Automatic Belief Revision in SNePS*
CSE Technical Report 2000-01

Stuart C. Shapiro and Frances L. Johnson
Department of Computer Science and Engineering
and Center for Multisource Information Fusion
and Center for Cognitive Science
State University of New York at Buffalo
226 Bell Hall
Buffalo, NY 14260-2000, USA
{shapiro | flj}@cse.buffalo.edu

Abstract

SNePS is a logic- and network- based knowledge representation, reasoning, and acting system, based on a monotonic, paraconsistent, first-order term logic, with compositional intensional semantics. It has an ATMS-style facility for belief contraction, and an acting component, including a well-defined syntax and semantics for primitive and composite acts, as well as for "rules" that allow for acting in support of reasoning and reasoning in support of acting. SNePS has been designed to support natural language competent cognitive agents.

When the current version of SNePS detects an explicit contradiction, it interacts with the user, providing information that helps the user decide what to remove from the knowledge base in order to remove the contradiction. The forthcoming SNePS 2.6 will also do automatic belief contraction if the information in the knowledge base warrents it.

General Information

Platforms and Language

The current version of SNePS is written in ANSI Common Lisp, and runs on any platform that runs ANSI Common Lisp.

Size

To install SNePS, one needs about 10 Megabytes of disk space. Once the installation is completed, this might be trimmed to about 5 Megabytes by compressing/deleting Lisp source files and/or documentation.

Additional Information

More information may be found at the following URLs:

`http://www.cse.buffalo.edu/sneps/` home page of the SNePS Research Group

`http://www.cse.buffalo.edu/sneps/Bibliography/` a bibliography of over 270 papers on SNePS

`http://www.cse.buffalo.edu/sneps/Manuals/` a repository of SNePS Manuals

`ftp://ftp.cse.buffalo.edu/pub/sneps/` the SNePS ftp site

At the time of writing this paper, the most current available version of SNePS is SNePS 2.5. This paper also describes SNePS 2.6, which will be the next release. The major new feature of SNePS 2.6 is that, under certain circumstances, it will perform automatic belief revision (Johnson & Shapiro 2000).

Description of the System

SNePS is a logic- and network- based knowledge representation, reasoning, and acting system that has been developed over the course of the last thirty years by the author and over 60 students and colleagues (Shapiro & Rapaport 1992; Shapiro & The SNePS Implementation Group 1999). SNePS has been designed to support natural language competent cognitive agents. Its logic is based on Relevance Logic (Shapiro 1992), a paraconsistent logic (in which a contradiction does not imply anything whatsoever). The basic principles of SNePS are:

Propositional Semantic Network: The only well-formed SNePS expressions are nodes.

Term Logic: Every well-formed SNePS expression is a term.

Intensional Representation: Every SNePS term represents (denotes) an intensional (mental) entity.

Uniqueness Principle: No two SNePS terms denote the same entity.

*The development of SNePS was supported in part by: the National Science Foundation under Grant IRI-8610517; the Defense Advanced Research Projects Agency under Contract F30602-87-C-0136 (monitored by the Rome Air Development Center) to the Calspan-UB Research Center; by the Air Force Systems Command, Rome Air Development Center, Griffiss Air Force Base, New York 13441-5700, and the Air Force Office of Scientific Research, Bolling AFB DC 20332 under Contract No. F30602-85-C-0008, which supported the Northeast Artificial Intelligence Consortium (NAIC); NASA under contract NAS 9-19335 to Amherst Systems, Inc.; and by ONR under contract N00014-98-C-0062 to Apple Aid, Inc. The development of automatic belief revision was supported in part by the U.S. Army CECOM through a contract with CACI Technologies.

The following, more system-oriented, description is a slight rewriting of the Introduction to the SNePS 2.5 User's Manual (Shapiro & The SNePS Implementation Group 1999).

SNePS (the Semantic Network Processing System) is a system for building, using, and retrieving information from propositional semantic networks. A semantic network, roughly speaking, is a labeled directed graph in which nodes represent entities, arc labels represent binary relations, and an arc labeled R going from node n to node m represents the fact that the entity represented by n bears the relation represented by R to the entity represented by m.

SNePS is called a *propositional* semantic network because every proposition represented in the network is represented by a node, not by an arc. Relations represented by arcs may be thought of as part of the syntactic structure of the node they emanate from. Whenever information is added to the network, it is added in the form of a node with arcs emanating from it to other nodes.

Each entity represented in the network is represented by a unique node. This is enforced by SNePS 2 in that whenever the user specifies a node to be added to the network that would look exactly like one already there, in the sense of having the same set of arcs going from it to the same set of other nodes, SNePS 2 retrieves the old one instead of building the new one.

The core of SNePS 2 is a system for building nodes in the network, retrieving nodes that have a certain pattern of connectivity to other nodes, and performing certain housekeeping tasks, such as dumping a network to a file or loading a network from a file.

SNIP, the SNePS Inference Package, interprets certain nodes as representing reasoning rules, called *deduction rules*. SNIP supports a variety of specially designed propositional connectives and quantifiers, and performs a kind of combined forward/backward inference called *bi-directional* inference.

SNeBR, the SNePS Belief Revision system (Martins & Shapiro 1988), recognizes when a contradiction exists in the network, and interacts with the user whenever it detects that the user is operating in a contradictory belief space. Under certain circumstances, SNePS 2.6 will perform automatic belief contraction (*see* (Johnson & Shapiro 2000)).

SNeRE, The SNePS Rational Engine, is a package that allows for the smooth incorporation of acting into SNePS-based agents, including acting in the service of inference and *vice versa* (Kumar & Shapiro 1994; Kumar 1996).

SNePSUL, the SNePS User Language, is the standard command language for using SNePS. It is a Lispish language, usually entered by the user at the top-level SNePSUL read-eval-print loop, but it can also be called from Lisp programs.

SNePSLOG is a logic programming interface to SNePS, and provides direct access in a predicate logic notation to almost all the facilities provided by SNePSUL. A Tell/Ask interface allows SNePSLOG expressions to be used within normal Lisp programs.

SNaLPS, the SNePS Natural Language Processing System, consists of a morphological analyzer, a morphological synthesizer, and a Generalized Augmented Transition Network (GATN) Grammar interpreter/compiler (Shapiro 1982). Using these facilities, one can write natural language (and other) interfaces for SNePS.

XGinseng is an X Windows-based graphical editing and display environment for SNePS networks. XGinseng is the best environment to use for preparing diagrams of SNePS networks for inclusion in papers. It can also be used to build and edit SNePS networks. (See the screen shot at `http://www.cse.buffalo.edu/sneps/screen.gif`)

Applying the System

Methodology

Problems are encoded in SNePS logic, which can express any formula expressible in first-order logic, but also contains features specifically designed for a network-oriented KRR system for natural language competence and commonsense reasoning (Shapiro 2000), such as set-oriented (instead of binary) connectives and numerical quantifiers. SNePS is an intensional term-logic, meaning, in part, that propositions are denoted by functional terms, so propositions may be arguments of propositions without the need for quotation, modal logic, or leaving first-order logic (Shapiro 1993).

To take full advantage of automatic belief revision in SNePS 2.6, the user should give the system information about the sources of information, should order the sources by credibility, and may provide credibility ordering of the information directly. The following shows some information, source information, source credibility ordering information, and direct information credibility ordering as it might be given to the system using the SNePSLOG interface:

```
fun(learning).
~fun(spitting).

Source(Lisa, fun(learning)).
Source(Lisa, ~fun(spitting)).
Source(Bart, fun(spitting)).

Sgreater(Lisa,Marge).
Sgreater(Marge, Bart).
Sgreater(Bart,Homer).

Greater(fun(learning),~fun(spitting)).
```

Specifics

Significance of Being Logic-Based SNePS is a knowledge representation, reasoning, and acting system. We believe that every knowledge representation and reasoning formalism must have a well-defined syntax, a well-defined semantics, and a well-defined inference mechanism, to implement reasoning, that is sound with respect to the semantics. Thus, we believe that every knowledge representation and reasoning formalism is a logic, although it might not be standard classical first-order predicate logic.

Semantics SNePS is a paraconsistent, first-order term logic, with compositional intensional semantics. It is currently monotonic, although it has an ATMS-style facility for belief contraction—removal of an assumption from the belief space along with all derived beliefs that thereby loose all their supports. It also has an acting component, including a well-defined syntax and semantics for primitive and composite acts, as well as for "rules" that allow for acting in support of reasoning and reasoning in support of acting.

Importance SNePS has been designed, and continues to be developed so that a SNePS knowledge base can form the mind of a natural language competent cognitive agent. The features mentioned above are all in support of this purpose.

SNePS is paraconsistent so that it can continue to reason, even while containing contradictions, without a contradiction in one area of its beliefs "polluting" its conclusions in an unrelated area. This is based on the fact that people, likewise, can have contradictory beliefs without assenting to every question.

SNePS is first-order to make use of well-known first-order inference techniques. Nevertheless, the "end-user" uses a language that includes only the individual symbols of SNePS, so that language looks higher-order. (*See* (Shapiro *et al.* 1981).)

SNePS is a term logic, meaning that every well-formed expression is a term in the language—there are no sentential-level expressions. For example, since function symbols can take functional terms as arguments, and propositions are considered to be first-class individuals in the domain, denoted by terms, propositions can take propositions as arguments without the need for quotation, modal logic, or leaving first-order logic. (*See* (Shapiro 1993).) This was illustrated near the end of the subsection on methodology by the SNePSLOG expression `Source(Lisa, fun(learning))`, where the functional term `fun(learning)` denotes the proposition that learning is fun, and the functional term `Source(Lisa, fun(learning))` denotes the proposition that Lisa is the source of the information that learning is fun. Again, this is based on the fact that people talk about propositions, treating them as individuals in the domain of discourse.

SNePS has a compositional semantics for the usual reason that a single term can be included as an argument in multiple functional terms while maintaining a single denotation.

Every SNePS term denotes an intensional, or mental, entity, because SNePS knowledge bases are intended to serve as minds of cognitive agents, and no two entities that are conceptually different are identical. Even the equation $2 + 3 = 5$ is informative only if $2 + 3$ and 5 denote conceptually distinct entities. (*See* (Maida & Shapiro 1982; Shapiro & Rapaport 1985).)

The current version of SNePS is monotonic, although previous versions contained default rules, and we intend to reintroduce them in a future version.

Although SNePS is paraconsistent, we believe that when a contradiction becomes explicit, the user should be afforded the opportunity of removing the contradiction by removing some hypothesis that underlies it. An explicit contradiction, the presence of both some proposition `P` and its negation, `~P`, in the belief space, is easily recognized by the system because, in accordance with the Uniqueness Principle, the data structure representing `P` is directly pointed to by the negation operator in the data structure representing `~P`. In SNePS 2.6, when the choice of which hypotheses to remove is "obvious", the system will do so automatically, and notify the user of the hypothesis removed and of the other beliefs that are no longer in the belief space because they are no longer supported.

Since SNePS is designed for cognitive agents, it is important for acting and reasoning to be integrated. For example, if `light1` denotes some traffic light, `street1` the street where that traffic light is, `green(`x`)` the proposition that x is green, and `cross(`x`)` the act of walking across x, then `whendo(green(light1), cross(street1))` denotes the proposition that when the agent comes to believe that the traffic light is green, it should cross the street. If, moreover, `lookat(`x`)` denotes the act of looking at x, then `ifdo(green(light1), lookat(light1))` denotes the proposition that if the agent wants to know whether to believe that the traffic light is green, it should look at it.

Influence SNePS and its immediate predecessors (*see* (Shapiro & Rapaport 1992)) have been influential in the fields of artificial intelligence, knowlege bases, and deductive databases. It was the first network-based knowledge representation system to clearly distinguish "system relations," represented by arcs from "conceptual relations" represented by nodes, the first network-based knowledge representation system to have a way of representing all of first-order logic, the first reasoning system to be able to reason with recursive rules without getting into an infinite loop, the first knowledge representation system to be fully and exclusively intensional, the first knowledge representation system to represent propositions about propositions without the need for quotation or modal operators, and the first reasoning system to include an assumption-based truth maintenance system. We believe that SNePS is the first system to integrate reasoning and acting in a syntactically and semantically clean way. In our current work on automatic belief revision, we are setting out to explicate how integrity constraints and belief revision postulates can be applied to and implemented in deductively open belief spaces.

Users and Useability

The potential users of SNePS should have a familiarity with logic at least at the level of students of an introductory logic course. A tutorial may be accessed from the SNePS Research Group web pages. SNePS has not been designed to apply to any specific area of application, so it is flexible enough to handle a wide variety of areas, but all the domain knowledge must be supplied by the user. In that sense, SNePS is like an expert system shell or logic programming language. However, it is different in having been designed to build cognitive agents, so the user is an informant rather than a programmer, and it is sometimes difficult even for the one user who has supplied all the domain knowledge to predict the conclusions SNePS will reach. Nevertheless, SNePS has been, and is being used by people outside the SNePS Research Group.

Evaluating the System

Benchmarks

SNePS comes with a suite of demonstration problems and applications that can be used to familiarize oneself with how to use it, and may be used for comparison with other systems. Demonstrations that were taken from other sources include The Jobs Puzzle from (Wos *et al.* 1984, Chapter 3.2), Schubert's steamroller problem (*see* (Stickel 1985)), and a database management system example from (Date 1981).

SNePS is moderately user friendly. It is "academic" software, rather than a commercial product, and this aspect could be somewhat improved.

Comparison

Schubert's steamroller problem was run on a 1993 version of SNePS (Choi 1993), and the results compared with those reported in (Stickel 1985). The SNePS version produced fewer unifications and was faster than most unsorted logic solutions, but was outperformed by sorted logic solutions. The current version of SNePS is much faster on this problem than the 1993 version, partially due to improvements in SNePS, and partially due to faster, bigger computers.

The SNePS representation of the Jobs Puzzle is much simpler and closer to the English version of the puzzle than the clause form representation presented in (Wos *et al.* 1984, p. 58ff).

Problem Size

Currently, SNePS can handle knowledge bases on the order of about 1,000 SNePS terms.

SNePS 2.5 is more than a prototype system. It is useable for research and experimentation. However, we would not claim it to be ready for full-scale applications. Before it would be ready for large-sized problems, it would need a concentrated effort of profiling and optimization.

References

Brachman, R. J., and Levesque, H. J., eds. 1985. *Readings in Knowledge Representation.* San Mateo, CA: Morgan Kaufmann.

Choi, J. 1993. *Experience-Based Learning in Deductive Reasoning Systems.* Technical report 93-20, Department of Computer Science, State University of New York at Buffalo, Buffalo, NY.

Date, C. J. 1981. *An Introduction to Database Systems*, volume 1. Reading, MA: Addison-Wesley, 3 edition.

Johnson, F. L., and Shapiro, S. C. 2000. Implementing integrity constraints in an existing belief revision system. In *Papers from the Belief Change Workshop at NMR2000.* in press.

Kumar, D., and Shapiro, S. C. 1994. Acting in service of inference (and *vice versa*). In Dankel II, D. D., ed., *Proceedings of The Seventh Florida AI Research Symposium (FLAIRS 94).* The Florida AI Research Society. 207–211.

Kumar, D. 1996. The SNePS BDI architecture. *Decision Support Systems* 16(1):3–19.

Lehmann, F., ed. 1992. *Semantic Networks in Artificial Intelligence.* Oxford: Pergamon Press.

Maida, A. S., and Shapiro, S. C. 1982. Intensional concepts in propositional semantic networks. *Cognitive Science* 6(4):291–330. Reprinted in (Brachman & Levesque 1985, pp. 170–189).

Martins, J. P., and Shapiro, S. C. 1988. A model for belief revision. *Artificial Intelligence* 35:25–79.

Shapiro, S. C., and Rapaport, W. J. 1985. SNePS considered as a fully intensional semantic network. Technical Report 85–15, Department of Computer Science, SUNY at Buffalo.

Shapiro, S. C., and Rapaport, W. J. 1992. The SNePS family. *Computers & Mathematics with Applications* 23(2–5):243–275. Reprinted in (Lehmann 1992, pp. 243–275).

Shapiro, S. C., and The SNePS Implementation Group. 1999. *SNePS 2.5 User's Manual.* Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY.

Shapiro, S. C.; McKay, D. P.; Martins, J.; and Morgado, E. 1981. SNePSLOG: A "higher order" logic programming language. SNeRG Technical Note 8, Department of Computer Science, State University of New York at Buffalo. Presented at the Workshop on Logic Programming for Intelligent Systems, R.M.S. Queen Mary, Long Beach, CA, 1981.

Shapiro, S. C. 1982. Generalized augmented transition network grammars for generation from semantic networks. *The American Journal of Computational Linguistics* 8(1):12–25.

Shapiro, S. C. 1992. Relevance logic in computer science. In Anderson, A. R.; Belnap, Jr., N. D.; and

Dunn, M., eds., *Entailment*, volume II. Princeton: Princeton University Press. 553–563.

Shapiro, S. C. 1993. Belief spaces as sets of propositions. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)* 5(2&3):225–235.

Shapiro, S. C. 2000. SNePS: A logic for natural language understanding and commonsense reasoning. In Iwańska, Ł., and Shapiro, S. C., eds., *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language.* Menlo Park, CA: AAAI Press/The MIT Press.

Stickel, M. E. 1985. Automated deduction by theory resolution. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI'85)*, volume 2, 1181–1186. Los Altos, CA: Morgan Kaufmann.

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