Three Stories on Automated Reasoning for Natural Language Understanding

Johan Bos
University of Rome "La Sapienza"
bos@di.uniroma1.it

Abstract

Three recent applications of computerised reasoning in natural language processing are presented. The first is a text understanding system developed in the late 1990s, the second is a spoken-dialogue interface to a mobile robot and automated home, and the third is a system that determines textual entailment. In all of these applications, off-the-shelf tools for reasoning with first-order logic (theorem provers as well as model builders) are employed to assist in natural language understanding. This overview is not only an attempt to identify the added value of computerised reasoning in natural language understanding, but also to point out the limitations of the first-order inference techniques currently used in natural language processing.

Introduction

Since the mid 1990s I've been using tools from automated deduction (mainly theorem provers and model builders) to solve problems in natural language understanding. I've done this both from the perspective of *computational linguistics* (testing and improving a linguistic theory by means of a computational implementation) as well as that of *natural language processing* (using inference in applications such as question answering and textual entailment). In this paper I will write about some recent projects that I was involved in, with the aim of convincing researchers — both from the natural language engineering and the automated deduction communities — that computerised reasoning can be successfully applied to natural language understanding.

Right — what's new about this claim? Isn't it obvious that one needs some form of computerised reasoning to model natural language understanding? Indeed, in my opinion it is. However, it is astonishing to see how few tools from automated reasoning have made it in real applications. Why is that? I think there are a couple of reasons for this.

First of all, a rather high level of interdisciplinarity is required. One needs not only to know about linguistics (and in particular (formal) semantics), but also about natural language processing, knowledge representation, and automated inference. As a matter of fact, not many researchers match this profile, and looking back to the 1980s and 1990s, there seem fewer and fewer interested in taking on the pursuit. Or as Steve Pulman put it on the recent ICoS (Inference in Computational Semantics) conference: "I feel like an endangered species."

Secondly, there is an enormous gap between formal linguistic theory and practical implementation. There is a vast amount of formal linguistics theory on the semantics of natural language. However interesting most of it is, it doesn't always lead directly to computational implementation. Many of the phenomena that are accounted for in formal theory are quite rare in natural data, so a natural language engineer won't loose much by ignoring it. But more crucially, from a an automated reasoning point of view, almost all semantic phenomena are formalised in higher-order logic (a trend set by noone less than Richard Montague), which is, as is well-known, a computational nightmare.

Third, it is not trendy to use theorem proving in natural language processing. In contrast, stochastic approaches dominate the field, and after having been successfully applied to speech recognition and syntactic processing, it won't take long until statistics will play a major role in semantic processing. Having said that, time and time again I am surprised by the opinion that using theorem proving in natural language understanding is classified as the "traditional approach." What tradition? OK – it was tried in the 1970s and 1980s, but it never got to work really well, for reasons that should not really surprise us: theorem proving hadn't matured into a state as we know it today, and moreover, trivially, computers lacked the memory and speed to perform the computations required by inference. Yet, we have only started to understand the limitations and opportunities of computerised reasoning in natural language understanding.

After this introduction I will present three applications demonstrating successful use of first-order inference tools. The first is a text-understanding system that calculates presuppositions of sentences and performs consistency and informativeness checks on texts. The second is a spoken dialogue system, interfaced to a mobile robot and an automated home environment, that uses theorem proving and model building for planning its linguistic and non-linguistic actions. The third is a system for recognising textual entailment. In all of these applications I will discuss the reasoning tools used, how they are used, what added value they had, and what their limitations were.

1 Presupposition Projection

In the mid 1990s I started to implement tools for the semantic analysis of English texts, as part of my thesis work at the University of the Saarland in Saarbrücken, Germany. One of the aims was to follow linguistic theory as close as possible and see how much of it could be implemented straight away. I adopted Discourse Representation Theory (DRT), initially developed by Hans Kamp, because it accounted for a wide range of linguistic phenomena in a unified framework, and, crucially, had a model-theoretic semantics [KR93]. In particular I was interested in modelling presupposition projection, and I followed a recent proposal by Rob van der Sandt, whose theory of presupposition was casted in DRT [VdS92].

My implementation efforts resulted in the DORIS system (Figure 1). It had a reasonable grammer coverage (substantially more than a toy grammar, but certainly not reaching the level of today's wide coverage grammars). It parsed English sentences and computed an underspecified discourse representation of the input sentence, followed by resolving ambiguities. The linguistic phenomena covered included pronouns, quantifier and negation scope, and presuppositions [Bos01].

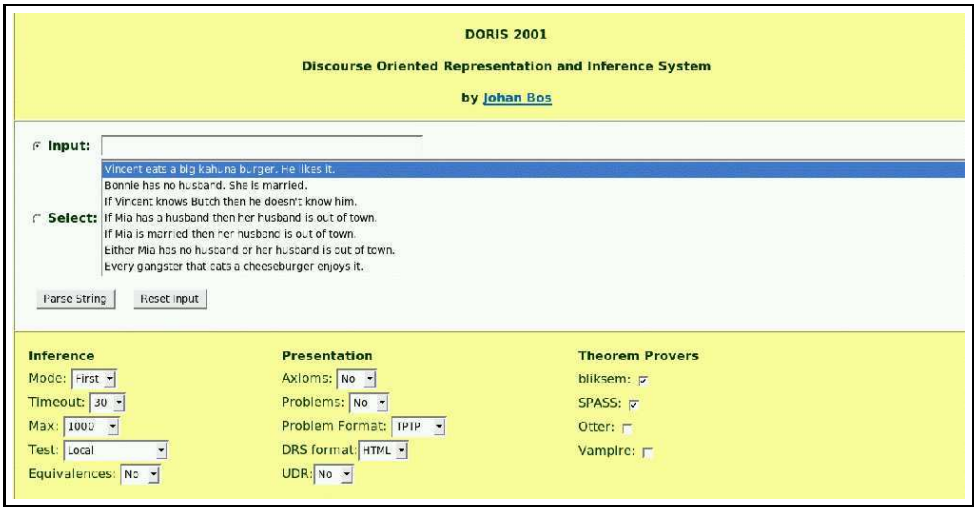

Figure 1: Screenshot of the DORIS system. In the upper window users can type or select an English sentence. The lower window provides several parameters, for instance the selection of various theorem provers.

Presuppositions are propositions taken for granted by the speaker, and "triggered" by words or phrases. Rather informally, p presupposes q if both p and $\neg p$ entail q. For instance, the phrase "Mia's husband" presupposes that Mia is married, and that Mia is a woman, because "Jody likes Mia's husband" and the negation of this sentence "Jody doesn't like Mia's husband" both entail that Mia is married and is a woman. The problem with presuppositions is that they are sometimes neutralised by the linguistic context, and that it is quite hard to pin down exactly when they are and when they are not, especially in sentences that contain some form of implication, negation, or disjunction. Consider, for instance, the following three sentences (with the relevant presupposition trigger typeset in bold face):

(a) If Mia has a date with Vincent, then **her husband** is out of town.
(b) If Mia has a husband, then **her husband** is out of town.
(c) If Mia is married, then **her husband** is out of town.

Here (a) presupposes that Mia is married, but (b) and (c) do not. Van der Sandt's theory explained this by constructing various possibilities of positioning the presupposition, and then checking whether these were acceptable by posing acceptability constraints upon them: consistency as well as informativeness of the resulting text, both on the global and local level of discourse. For (c), there are two positions where the presupposition can "land" (which are underlined):

(c-1) Mia has a husband. If Mia is married, then **her husband** is out of town.
(c-2) If Mia is married/has a husband, then **her husband** is out of town.

However, in the first paraphrase (c-1) the antecedent of the conditional violates the constraint of (local) informativeness: if Mia has a husband, then the fact that she

is married is not new information. In the second paraphrase (c-2) all acceptability constraints are satisfied. As a consequence, (c-1) is rejected, and (c-2) is accepted, as possible interpretation.

Despite the adequacy of the predictions of the theory, there was a still a problem: the acceptability contraints required logical inference. But how cuold you implement this? Even though DRT was an established semantic theory backed up with a model-theory, there were no (efficient) theorem provers available that could reason with the semantic representations employed by DRT. Discussions with Patrick Blackburn and Michael Kohlhase (both in Saarbrücken, at the time) developed the idea of using first-order theorem provers to implement Van der Sandt's acceptability constraints. As the core of DRT is a first-order language, it turned out to be pretty straightforward to translate the DRT representations into ordinary first-order formula syntax, something that first-order theorem provers could digest. Soon after, contacts were made with Hans de Nivelle, whose theorem prover BLIKSEM was among the first that was put to the test [DN98]. And it looked promising: BLIKSEM could handle most of the problems given to it in reasonable time.

However, as some natural language examples could cause hundreds of consistency checking tasks (due to a combinatorial explosion of linguistic ambiguities), it took a long time before BLIKSEM had dealt with them all. Michael Kohlhase and Andreas Franke came to the rescue, by offering MATHWEB, a web-based inference service [FK99]. MATHWEB farmed out a set of inference problems to different machines using a common software bus. Using the internet and many machines around the world (I recall that there were machines running in Edinburgh, Budapest, Saarbrücken, and Sydney, among other sites), MathWeb could basically be viewed as a parallel supercomputer. (This sounds perhaps quite ordinary right now, but at the time it was a sensation.) To cut a long story short, DORIS was interfaced directly to MATHWEB, and many different theorem provers for first-order logic were added: SPASS [WAB+99], FDPLL [Bau00], OTTER [MP96], and VAMPIRE [RV02].

In sum, the DORIS system demonstrated that first-order inference could play an interesting role in natural language processing, albeit with limitations [BBKdN01]. It generated a new application area for automated deduction (in fact, some of the problems generated by DORIS made it to the TPTP collection, thanks to Geoff Sutcliffe), and it opened a whole new vista of research in computational semantics. (Incidentally, it also helped to precisely formulate the acceptability constraints of Van der Sandt's theory of presupposition projection.)

So, what were these limitations? Scalability was one of them. A theorem prover would do well on a couple of sentences, but — not surprisingly given the computational properties of first-order logic — it would just choke on larger texts. Linguistic coverage was another. Some linguistic phenomona require richer semantic representations and therefore harder problems (for instance, tense and aspect require a richer sortal hierarchy, cardinal expression require counting, and plurals require elements of set theory).

The last version of DORIS was released in 2001 [Bos01]. Although it was an important step in the development of computational semantics, its limited grammatical coverage and unfocussed application domain left it without a future. At the time I thought that it would take at least twenty years to develop a parser that achieved both wide-coverage *and* syntactic representations of enough detail to construct meaningful semantic repre-

sentations (I was, fortunately, very wrong! See Section 3). In order to reach a new level of sophistication in computational semantics, I instead focussed on small domains, in which the grammatical coverage and necessary background knowledge could be specified a-priory. Human-computer dialogue systems turned out to be the killer application.

2 Spoken Dialogue Systems

At the University of Edinburgh I was involved in developing a spoken dialogue system which was interfaced with a (real) robot. The robot was a RWI Magellan Pro robot, with sonars, infrared sensors, bumpers, and a video camera. It had an on-board computer connected to the local network via a wireless LAN interface. The robot moved about at the basement of Buccleuch Place, and people could direct it, ask it questions, or provide it with new information, all via speech. A typical conversation could be:

Human: Robot?
Robot: Yes?
Human: Where are you?
Robot: I am in the hallway.
Human: OK. Go to the rest room!

Figure 2: An early version of Godot the talking robot with the roboticist Tetsushi Oka (6 Buccleuch Place, Edinburgh, 2001).

Such kinds of dialogues were relatively straightforward to model with the then state-of-the-art in human-machine dialogue. Yet, the robot was still "semantically challenging": it had no means to draw inferences. What I aimed to do was using components of the DORIS system to give the robot means to do consistency checking, answer questions,

and calculate its next actions. In particular, I was interested in letting the robot react to inconsistent information or obvious information, envisioning dialogues such as:

Human: Where are you?
Robot: I am in the hallway.
Human: You are in my office.
Robot: No, that's not true!
Human: You are in the hallway.
Robot: Yes, I know.

I knew this was feasible because of the DORIS experience: theorem provers can easily handle the amount of information, and the amount of background knowledge, given the limited domain and environment, was easy to compute and maintain. And so it turned out to be: using SPASS as theorem prover, the robot was checking whether each assertion of the user was consistent with its current state and knowledge of the dialogue.

After having implemented this, I was interested in using first-order theorem proving for planning the actions of a directive, primarily from a semantic point of view. I considered commands that involved negation, disjunction, or quantification. Examples of utterances that I had in mind included:

(a) Turn on a light.
(b) Switch in every light.
(c) Switch on every light in the hallway.
(d) Turn off every light except the light in the office.
(e) Go to the kitchen or the rest room.

In (a), the robot had to turn on a light that was currently switched off (and of course it had to complain when all the lights were already on). In (b), it had to turn on all lights that were currently off (some of the lights could be on already). In (c), it should only consider lights in the hallway (restrictive quantification). In (d), it should consider all lights minus those in the office (negation). In (e), it should have either gone to the kitchen or to the rest room (disjunction).

This was hard to do with a theorem prover. It seemed a natural task for a finite model builder though. Via the DORIS system I already came into contact with Karsten Konrad's model builder KIMBA [KW99], but I had never used model builders other than checking for satisfiability. I started using Bill McCune's MACE because it searched models by iteration over domain size, and generally generating models that were both domain-minimal and minimal in the extensions of the predicates [McC98]. Model building was successfully integrated such that the minimal models produced by MACE where used to determine the actions that the robot had to perform [BO02]. The robot in action was regularly demonstrated to visitors at Buccleuch Place (Figure 2) as well as to the general public at the Scottish museum in Edinburgh (Figure 3).

Of course there were the usual limitations. As I was using a first-order language with possible worlds to model the semantics of actions, I was forced to erase the dialogue memory after every second utterance in order to keep the response time acceptable. Also, the number of different objects in the domain was very limited (given the background axioms of the robot, MACE produces models in reasonable time for models up to domain size 20). Nevertheless, the overall system was impressive, and showed what one could do with general purpose, off-the-shelf, first-order inference tools in a practical system.

Figure 3: Godot the talking robot at the Scottish Museum (Edinburgh, 2003).

3 Recognising Textual Entailment

The rapid developments in statistical parsing were of course of interest to the semanticist. Yet most of these parsers produced syntactic derivations that were unsuitable to produce semantic representations in a systematic, principled way. It is not an exaggeration to say that the release of a stastistical wide-coverage parser for CCG (Combinatorial Categorial Grammar) in 2004 corresponded to a breakthrough in computational semantics. This CCG parser, implemented by Stephen Clark and James Curran [CC04], and trained on an annotated treebank developed by Julia Hockenmaier and Mark Steedman [HS02], had the best of both worlds: it achieved wide coverage on texts, and produced very detailed syntactic derivations. Because of the correspondence between syntax and semantic rules in CCG, this framework was the ideal setting for doing semantics.

Because CCG is a heavily lexicalised theory, it has a large number of lexical categories, and very few rules. In order to translate the output of the parser (a CCG derivation) into a DRT representation (which was my main aim, in order to reuse the existing tools that I developed for DRT and inference), I coded a lambda-expression for each of the ca. syntactic 400 categories that were known to the parser. Using the lambda-calculus to produce DRT representations, we simply had a parser that translated newspaper texts into semantic representations, with a coverage of around 95% [BCS$^+$04, Bos05]. This was a great starting point for doing computerised reasoning for natural language on a wider scale.

In the same year the first challenge to recognising textual entailment (RTE) were organised. This is basically a competition for implemented systems to detect whether one (small) text entails another (small) text. To give an impression of the task, consider

an example of a positive and and an example of a negative entailment pair are (where T is the text, and H the hypothesis, using the terminology of the RTE):

Example: 115 (TRUE)

T: On Friday evening, a car bomb exploded outside a Shiite mosque in Iskandariyah, 30 miles south of the capital, killing seven people and wounding 10, doctors said on condition of anonymity.

H: A bomb exploded outside a mosque.

Example: 117 (FALSE)

T: The release of its report led to calls for a complete ivory trade ban, and at the seventh conference in 1989, the African Elephant was moved to appendix one of the treaty.

H: The ban on ivory trade has been effective in protecting the elephant from extinction.

In the RTE challenge a participating system is given a set of entailment pairs and has to decide, for each T-H pair, whether T entails H or not. This turned out to be a very hard task. The baseline (randomly guessing) already gives a score of 50%, as half of the dataset correspond to true entailments, and the other half to false ones. The best systems on the RTE-1 campaign achieved a score approaching 60%.

With the CCG parser and semantics at my disposal, I decided to implement a system that used logical inference to approach the RTE challenge. The overall idea, given the available tools, was straightforward: produce a DRT representation for T and H, translate these to first-order logic, and then use an off-the-shelf prover to check whether $T' \rightarrow H'$. We used VAMPIRE as theorem prover [RV02], motivated by its performance at the recent CASCs.

At first, the performance was limited. The system performed quite well on cases such as 115 above, but unfortunately the RTE challenge doesn't contain many of these. Most of the examples require a lot of background knowledge to draw the correct inference. I used WordNet (an electronic dictionary) to compute some of these background axioms automatically. Yet still there were knowledge gaps. It would be nice to have a theorem prover that would be able to say that it "almost found a proof", instead of just saying "yes" or "(probably) no".

Back to model building. Finite model builders, such as MACE [McC98], said more than just "yes" or "no". They also give you a finite model if there is one. As this model is a clearly defined mathematical entity, it is ideal to simulate the "almost found a proof" scenario. That is, by generating minimal models for T' and for $T' \wedge H'$, we hypothesised that comparing the number of entities of the two models would give us a useful handle on estimating entailment. If the difference is relatively small, it is likely that T entails H. Otherwise it is not. (To deal with negation, one also has to calculate the models for $\neg T'$ and $\neg(T' \wedge H')$ and compare the model sizes. To deal with disjunction, one has to do this for all minimal models for T and H — but as of now it is unclear to me how to implement the latter in a neat way...)

This turned out to work well — not as good as one would hope (because it is hard to get the right background knowledge), but significantly better than the baseline. We used standard machine learning techniques to estimate the thresholds of the model sizes. We used both the size of the domain as well as the number of instantiated relations in the model. It turned out that MACE was quite slow for longer sentences. We tried

PARADOX [CS03], which is faster, but it does not always return minimal models (with respect to the predicate extensions). To overcome this, we used PARADOX to calculate the domain size, and then called MACE to generate the model given that domain size.

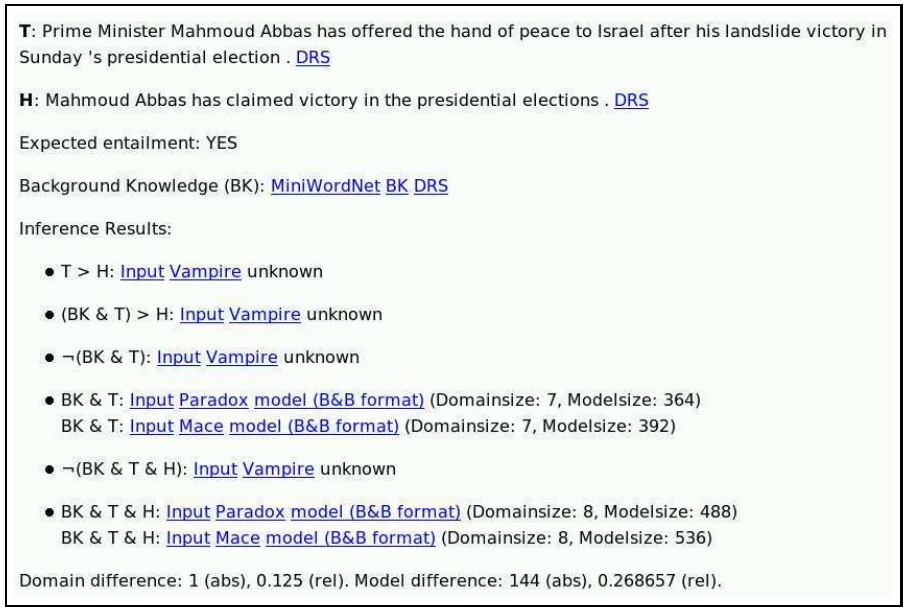

Figure 4: Example output screen for recognising textual entailment using the theorem prover VAMPIRE and the model builders MACE and PARADOX.

Conclusion

First-order inference tools, such as automated theorem provers and model builders, can be successfully used in natural language understanding applications. Obviously, there are limitations, but in many interesting applications these limitations play a subordinate role. Whether computerised (logical) reasoning will ever become part of mainstream research in natural language processing is questionable, though. We will have to see to what extent statistical approaches (currently dominating computational linguistics) can be applied to natural language understanding tasks. Meanwhile, collaboration between researchers working in automated deduction and computational linguistics should be stimulated to get a better understanding of the boundaries of applying automated reasoning to natural language understanding.

References

[Bau00] Peter Baumgartner. FDPLL – A First-Order Davis-Putnam-Logeman-Loveland Procedure. In David McAllester, editor, *CADE-17 – The 17th International Conference on Automated Deduction*, volume 1831 of *Lecture Notes in Artificial Intelligence*, pages 200–219. Springer, 2000.

[BBKdN01] Patrick Blackburn, Johan Bos, Michael Kohlhase, and Hans de Nivelle. Inference and Computational Semantics. In Harry Bunt, Reinhard Muskens, and Elias Thijsse, editors, *Computing Meaning Vol.2*, pages 11–28. Kluwer, 2001.

[BCS+04] J. Bos, S. Clark, M. Steedman, J.R. Curran, and Hockenmaier J. Wide-Coverage Semantic Representations from a CCG Parser. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04)*, Geneva, Switzerland, 2004.

[BO02] Johan Bos and Tetsushi Oka. An Inference-based Approach to Dialogue System Design. In Shu-Chuan Tseng, editor, *COLING 2002. Proceedings of the 19th International Conference on Computational Linguistics*, pages 113–119, Taipei, Taiwan, 2002.

[Bos01] Johan Bos. DORIS 2001: Underspecification, Resolution and Inference for Discourse Representation Structures. In Patrick Blackburn and Michael Kohlhase, editors, *ICoS-3, Inference in Computational Semantics*, pages 117–124, 2001.

[Bos05] Johan Bos. Towards wide-coverage semantic interpretation. In *Proceedings of Sixth International Workshop on Computational Semantics IWCS-6*, pages 42–53, 2005.

[CC04] S. Clark and J.R. Curran. Parsing the WSJ using CCG and Log-Linear Models. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL '04)*, Barcelona, Spain, 2004.

[CS03] K. Claessen and N. Sörensson. New techniques that improve mace-style model finding. In *Model Computation – Principles, Algorithms, Applications (Cade-19 Workshop)*, Miami, Florida, USA, 2003.

[DN98] Hans De Nivelle. A Resolution Decision Procedure for the Guarded Fragment. In *Automated Deduction - CADE-15. 15th International Conference on Automated Deduction*, pages 191–204. Springer-Verlag Berlin Heidelberg, 1998.

[FK99] Andreas Franke and Michael Kohlhase. System description: Mathweb, an agent-based communication layer for distributed automated theorem proving. In *CADE'99*, 1999.

[HS02] J. Hockenmaier and M. Steedman. Generative Models for Statistical Parsing with Combinatory Categorial Grammar. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA, 2002.

[KR93] H. Kamp and U. Reyle. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT.* Kluwer, Dordrecht, 1993.

[KW99] Karsten Konrad and D.A. Wolfram. System description: Kimba, a model generator for many-valued first-order logics. In *16th International Conference on Automated Deduction CADE-16*, 1999.

[McC98] W. McCune. Automatic Proofs and Counterexamples for Some Ortholattice Identities. *Information Processing Letters*, 65(6):285–291, 1998.

[MP96] W. McCune and R. Padmanabhan. *Automated Deduction in Equational Logic and Cubic Curves*. Lecture Notes in Computer Science (AI subseries). Springer-Verlag, 1996.

[RV02] A. Riazanov and A. Voronkov. The Design and Implementation of Vampire. *AI Communications*, 15(2–3), 2002.

[VdS92] R.A. Van der Sandt. Presupposition Projection as Anaphora Resolution. *Journal of Semantics*, 9:333–377, 1992.

[WAB+99] Christoph Weidenbach, Bijan Afshordel, Uwe Brahm, Christian Cohrs, Thorsten Engel, Enno Keen, Christian Theobalt, and Dalibor Topic. System description: Spass version 1.0.0. In Harald Ganzinger, editor, *16th International Conference on Automated Deduction, CADE-16*, volume 1632 of *LNAI*, pages 314–318. Springer-Verlag, Berlin, 1999.