**Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH**

**Technical
Memo**

TM-96-02

Knowledge Bases in the World Wide Web: A Challenge for Logic Programming

Harold Boley

Second, Revised Edition

October 1997

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
67608 Kaiserslautern, FRG
Tel.: + 49 (631) 205-3211
Fax: + 49 (631) 205-3210

Stuhlsatzenhausweg 3
66123 Saarbrücken, FRG
Tel.: + 49 (681) 302-5252
Fax: + 49 (681) 302-5341

Knowledge Bases in the World Wide Web: A Challenge for Logic Programming

Harold Boley

DFKI-TM-96-02

Earlier versions appeared in:
Proc. JICSLP'96 Post-Conference Workshop on Logic Programming Tools for INTERNET Applications, Bonn, September, 1996; Proc. KI-97 Workshop on Intelligent Information Integration, Freiburg, September, 1997; Proc. 12. WORKSHOP LOGISCHE PROGRAMMIERUNG – WLP'97, Munich, September, 1997 (German version)

This work has been supported by a grant from The Federal Ministry of Education, Science, Research and Technology (FKZ 413-5839-ITW-9304/3).

ISSN 0946-0071

Knowledge Bases in the World Wide Web: A Challenge for Logic Programming

Harold Boley
DFKI GmbH, Kaiserslautern, Germany
boley@informatik.uni-kl.de

October 25, 1997

Abstract

Regarding the World Wide Web, knowledge bases can be categorized between (HTML-)documents and (SQL-)databases. In order to standardize them, the use of Horn logic for Web publications is proposed. The central part outlines the design of a Web search engine for processing distributed Horn-logic knowledge bases. Some of the research issues to be solved are elaborated from the perspective of (parallel, modular) logic programming. A proposal for realization is based on the Infomaster system, here restricted from heterogeneous notations to a standardized format. Possible Inter- and intranet applications are discussed. Finally, an LP-community effort for building up (documented) Horn-logic knowledge bases in the Web is encouraged. [1]

1 Web-Knowledge Reuse via Horn Logic

The World Wide Web has quickly established itself as the standard accepted medium for information supply, for example for (hyper-)texts, images and databases. But what about the supply of standardized **knowledge** bases in the WWW, which could constitute a leap toward knowledge reuse? After a promising initialization of the WWW publication of *KIF/Ontolingua knowledge bases* (e.g. [8]) by Thomas Gruber right at the beginning of the expansion of the Web, this effort slowed down since he left the Knowledge Systems Laboratory

[1] Thanks go to the following people: Michael Sintek helped a lot with both form and content of this paper; remarks by Andreas Abecker, Ansgar Bernardi, Thomas Kieninger, Otto Kühn, Norbert Fuchs, and others contributed to further improvements; the reviewers of the 1st Workshop on Logic Programming Tools for INTERNET Applications motivated section 6; later, section 4 was added; two reviewers of the KI-97 Workshop on Intelligent Information Integration inspired the 'Horn logic as common subset' discussion; final improvements came about – in a special way – through Ulrich Geske and through two reviewers of the 12. WORKSHOP LOGISCHE PROGRAMMIERUNG – WLP'97; Jane Bensch helped with alternating German and English drafts.

at Stanford University (but see section 4). Also the growth rate of WWW knowledge which is represented (e.g. CGs, CycL, CommonKADS) or accessed (e.g. KQML, April) in other proposed AI formats is not keeping track with the general Web expansion rate. This may be due to the complex structure and insufficient support of these formats – in comparison, e.g., to the use of the *SQL-database standard* for HTTP/SQL-gateways: Despite the distribution of ANSI proposals for KIF and CGs it seems that even in these cases there is not enough 'investment security' for knowledge publications.

Could, in this situation, a (HTML-)version of Horn logic, the established basis in logic programming, play the role of a declarative WWW representation standard?[2] First of all, the relational databases that are already available in the Web could be principally viewed as Datalog special cases of such Horn-logic knowledge bases, normally restricted to ground facts and non-recursive rules – apart from problems in connection with *null values* (database schemas will be discussed later on). Moreover the accumulated program libraries in pure Prolog (from Edinburgh to ISO) could be easily revised for Web publication. Also, special knowledge bases in modern logic languages such as λProlog or Gödel, or in *functional-logic languages* [3] such as Escher, Curry or Relfun could be more or less easily reproduced in Horn logic. Finally, further translators between subsets of the above-mentioned exchange formats and Horn logic could be tackled: e.g. (very) partial KIF-Prolog translators have been written already; the Horn-logic reduction of sort lattices will be mentioned below. A desirable side effect would be that translation attempts between concrete knowledge bases (via Horn logic) would increase the comparability of new LP proposals. If demanded by knowledge-representation or WWW-presentation requirements, the Horn-logic kernel language, along with its querying and browsing mechanisms, could gradually be extended: logically, for e.g. negation, disjunction, explicit quantifiers, and/or implicational goals; functionally, for, say, returned values, nestings, and/or higher-order notation; on a level corresponding to Java Applets, perhaps even non-declarative (OOP) extensions, e.g. for graphics/animation, would become necessary. Still, such content pages in the Web could take advantage of a basically rule-oriented formalization, with Horn logic as the common subset, instead of relying only on ambiguous (and multilingual) natural language(s). With the addition of *integrity constraints* [4], again transferred to WWW pages, Horn-logic knowledge bases can also be easily maintained interactively [1]. From the perspective of the user of relational databases this would be a transition to a standard for *deductive databases* [4] as far as expressive power is concerned.

[2]The following preliminaries address readers who know more about the Web than about Horn logic. Using it, rule knowledge can be represented in a very restricted natural-language top-level syntax of the form C `if` P_1 `and` ... `and` P_n, where, however, the conclusion C and the premises P_i have the form of *atoms*, i.e. they are parenthesized predicate applications to constants, variables, or structures [10]. Printed Horn-logic knowledge uses quite varied syntaxes: Formal tradition and compactness for example support a left arrow as the `if` in rules, ASCII-usability, in contrast, supports Prolog's clause notation; on the other hand mathematics and logic would go against Prolog's capitalized variables. Structures could be written with list-like square brackets to distinguish them from atoms, which would also make functional-logic integration easier [3]. Due to the Web a uniform syntax is becoming more pressing.

Altogether, because of their simplicity-expressiveness tradeoff constituting a local optimum in the intersection of several formalisms, the number of available Horn-logic WWW knowledge bases can most likely reach the critical mass that causes automatically processed queries provide interesting results to end-users. An example for possible applications will be presented in section 5.

2 Prolog Goals for Web-Distributed Clauses

Another important point arises from the Web distribution of knowledge bases. The characteristic *fact-retrieval* use of databases in the Web considers an SQL database as a black box on a given server, which can be queried via a form. In contrast to this is the *document-retrieval* use of (hyper)texts via search engines, which find appropriate Web pages arbitrarily distributed on servers using keyword combinations. The proposed Horn-logic knowledge bases in the WWW constitute an information category between SQL databases and HTML documents. The following question arises: Is it possible, on the one hand, to distribute this knowledge using the paradigm of HTML documents and to access it, on the other hand, using the paradigm of SQL-form queries or Prolog goals? In practice this would mean: Web-distributed knowledge bases should not be queried by the (primitive, since string-oriented) keywords of usual search engines; instead Prolog-like **goals** should be used, perhaps via a fancy HTML form-interface, that is atoms, which may contain logical (i.e. consistently bound) variables, which, in the Prolog way, may be combined conjunctively (with a comma) or disjunctively (with a semicolon), and which perhaps – on this query top-level – could also be negated in a way suitable for subwebs (global *negation as failure* and SLDNF-resolution appear to make little sense in the 'open Web world'; see discussion below). Apart from the increased expressive power, compared to SQL, through variables and structures in clauses (facts **and** rules), such knowledge bases inherit an advantage from HTML documents – the possibility of seeing a found answer to the current query (atom or string) in the context of neighboring potential answers (atoms or strings), i.e. to associatively browse (in knowledge bases or documents) instead of only communicating with a deductive black box:[3] As an optional 'explanation' of answer bindings it would be possible to highlight both the atomic facts to which a goal is reduced and the rules used to derive it, either within the knowledge bases or – more focused – the predicate definitions containing them.

While document-retrieval engines deliver, for each keyword-containing document, its summary (normally just consisting of the first few document lines) and link,

[3] Other non-deductive uses of such knowledge bases are also useful, e.g. through machine-learning algorithms on the basis of inductive logic programming [1]. In particular, once knowledge is (Horn-)formalized in the Web, the search engine sketched below could – during its periodic indexing of the distributed knowledge pages – also propose new connections between knowledge items in geographically disparate pages, perhaps bringing their authors into contact with each other; a simple example of a connection would be the use of an identical predicate name, as discussed below.

the proposed knowledge-inferring engine would offer, for each subgoal-resolving knowledge base, its summary (e.g. the used clauses) and link, along with the answer bindings of successful refutations. Thus, instead of keyword indexing, some suitable clause-head indexing will be needed. Deviating farther from document retrieval, a clause body (the rule premises) can be viewed as a sequence of abstract (local or global) knowledge-base links, which have to be followed by the knowledge-inferring engine in further resolution steps. Even for unsuccessful refutation attempts the knowledge-base links found can be valuable because submitting refined queries to one of the newly discovered knowledge bases' possible own inference server, or to its downloaded version, or just browsing through it may lead to problem reformulations and unexpected solutions.

Like for the goal inputs, a fancy HTML interface could be used for the resolved-clause and answer-binding outputs. For instance, the found ASCII-HTML representation of a commented predicate definition or knowledge base of clauses may be rendered by the browser as a cross-referenced, indexed pretty-print presentation – in the tradition of Donald E. Knuth's *literate programming*.

Such use of Web-distributed knowledge could theoretically be implemented using keyword searches (on a lower level also atoms are strings). In practice the realization is an interesting challenge for logic programming.[4]

3 Design Issues for a Horn Search Engine

Below follows a first sketch of a possible search engine for knowledge bases (KBs) Horn-published in the Web, henceforth called a **KB-search engine**.

This would be useful even if it would just take better into account the formal (Horn-logic) syntax of knowledge-base entries than present text-search engines do. But here initial semantic and inferential abilities will be added.

First, **typing** of knowledge bases is helpful, as by providing (`HORN-`)`KB` labels for or in their HTML pages, e.g. already in the *header*, to consider only relevant, i.e. KB pages, while doing a search (the other way round, 'text-only'-search engines could thus skip KB items). A variant, at least for Datalog, is the *tag* `table` of HTML 3.2, with which predicate definitions can be presented as tables (but a transition to the more general SGML might be advantageous also for this Web use, exploiting *generic identifiers* for KBs, predicates, clauses, and perhaps below). In any case it is useful, analogously to an SQL-database schema, to declare the signatures of n-ary predicates in order to establish the

[4] Unlike in the more often studied administration of arbitrary Web pages **through** (extra-)logic programs (e.g. [6], [11]), Web extensions **for** (Horn-)logic programs with arbitrary implementation languages are considered here. Both LP-WWW connections could in principle be examined simultaneously.

meaning of the n arguments. The *sorts* usable here can themselves be defined as unary predicates in sort lattices (e.g. via single-premise rules or binary `subsumes`-relationships between the predicates). Now, the KB-search engine, before using a found predicate definition, first tests its signature compatability wrt the perhaps itself sorted goal atom. This optional use of order-sortedness for **semantic prefiltering** avoids elementary predicate ambiguities, such as those known from the corresponding tiresome keyword ambiguities of todays text-search engines. In this way it is in principle possible to attain a higher *precision* using formal KBs than using natural-language texts. Modeled on the joint FAQ elaboration for newsgroups, expert groups would be able, again with the help of the Internet, to agree on their special vocabulary and their sort sublattices, with interfaces to related knowledge areas. Gradually it would thus be possible to reduce the ambiguity of predicate names.

The **inferential kernel problem** of a KB-search engine in the Web corresponds to the problem of `OR`-*parallel architectures* such as Aurora [13], whose resources have to be balanced between local search (normally sequential) and global search (parallel, communication-intensive). The KB-search engine has to suitably interleave resolution on a (found) KB page with the Web supply of (further) KB pages. Here is also a connection to the theme of *modular/contextual LP* [5]: When should a goal be proven locally in the current KB page (module), when should be globally proceeded to other KB pages (modules)? When should backtracking cross page (module-)boundaries? While a page-local sequence of resolution steps is useful mainly for recursive rules, the global search process has to continuously spread in parallel over the KB pages (e.g. with algorithms such as *fish-search* [14]). Local non-terminations then cannot endanger the global success of a query: Even if it could be presupposed that all knowledge bases published in the Web would be terminating individually, there could still be non-terminations caused by the interaction of distributed knowledge bases, e.g. via two distributed co-recursive rules. Since it is difficult to capture the (model theoretic) semantics of a predicate definition even with signature declarations if its clauses are distributed over several knowledge bases, it will be beneficial to agree on more or less strong restrictions here. The strongest **locality principle** would completely forbid predicate distribution, a middle one would only allow the distribution of facts, and the weakest would also tolerate distributed non-recursive rules. Furthermore the important question arises whether, for uniformity of the semantics, also the clauses **within** a KB page should be interpreted according to pure Horn logic in an `OR`-parallel (breadth-oriented) manner instead of the Prolog-like use of backtracking. In any case one would rather expect solutions to surface Lycos-type, in groups, than Prolog-type, by individual `more` requests; their order could also reflect certainty factors, starting from the facts used. Due to the openess and dynamics of the WWW, the issue of logical 'completeness' of solution sets would hardly arise: The KB-search engine even has to cut off the search processes which are still running (possibly non-terminating), if no (further) solution group is requested; an SQL-type (total-)set orientation would thus not be sensible here.

Finally, it is necessary to examine how the **indexing techniques** of text-search engines, databases, and Prolog can be transferred to the KB-search engine. Within a KB page, the signature declarations preceding the predicate definitions ('procedures') can play the role of indexed HTML headings. A finer KB indexing down to the level of single clauses would be conceivable analoguously to a complete text indexing, with advantages and disadvantages similar to those of Alta Vista.

The KB-search engine outlined above mainly works *interpretively* on KB sources, which also can be used with normal browsers. This permits, apart from black-box queries, to also directly inspect and configure knowledge bases much like text building blocks. Only by offering this direct manipulation of (Horn-logic) standardized knowledge do knowledge bases have a certain chance to become common property in the way of EXCEL tables: For this not only conducting knowledge-base **queries** (spread-sheet **calculations**) is important, but also helping with the **construction**, **change**, **combination** and **bidirectional text coupling** of knowledge bases (spread sheets). This could be supported by the hypertext mechanisms of the Web. For instance, the submodules of a knowledge base would have to be accessible through the KB-search engine as well as interactively via a special HTML *link*. If, due to efficiency considerations, the *compilation* of a Horn-logic knowledge base becomes necessary, it could be tried to simulate direct manipulation, e.g. via back-references to the source and automatic recompiliation after changes. In contrast to an even more efficient native-code compilation, a WAM compilation would preserve the advantage of pseudo-code transmission and client-side emulator execution first used by Java.

4 A Standardized Infomaster Adaptation

One possibility for the implementation of the KB-search engine would consist of an adaptation of the *Infomaster*^{TM} technology developed by Michael Genesereth and the Logic Group at Stanford University [7]. Infomaster is a *virtual information system* for accessing distributed heterogeneous databases and knowledge bases, implemented in the Agents Communication Language ACL, a combination of KQML, KIF, and ontologies. For this, *facilitators* are used, which – in our formulation – reduce ACL(-converted) goals (e.g. to conjoined subgoals), deliver them to fact bases and other facilitators, and translate various representations into each other.

On the one hand, the KB-search engine can be seen as a specialization of Infomaster in the sense that, due to the approached Horn-logic KB standard, the use of the translation function of facilitators is only limited (see below). On the other hand, the aim of Infomaster, expanding its use for the Stanford Information Network, is to become the basic technology for a World Information Network (WIN), which apparently shall be built in competition to the WWW; here, instead, the existing general Web infrastructure of networked HTML pages

shall be used directly for knowledge bases.

For an Infomaster implementation of the KB-search engine the need thus arises to translate between the chosen Horn-logic HTML syntax and ACL. There has already been a lot of preliminary work done to this end, e.g. with the translators, developed by our group, between pure Prolog and a subset of KIF (via Relfun's Prolog- and Lisp-like syntaxes). These translators were provided in the context of a Stanford-Kaiserslautern collaboration on KIF and facilitators. As they – like the Infomaster software – are programmed in Lisp, an integration seems simple.

From the Infomaster point of view, distributed Horn-logic knowledge bases constitute a standardized simplification of the notational heterogeneity allowed there. Conversely, we avoid the following additional problems of this heterogeneity:
a) A direct manipulation of found sub-KBs on the model of text building blocks and EXCEL tables is made more difficult.
b) The translation of found sub-KBs to other representations need not be uniquely reversible, which leads to back-translation problems for error handling and maintenance.
c) Dynamic translations create worse run-time behavior than a static knowledge 'standardization' done once.
d) Each knowledge base, which is (further) maintained in a single standardized format, spares the additional maintenance of a pair of translators.

Of course, tackling with heterogeneous **notations** achieves a higher generality. However, as discussed earlier, even within the standardized Horn-logic notation it will be difficult to reduce the heterogeneity of **vocabularies** (e.g., sorts and signatures) to be expected in a world-wide distributed setting. The realization of the KB-search engine via an adapted Infomaster technology allows a more comprehensive examination of these tradeoffs of standardization and heterogeneity.

5 Internet Experience and Intranet Prospects

Due to the comparatively simple formalizabilty of technical knowledge, it is sensible to first construct Horn-logic knowledge bases for this area. In the VEGA project we have done this mainly for environmental subareas of materials science. However, knowledge bases such as VEGA's `RTPlast` on recyclable thermoplastics and our recently started `Nutrimine` on micro nutrients are published mainly as pure ASCII files (in Relfun syntax) in the Web (cf. [2]). It is planned to include more Horn-logic knowledge bases, e.g. to compare functional-logic programs. But it is still open which operational variant of Horn logic is the most suitable for the publication of practical knowledge bases in the Web: Experience would encourage to exploit the textual order of clauses (`OR` sequentiality) and their premises (`AND` sequentiality), while the distributed Web search suggests an at least `OR`-parallel interpretation. After the decisions necessary here, mainly a

more exact specification of the HTML additions for Horn-logic knowledge bases and a revision of the search engine operating on them has to be completed before a prototype implementation (e.g. in Java) can be begun. For reasons of acceptance, the efficiency of the KB-search engine plays a major role of course; it could be increased both by progress in LP (e.g. better `OR`-parallel strategies) and in WWW technology (e.g. better efficiency of Java).

Interesting possibilities for WWW knowledge bases are also offered, in the form of *intranets*, by **organization-internal** information supply on the basis of Web technology. In analogy to a central *corporate memory* [9] the intranet knowledge bases of a networked organization (via LAN or WAN) would become the knowledge component of a *distributed corporate memory*. The information deficits in large organizations, which cost time and money, show that even for intranet knowledge bases it would be useful to have a KB-search engine, which, because of the smaller knowledge volumes, would be able to work faster here than in the open Internet. If organizations use the externally agreed upon (Horn-logic) standard also as their internal representation format, then they can move knowledge bases back and forth between their intranet and the public Internet as needed: Inward to take over or adapt new know-how; outward for (subset or previous-version) demos and perhaps (cryptographical, Internet-payment) sales of their expertise. Analogously, a unified representation will permit flexible alliances between knowledge-intensive foundations or firms.

6 Web knowledge bases: A Community Effort

Besides the technical issues discussed in the body of this paper, the building up of knowledge bases in the World Wide Web has a strong 'community' component: While interest in providing and using Inter- or intranet KBs is growing, we need to converge on a standard representation format. Since Horn logic already is the language of choice for many non-Web KB publications and at the core of many LP extensions, it looks like the right place to start with. If Web-KB providers use Prolog syntax, their KBs will be translatable into the ultimate format indexed and retrieved by the envisaged Horn-logic KB-search engine, since there already exists a sufficient number of 'legacy' Prolog KBs also requiring such a translator. When attempting to build reusable Web Horn KBs, the harder part will probably be common vocabulary or 'ontology' decisions concerning compatible definitions of predicate names. However, the sorts needed for hierarchically structuring non-trivial domains can, for uniformity, be regarded as unary predicates, again defined by (simple) Horn rules. Once a search engine hits enough clauses on typical goals, a self-accelerating effect in the growth of Web KBs may start, similar to the one we are witnessing for other information categories in the Web. This can be further encouraged by coupling (essential) Horn clauses with natural-language texts acting like an inline documentation: when such texts will be found by ordinary search engines, Web users get acquainted to the more formal clause representation of knowledge.

This paper has attempted to show that the LP community is principally well-prepared to venture on such a major Web effort. Among the approaches to create a Web infrastructure for reusing logic programs or KBs, the present proposal comes closest to the one independently developed in [12]: "Lightweight deductive databases" are deductive databases incorporated into Web pages to provide a source of distributed structured information. Their principal features, *distributed maintenance*, *extensibility*, and *reusability*, as well as notions from modular LP are common with our proposal. However, like other search engines, we stick to *server-side processing* as the default. Additionally, we stress *purity* in the sense that no extra operators, e.g. for invoking goals in other modules, are used. Instead, a pure Horn-logic KB should be loadable unchanged into a `KB`-tagged Web page, where the KB-search engine deals with the modular structure of pages much like textual search engines do. Purity also supports `OR`-*parallelism* as a paradigm to exploit Web-distributed KBs, implementable by 'agent replication' – as, e.g., in Aurora [13]. Finally, besides deducing answer bindings, we feel it is important that a KB-search engine can also 'explain' solutions by highlighting clauses used for a deduction in the context of their KB pages.

Altogether we do not propose a new LP language for the Web but the pragmatic reuse of Horn-logic KBs – in a Prolog-like syntax – by a new search engine. Once some experience has been accumulated with Web-searching such KBs, Web-oriented – but still pure – extensions of the Horn-logic kernel language could be carefully introduced. But before going into the nitty-gritty details – and quarrels – of defining a 'Web LP' language, we should experiment with one or more search engines for a fixed (Horn-)language.

7 Bibliography

[1] A. Abecker, H. Boley, K. Hinkelmann, H. Wache, and F. Schmalhofer. An Environment for Exploring and Validating Declarative Knowledge. In: *Proc. Workshop on Logic Programming Environments at ILPS'95*, Portland, Oregon, Dec. 1995.

[2] H. Boley, U. Buhrmann, and Chr. Kremer. Towards a Sharable Knowledge Base on Recyclable Plastics. In: J. K. McDowell and K. J. Meltsner, Eds., *Knowledge-Based Applications in Materials Science and Engineering*, pp. 29-42, TMS, 1994.
`http://www.dfki.uni-kl.de/~vega/relfun.html`
(Examples: RTPlast, for a related effort see Nutrimine)

[3] H. Boley. Extended Logic-plus-Functional Programming. In: L.-H. Eriksson, L. Hallnäs, and P. Schroeder-Heister, Eds., *Proc. of the 2nd International Workshop on Extensions of Logic Programming – ELP '91*, Springer LNAI 596, 1992.

[4] F. Bry, R. Manthey, and H. Schütz. Deduktive Datenbanken. In: N. E. Fuchs, Ed., *KI – Themenheft Logische Programmierung*, 3/96, ScienTec Publishing GmbH, Sept. 1996.

[5] M. Bugliesi, E. Lamma, and P. Mello. Modularity in Logic Programming. *Journal of Logic Programming*, 19/20, pp. 443-502, 1994.

[6] D. Cabeza and M. Hermenegildo. `html.pl`: *A Simple HTML Package for Prolog and CLP Systems – Description and User's Manual (Version 96.1.1)*. Computer Science Department, Technical University of Madrid (UPM), March 1996.

[7] D. F. Geddis, M. R. Genesereth, A. M. Keller, and N. P. Singh. Infomaster: A Virtual Information System. *Intelligent Information Agents Workshop, Fourth International Conference on Information and Knowledge Management*, Baltimore, Maryland, Dec. 1995.
`http://infomaster.stanford.edu/` (Documentation: User documentation)

[8] Th. R. Gruber and G. R. Olsen. An Ontology for Engineering Mathematics. In: J. Doyle, P. Torasso, and E. Sandewall, Eds., *Fourth International Conference on Principles of Knowledge Representation and Reasoning*, Gustav Stresemann Institut, Bonn, Germany, Morgan Kaufmann, 1994.
`http://www-ksl.stanford.edu/knowledge-sharing/ontologies/html/index.html`
(Reference Documents: The Engineering Math Ontologies)

[9] K. Hinkelmann and O. Kühn. Revising and Updating a Corporate Memory. In: *Proc. of the European Symposium on Validation and Verification of Knowledge-based Systems – EUROVAV-95*, June 1995.

[10] R. A. Kowalski. Logic Programming in Artificial Intelligence. *12th International Joint Conference on Artificial Intelligence – IJCAI'91*, pp. 596-603, 1991.

[11] S. W. Loke and A. Davison. Logic Programming with the World-Wide Web. In: *Proc. of the 7th ACM Conference on Hypertext – Hypertext'96*, ACM Press, 1996.

[12] S. W. Loke, A. Davison, and L. Sterling. Lightweight Deductive Databases on the World-Wide Web. In: P. Tarau, A. Davison, K. De Bosschere, and M. Hermenegildo, Eds., *Proc. of the 1st Workshop on Logic Programming Tools for INTERNET Applications*, JICSLP'96, Bonn, Sept. 1996.

[13] E. Lusk, D. H. D. Warren, S. Haridi, et al. The Aurora Or-Parallel Prolog System. In: *International Conference on Fifth Generation Computer Systems 1988*, ICOT, Tokyo, Japan, Nov. 1988.

[14] R. D. J. Post and P. M. E. De Bra. Information Retrieval in the World Wide Web: Making Client-based Searching Feasible. In: *Proc. of the First International WWW Conference*, Geneva, May 1994.

Knowledge Bases in the World Wide Web: A Challenge for Logic Programming

Harold Boley

TM-96-02
Technical Mem