterize a truncated cone and not, e.g., a line or a circle. Thus, specification of attribute interaction as well as reference to concrete notions is mandatory in this application. In [2] an extension scheme for a concept language by concrete domains has been provided. Predicates of the concrete domains applied to tuples of chainings of attributes are the new kind of concept terms that provide the required additional expressiveness. In this formalism it is also possible to define features such as pairs of truncated cones or shoulders.

$$\begin{aligned}\mathsf{Biconic} &= \mathsf{Composed} \sqcap \exists \mathsf{left.Truncone} \sqcap \exists \mathsf{right.Truncone} \sqcap\\ & (\mathsf{left}\ \mathsf{c}_2 = \mathsf{right}\ \mathsf{c}_1) \sqcap (\mathsf{left}\ \mathsf{r}_2 = \mathsf{right}\ \mathsf{r}_1)\\ \mathsf{Right\text{-}shoulder} &= \mathsf{Biconic} \sqcap \forall \mathsf{left.Cylinder} \sqcap \forall \mathsf{right.Ring}\end{aligned}$$

Association, Aggregation, and Derived Attributes

Many features are aggregations or associations of simpler features, of primitive surfaces, or of basic technological data. As these features are found in the abstraction phase, new objects corresponding to these compound features have to be introduced. Conventional assertional formalisms [12] of terminological systems do not provide appropriate reasoning services. It is also not clear from the concept definitions (i.e., the representation of the features) when these object generations are reasonable. Terminological formalisms make no difference between *part-of* attributes and other roles.

With the help of rules, this representation and reasoning deficit can be overcome. The following rule specifies that two truncated cones can be aggregated if they are neighbouring. Note that the aggregate will always be a Biconic.[3]

$$(\mathsf{Truncone}(x) \wedge \mathsf{Truncone}(y) \wedge x.\mathsf{c}_2 = y.\mathsf{c}_1 \wedge x.\mathsf{r}_2 = x.\mathsf{r}_1) \\ \rightarrow \exists z : (x.\mathsf{left} = z \wedge y.\mathsf{right} = z)]$$

A similar problem comes up in conjunction with derived attributes. Sometimes it is clumsy to define specializations of a concept C directly in terms of C's attributes. In this case a new attribute **d**, depending on the existing ones, can help. A specialization D of C can then be defined by restricting the value of the derived attribute. In order to recognize a member x of D in an Abox, an assertion of the form $x.\mathsf{d} = n$, n the derived attribute filler, has to be added to the Abox as soon as x has been recognized as a member of C. This can also be specified by rules, e.g., **length** is the derived attribute in

---

[2] Attributes are functional roles sometimes also called features, not to be confused with CAD/CAM features

[3] $x.\mathsf{c}_2 = y$ stands for $\mathsf{c}_2(x, y)$.

$$\mathsf{Cylinder}(x) \quad \rightarrow \quad \exists l(x.\mathsf{length} = l \wedge l = x.\mathsf{c}_2 - x.\mathsf{c}_1)$$

Finally, using rules, also features with a varying size can be represented. For example, using PROLOG's notation of lists (e.g., $[x, y|r]$), sequences of neighbouring truncated cones can be defined by

$$\begin{array}{rcl} \mathsf{Truncone}(x) & \rightarrow & \mathsf{Tc\text{-}list}([x]) \\ \left.\begin{array}{r} \mathsf{Truncone}(x) \wedge \mathsf{Tc\text{-}list}([y|r]) \wedge \\ x.\mathsf{c}_2 = y.\mathsf{c}_1 \wedge x.\mathsf{r}_2 = y.\mathsf{r}_1 \end{array}\right\} & \rightarrow & \mathsf{Tc\text{-}list}([x, y|r]) \end{array}$$

Representing this kind of features *within* the terminological formalism while keeping the decidability of the terminological inferences cannot be hoped for, because concrete domains together with a transitive closure operator or cyclic definitions in one concept language in general lead to an undecidable subsumption problem [3].

Miscellany

Since the details of a workpiece widely determine its features and there are very many different features, *forward chaining* seems to be the appropriate reasoning strategy for the rules in the abstraction phase [9]. Note that this phase computes a structure. Dually, the refinement phase uses this structure to refine and merge skeletal plans. Thus, *backward chaining* should be employed here. Applying a CLP scheme to the assertional formalism of the concept language with concrete domains [7] provides a nice operational and declarative semantics for the latter.

For two Aboxes $\mathcal{A}$ and $\mathcal{B}$ over a concept language with moderate expressiveness (e.g., attribute agreements and disagreements, value restrictions, and the boolean connectives) the following implication is *undecidable*[4]: $\exists \underline{x}\mathcal{A}(\underline{x}) \Rightarrow \exists \underline{y}\mathcal{B}(\underline{y})$. As a consequence, in general it is undecidable whether the premises of a (forward) rule are satisfied by an Abox and whether one rule of the above kind is more general than another.

It would be helpful for a knowledge engineer to have a component *explaining* why two concepts do not subsume each other or why an object in the Abox is not a member of a certain concept. The necessary information is contained in the models generated by the underlying tuned tableaux calculus.

If one considers the CIM idea and imagines different (large) terminologies of features for several related areas such as CAM and CAD, then interactive tools for *translation*, (*partitioning*) and *merging* of Tboxes (and more general, definitions) would be desirable.

Conclusions

Terminological formalisms are well suited for abstraction processes as far as generalization and classification is concerned. In addition, they provide interesting services (subsumption, disjointness, consistency, etc.) for the knowledge engineer. These capabilities are also useful in the context of constraint propagation over hierarchically structured domains [11]. But in general other abstraction mechanisms such as aggregation and association play an important role [8], too. Together with the problems related to derived attributes and varying-size aspects this demands an integration of terminological representation and reasoning into (Turing complete) formalisms such as rules [5]. And last but not

---

[4] Simple consequence of Section 6 in [1]

least, the development of an environment supporting the knowledge engineer with various knowledge-base analysis and construction tools seems to be necessary for a successful application of terminological systems in the large.

References

[1] F. Baader, H.-J. Bürckert, B. Nebel, W. Nutt, and G. Smolka. On the expressivity of feature logics with negation, functional uncertainty, and sort equations. DFKI Research Report RR-91-01, 1991.

[2] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, 1991.

[3] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. DFKI Research Report RR-91-10, 1991.

[4] A. Bernardi, H. Boley, C. Klauck, P. Hanschke, K. Hinkelmann, R. Legleitner, O. Kühn, M. Meyer, M.M. Richter, F. Schmalhofer, G. Schmidt, and W. Sommer. ARC-TEC: Acquisition, representation and compilation of technical knowledge. In *AVIGNON 91*, 1991. Also as DFKI Research Report RR-91-27.

[5] H. Boley, P. Hanschke, K. Hinkelmann, and M. Meyer. COLAB: A Hybrid Knowledge Compilation Laboratory. Presented at 3rd International Workshop on Data, Expert Knowledge and Decisions: Using Knowledge to Transform Data into Information for Decision Support, Reisensburg, Germany, Sept. 1991.

[6] H. Boley, P. Hanschke, M. Harm, K. Hinkelmann, T. Labisch, M. Meyer, J. Müller, T. Oltzen, M. Sintek, W. Stein, and F. Steinle. μCAD2NC: A declarative lathe-workplanning model transforming CAD-like geometries into abstract NC programs. DFKI Document D-91-15, November 1991.

[7] P. Hanschke. TAXLOG — a logic programming language with deeply-modeled type hierarchies. ARC-TEC DiscussionPaper 91-8, DFKI, June 1991.

[8] P. Hanschke and K. Hinkelmann. Combining terminological and rule-based reasoning for abstraction processes, Forthcoming 92.

[9] K. Hinkelmann. Forward logic evaluation: Developing a compiler from a partially evaluated meta interpreter. DFKI Technical Memo TM-91-13, October 1991.

[10] C. Klauck, R. Legleitner, and A. Bernardi. FEAT-REP: Representing features in CAD/CAM. In *4th International Symposium on Artificial Intelligence: Applications in Informatics*, Cancun, Mexiko, 1991. An extended Version is also available as Research Report RR-91-20, DFKI GmbH.

[11] M. Meyer and C. Jakfeld. How to use CONTAX - a Constraint System over Taxonomies. ARC-TEC Discussion Paper 91-04, DFKI GmbH, P. O. Box 2080, Kaiserslautern, Germany, March 1991.

[12] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems*, volume 422 of *LNAI*. Springer, 1990.

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
D-6750 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all publisched papers so far can be ordered from the above address.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-91-08
Wolfgang Wahlster, Elisabeth André, Som Bandyopadhyay, Winfried Graf, Thomas Rist:
WIP: The Coordinated Generation of Multimodal Presentations from a Common Representation
23 pages

RR-91-09
Hans-Jürgen Bürckert, Jürgen Müller, Achim Schupeta: RATMAN and its Relation to Other Multi-Agent Testbeds
31 pages

RR-91-10
Franz Baader, Philipp Hanschke: A Scheme for Integrating Concrete Domains into Concept Languages
31 pages

RR-91-11
Bernhard Nebel: Belief Revision and Default Reasoning: Syntax-Based Approaches
37 pages

RR-91-12
J.Mark Gawron, John Nerbonne, Stanley Peters:
The Absorption Principle and E-Type Anaphora
33 pages

RR-91-13
Gert Smolka: Residuation and Guarded Rules for Constraint Logic Programming
17 pages

RR-91-14
Peter Breuer, Jürgen Müller: A Two Level Representation for Spatial Relations, Part I
27 pages

RR-91-15
Bernhard Nebel, Gert Smolka:
Attributive Description Formalisms ... and the Rest of the World
20 pages

RR-91-16
Stephan Busemann: Using Pattern-Action Rules for the Generation of GPSG Structures from Separate Semantic Representations
18 pages

RR-91-17
Andreas Dengel, Nelson M. Mattos:
The Use of Abstraction Concepts for Representing and Structuring Documents
17 pages

RR-91-18
John Nerbonne, Klaus Netter, Abdel Kader Diagne, Ludwig Dickmann, Judith Klein:
A Diagnostic Tool for German Syntax
20 pages

RR-91-19
Munindar P. Singh: On the Commitments and Precommitments of Limited Agents
15 pages

RR-91-20
Christoph Klauck, Ansgar Bernardi, Ralf Legleitner
FEAT-Rep: Representing Features in CAD/CAM
48 pages

RR-91-21
Klaus Netter: Clause Union and Verb Raising Phenomena in German
38 pages

RR-91-22
Andreas Dengel: Self-Adapting Structuring and Representation of Space
27 pages

RR-91-23
Michael Richter, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: Akquisition und Repräsentation von technischem Wissen für Planungsaufgaben im Bereich der Fertigungstechnik
24 Seiten

RR-91-24
Jochen Heinsohn: A Hybrid Approach for Modeling Uncertainty in Terminological Logics
22 pages

RR-91-25
Karin Harbusch, Wolfgang Finkler, Anne Schauder: Incremental Syntax Generation with Tree Adjoining Grammars
16 pages

RR-91-26
M. Bauer, S. Biundo, D. Dengler, M. Hecking, J. Koehler, G. Merziger:
Integrated Plan Generation and Recognition
- A Logic-Based Approach -
17 pages

RR-91-27
A. Bernardi, H. Boley, Ph. Hanschke, K. Hinkelmann, Ch. Klauck, O. Kühn, R. Legleitner, M. Meyer, M. M. Richter, F. Schmalhofer, G. Schmidt, W. Sommer:
ARC-TEC: Acquisition, Representation and Compilation of Technical Knowledge
18 pages

RR-91-28
Rolf Backofen, Harald Trost, Hans Uszkoreit: Linking Typed Feature Formalisms and Terminological Knowledge Representation Languages in Natural Language Front-Ends
11 pages

RR-91-29
Hans Uszkoreit: Strategies for Adding Control Information to Declarative Grammars
17 pages

RR-91-30
Dan Flickinger, John Nerbonne:
Inheritance and Complementation: A Case Study of *Easy* Adjectives and Related Nouns
39 pages

RR-91-31
H.-U. Krieger, J. Nerbonne:
Feature-Based Inheritance Networks for Computational Lexicons
11 pages

RR-91-32
Rolf Backofen, Lutz Euler, Günther Görz:
Towards the Integration of Functions, Relations and Types in an AI Programming Language
14 pages

RR-91-33
Franz Baader, Klaus Schulz:
Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures
33 pages

RR-91-34
Bernhard Nebel, Christer Bäckström:
On the Computational Complexity of Temporal Projection and some related Problems
35 pages

RR-91-35
Winfried Graf, Wolfgang Maaß: Constraint-basierte Verarbeitung graphischen Wissens
14 Seiten

RR-92-01
Werner Nutt: Unification in Monoidal Theories is Solving Linear Equations over Semirings
57 pages

RR-92-02
Andreas Dengel, Rainer Bleisinger, Rainer Hoch, Frank Hönes, Frank Fein, Michael Malburg:
Π_{ODA}: The Paper Interface to ODA
53 pages

RR-92-03
Harold Boley:
Extended Logic-plus-Functional Programming
28 pages

RR-92-04
John Nerbonne: Feature-Based Lexicons: An Example and a Comparison to DATR
15 pages

RR-92-05
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner, Michael Schulte, Rainer Stark:
Feature based Integration of CAD and CAPP
19 pages

RR-92-07
Michael Beetz:
Decision-theoretic Transformational Planning
22 pages

RR-92-08
Gabriele Merziger: Approaches to Abductive Reasoning - An Overview -
46 pages

RR-92-09
Winfried Graf, Markus A. Thies:
Perspektiven zur Kombination von automatischem Animationsdesign und planbasierter Hilfe
15 Seiten

RR-92-11
Susane Biundo, Dietmar Dengler, Jana Koehler:
Deductive Planning and Plan Reuse in a Command Language Environment
13 pages

RR-92-13
Markus A. Thies, Frank Berger:
Planbasierte graphische Hilfe in objektorientierten Benutzungsoberflächen
13 Seiten

RR-92-14
Intelligent User Support in Graphical User Interfaces:

1. InCome: A System to Navigate through Interactions and Plans
   Thomas Fehrle, Markus A. Thies
2. Plan-Based Graphical Help in Object-Oriented User Interfaces
   Markus A. Thies, Frank Berger

22 pages

RR-92-15
Winfried Graf: Constraint-Based Graphical Layout of Multimodal Presentations
23 pages

RR-92-17
Hassan Aït-Kaci, Andreas Podelski, Gert Smolka:
A Feature-based Constraint System for Logic Programming with Entailment
23 pages

RR-92-18
John Nerbonne: Constraint-Based Semantics
21 pages

DFKI Technical Memos

TM-91-01
Jana Köhler: Approaches to the Reuse of Plan Schemata in Planning Formalisms
52 pages

TM-91-02
Knut Hinkelmann: Bidirectional Reasoning of Horn Clause Programs: Transformation and Compilation
20 pages

TM-91-03
Otto Kühn, Marc Linster, Gabriele Schmidt:
Clamping, COKAM, KADS, and OMOS: The Construction and Operationalization of a KADS Conceptual Model
20 pages

TM-91-04
Harold Boley (Ed.):
A sampler of Relational/Functional Definitions
12 pages

TM-91-05
Jay C. Weber, Andreas Dengel, Rainer Bleisinger:
Theoretical Consideration of Goal Recognition Aspects for Understanding Information in Business Letters
10 pages

TM-91-06
Johannes Stein: Aspects of Cooperating Agents
22 pages

TM-91-08
Munindar P. Singh: Social and Psychological Commitments in Multiagent Systems
11 pages

TM-91-09
Munindar P. Singh: On the Semantics of Protocols Among Distributed Intelligent Agents
18 pages

TM-91-10
Béla Buschauer, Peter Poller, Anne Schauder, Karin Harbusch: Tree Adjoining Grammars mit Unifikation
149 pages

TM-91-11
Peter Wazinski: Generating Spatial Descriptions for Cross-modal References
21 pages

TM-91-12
Klaus Becker, Christoph Klauck, Johannes Schwagereit: FEAT-PATR: Eine Erweiterung des D-PATR zur Feature-Erkennung in CAD/CAM
33 Seiten

TM-91-13
Knut Hinkelmann:
Forward Logic Evaluation: Developing a Compiler from a Partially Evaluated Meta Interpreter
16 pages

TM-91-14
Rainer Bleisinger, Rainer Hoch, Andreas Dengel:
ODA-based modeling for document analysis
14 pages

TM-91-15
Stefan Bussmann: Prototypical Concept Formation
An Alternative Approach to Knowledge Representation
28 pages

TM-92-01
Lijuan Zhang:
Entwurf und Implementierung eines Compilers zur Transformation von Werkstückrepräsentationen
34 Seiten

DFKI Documents

D-91-04
DFKI Wissenschaftlich-Technischer Jahresbericht 1990
93 Seiten

D-91-06
Gerd Kamp: Entwurf, vergleichende Beschreibung und Integration eines Arbeitsplanerstellungssystems für Drehteile
130 Seiten

D-91-07
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner
TEC-REP: Repräsentation von Geometrie- und Technologieinformationen
70 Seiten

D-91-08
Thomas Krause: Globale Datenflußanalyse und horizontale Compilation der relational-funktionalen Sprache RELFUN
137 Seiten

D-91-09
David Powers, Lary Reeker (Eds.):
Proceedings MLNLO'91 - Machine Learning of Natural Language and Ontology
211 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-91-10
Donald R. Steiner, Jürgen Müller (Eds.):
MAAMAW'91: Pre-Proceedings of the 3rd European Workshop on „Modeling Autonomous Agents and Multi-Agent Worlds“
246 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-91-11
*Thilo C. Horstmann:*Distributed Truth Maintenance
61 pages

D-91-12
Bernd Bachmann:
$H^{iera}C_{on}$ - a Knowledge Representation System with Typed Hierarchies and Constraints
75 pages

D-91-13
International Workshop on Terminological Logics
Organizers: Bernhard Nebel, Christof Peltason,
Kai von Luck
131 pages

D-91-14
Erich Achilles, Bernhard Hollunder, Armin Laux, Jörg-Peter Mohren: KRIS: Knowledge Representation and Inference System
- Benutzerhandbuch -
28 Seiten

D-91-15
Harold Boley, Philipp Hanschke, Martin Harm, Knut Hinkelmann, Thomas Labisch, Manfred Meyer, Jörg Müller, Thomas Oltzen, Michael Sintek, Werner Stein, Frank Steinle:
µCAD2NC: A Declarative Lathe-Worplanning Model Transforming CAD-like Geometries into Abstract NC Programs
100 pages

D-91-16
Jörg Thoben, Franz Schmalhofer, Thomas Reinartz:
Wiederholungs-, Varianten- und Neuplanung bei der Fertigung rotationssymmetrischer Drehteile
134 Seiten

D-91-17
Andreas Becker:
Analyse der Planungsverfahren der KI im Hinblick auf ihre Eignung für die Abeitsplanung
86 Seiten

D-91-18
Thomas Reinartz: Definition von Problemklassen im Maschinenbau als eine Begriffsbildungsaufgabe
107 Seiten

D-91-19
Peter Wazinski: Objektlokalisation in graphischen Darstellungen
110 Seiten

D-92-01
Stefan Bussmann: Simulation Environment for Multi-Agent Worlds - Benutzeranleitung
50 Seiten

D-92-08
Jochen Heinsohn, Bernhard Hollunder (Eds.):
DFKI Workshop on Taxonomic Reasoning Proceedings
56 pages

D-92-09
Gernod P. Laufkötter: Implementierungsmöglichkeiten der integrativen Wissensakquisitionsmethode des ARC-TEC-Projektes
86 Seiten

D-92-21
Anne Schauder: Incremental Syntactic Generation of Natural Language with Tree Adjoining Grammars
57 pages

DFKI Workshop on Taxonomic Reasoning Proceedings

Jochen Heinsohn, Bernhard Hollunder (Eds.)

D-92-08
Document