Wissensbasen im World Wide Web: Eine Herausforderung für die Logische Programmierung

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

Im World Wide Web lassen sich Wissensbasen zwischen (HTML-)Dokumenten und (SQL-)Datenbanken einordnen. Zu ihrer Standardisierung wird eine Verwendung der Hornlogik für Web-Publikationen vorgeschlagen. Der Hauptteil legt den Entwurf einer Web-Suchmaschine zur Auswertung verteilter hornlogischer Wissensbasen dar. Einige der zu lösenden Forschungsfragen werden aus der Perspektive der (parallelen, modularen) logischen Programmierung herausgearbeitet. Ein Realisierungsvorschlag basiert auf dem Infomaster-System, hier eingeschränkt von heterogenen Notationen auf ein standardisiertes Format. Mögliche Inter- und Intranet-Anwendungen werden diskutiert. Schließlich wird eine LP-Gemeinschaftsanstrengung zum Aufbau von (dokumentierten) hornlogischen Wissensbasen im Web angeregt.[1]

1. Web-Wissenswiederverwendung über die Hornlogik

Das World Wide Web hat sich in kurzer Zeit als Standard-Medium hoher Akzeptanz für die Informationsbereitstellung etabliert, z.B. für (Hyper-)Texte, Bilder und Datenbanken. Wie steht es jedoch mit der Bereitstellung von standardisierten **Wissens**basen im WWW, die einen Sprung in Richtung auf Wissenswiederverwendung darstellen könnte? Nach einer vielversprechenden Initialisierung der WWW-Publikation von *KIF/Ontolingua-Wissensbasen* (z.B. [8]) durch Thomas Gruber, gleich zu Beginn der Web-Ausbreitungswelle, verlangsamte sich diese Anstrengung seit seinem Weggang vom Knowledge Systems Laboratory der Stanford University (vgl. aber Abschnitt 4). Auch die Wachstumsrate des WWW-Wissens, das in anderen vorgeschlagenen KI-Formaten repräsentiert (z.B. CGs, CycL, CommonKADS) bzw. zugegriffen (z.B. KQML, April) wird, hält nicht mit der allgemeinen Web-Expansionsrate Schritt. Dabei spielt wohl die komplexe Struktur und der mangelnde Support dieser Formate - im Vergleich etwa zur Nutzung des *SQL-Datenbank-Standards* für HTTP/SQL-Gateways - eine Rolle: Trotz der Verbreitung von ANSI-Vorschlägen für KIF und CGs scheint selbst bei diesen keine ausreichende 'Investitionssicherheit' für Wissenspublikationen zu bestehen.

Könnte in dieser Situation eine (HTML-)Version der *Hornlogik*, also der etablierten Basis in der Logischen Programmierung, die Rolle eines deklarativen WWW-Repräsentationsstandards übernehmen?[2] Zunächst wären die bereits im Web verfügbaren relationalen Datenbanken im

---

[1] Dank sagen möchte ich wie folgt: Michael Sintek half viel bei Form und Inhalt dieser Arbeit; Bemerkungen von Andreas Abecker, Ansgar Bernardi, Thomas Kieninger, Otto Kühn, Norbert Fuchs und anderen trugen zu weiteren Verbesserungen bei; die Gutachter des *1st Workshop on Logic Programming Tools for INTERNET Applications* motivierten Abschnitt 6; später kam Abschnitt 4 hinzu; zwei Gutachter des *KI-97 Workshop on Intelligent Information Integration* inspirierten die Diskussion der 'Hornlogik als gemeinsamer Teilmenge'; abschließende Verbesserungen kamen - auf besondere Weise - durch Ulrich Geske und durch zwei Gutachter des *12. WORKSHOP LOGISCHE PROGRAMMIERUNG - WLP'97* zustande; Jane Bensch half beim Hin und Her zwischen deutschen und englischen Vorversionen.

[2] Für Leser, die mehr über das Web als über die Hornlogik wissen, sei folgendes vorausgeschickt. Regelwissen kann damit in einer sehr eingeschränkten natürlichsprachlichen Top-level-Syntax der Form *C* `if` P_1 `and` ... `and` P_n repräsentiert werden, wobei allerdings die Folgerung *C* und die Prämissen P_i die Form von *Atomen* haben, d.h. geklammerte Prädikatanwendungen auf Konstanten, Variablen bzw. Strukturen sind [10]. Gedrucktes

wesentlichen als Datalog-Spezialfälle solcher Hornlogik-Wissensbasen, meist eingeschränkt auf Grundfakten und nichtrekursive Regeln, betrachtbar - abgesehen von Problemen im Zusammenhang mit *null values* (Datenbankschemata werden weiter unten diskutiert). Außerdem könnten die in purem Prolog (von Edinburgh bis ISO) akkumulierten Programmbibliotheken leicht für die Web-Publikation aufbereitet werden. Auch spezielle Wissensbasen in modernen logischen Sprachen wie λProlog oder Gödel bzw. in *funktional-logischen Sprachen* [3] wie Escher, Curry oder Relfun ließen sich oft mehr oder weniger einfach auf die Hornlogik abbilden. Schließlich könnten weitere Translatoren zwischen Teilmengen einiger der oben angesprochenen Austauschformate und der Hornlogik in Angriff genommen werden: Z.B. wurden bereits (sehr) partielle KIF-Prolog-Translatoren erstellt; unten wird die Hornlogik-Reduktion von Sortenverbänden angesprochen. Ein wünschenswerter Nebeneffekt dabei wäre, daß Übersetzungsansätze zwischen konkreten Wissensbasen (über die Hornlogik) die Vergleichbarkeit neuerer LP-Vorschläge erhöhen würden. Wenn es Anforderungen der Wissensrepräsentation oder WWW-Präsentation gebieten, könnte die Hornlogik-Kernsprache, zusammen mit ihren Anfrage- und Browsing-Mechanismen, allmählich erweitert werden: logisch, für z.B. Negation, Disjunktion, explizite Quantoren, und/oder Implikations-Ziele; funktional, für etwa Rückgabewerte, Schachtelungen, und/oder eine Notation höherer Ordnung; auf einer Ebene die der von Java Applets entspricht, könnten sogar nichtdeklarative (OOP-)Erweiterungen, z.B. für Graphik/Animation, notwendig werden. Dennoch könnten solche Inhaltsseiten im Web Vorteile aus der grundsätzlich regelorientierten Formalisierung ziehen, mit der Hornlogik als gemeinsamer Teilmenge, statt sich nur auf mehrdeutige (und multilinguale) natürliche Sprache(n) zu verlassen. Unter Hinzunahme von *Integritätsbedingungen* [4], wiederum transferiert auf WWW-Seiten, können Hornlogik-Wissensbasen auch gut interaktiv gewartet werden [1]. Dies wäre aus der Perspektive der Nutzer relationaler Datenbanken bezüglich der Ausdrucksmächtigkeit ein Übergang zu einem Standard für *deduktive Datenbanken* [4]. Insgesamt kann die Anzahl der in Hornlogik verfügbaren WWW-Wissensbasen, wegen ihres Einfachheits-Ausdrucksstärke-Tradeoffs, der ein lokales Optimum im Schnittbereich mehrerer Formalismen darstellt, am ehesten die kritische Masse erreichen, von der ab automatisch ausgewertete Anfragen für Endnutzer interessante Resultate erzielen. Ein Beispiel für mögliche Anwendungen wird in Abschnitt 5 vorgestellt.

2. Prolog-Ziele für Web-verteilte Klauseln

Durch die Web-Verteilung kommt ein weiterer wichtiger Gesichtspunkt hinzu. Die charakteristische *fact retrieval*-Verwendung von Datenbanken im Web betrachtet eine SQL-Datenbank als Black Box auf einem gegebenen Server, die über ein Formular abfragbar ist. Im Gegensatz dazu steht die *document retrieval*-Verwendung von (Hyper-)Texten durch Suchmaschinen, die über Schlüsselwort-Kombinationen passende Web-Seiten auf beliebig verteilten Servern finden. Die vorgeschlagenen Hornlogik-Wissensbasen im WWW konstituieren eine zwischen SQL-Datenbanken und HTML-Dokumenten stehende Informationskategorie. Es stellt sich folgende Frage: Besteht die Möglichkeit, dieses Wissen einerseits nach dem Vorbild von HTML-Dokumenten zu verteilen und es andererseits nach dem Vorbild von SQL-Formularabfragen oder von Prolog-Zielen anzufragen? Konkret hieße dies also: Web-verteilte Wissensbasen sollen nicht mit den (primitiven, weil zeichenkettenorientierten) Schlüsselwörtern üblicher Suchmaschinen angefragt werden; stattdessen sollen, evtl. über eine ansprechende Formular-Schnittstelle, Prolog-artige **Ziele** verwendet werden, d.h. Atome, die logische (also konsistent zu bindende) Variablen enthalten dürfen, die auf Prolog-bekannte Weise konjunktiv (mit Komma) oder disjunktiv (mit Semikolon) verknüpft sein können und - auf diesem Anfrage-Toplevel - evtl. auch Subweb-geeignet negiert schreibbar wären (eine globale *negation as failure* und SLDNF-Resolution würde in der 'offenen Web-Welt' wohl wenig Sinn machen; vgl. Diskussion unten). Außer der gegenüber SQL gesteigerten Ausdrucksmächtigkeit durch Variablen und Strukturen in Klauseln (Fakten **und** Regeln), erben solche Wissensbasen einen Vorteil von HTML-Dokumenten - die

---

Hornlogik-Wissen benutzt recht uneinheitliche Syntaxen: Formale Tradition und Kompaktheit sprechen z.B. für einen Linkspfeil als `if` in Regeln, die ASCII-Verwendbarkeit dagegen für Prologs Klausel-Notation; andererseits sprächen Mathematik und Logik gegen Prologs Großschreibung von Variablen. Strukturen könnten zur Unterscheidung von Atomen mit Listen-artigen eckigen Klammern notiert werden, was auch die funktional-logische Integration erleichtern würde [3]. Durch das Web wird eine einheitliche Syntax dringlicher.

Möglichkeit, die gefundene aktuelle Antwort auf eine Anfrage (Atom bzw. Zeichenkette) im Kontext benachbarter potentieller Antworten (Atome bzw. Zeichenketten) zu sehen, d.h. (in Wissensbasen bzw. Dokumenten) zu *browsen* statt nur mit einer Black Box zu kommunizieren:[3] Als eine optionale 'Erklärung' von Antwortbindungen könnten neben den zur Abstützung eines Ziels benutzten atomaren Fakten auch die zu seiner Herleitung verwendeten Regeln innerhalb der sie enthaltenden Wissensbasen oder - fokussierter - Prädikat-Definitionen hervorgehoben werden.

Während *document retrieval*-Maschinen für jedes Schlüsselwort-enthaltende Dokument eine Kurzfassung (normalerweise einfach die ersten wenigen Dokument-Zeilen) und einen *link* liefern, soll die vorgeschlagene Wissensfolgerungs-Maschine für jede Teilziel-resolvierende Wissensbasis eine Kurzfassung (z.B. die verwendeten Klauseln) und einen *link* anbieten, zusammen mit den Antwortbindungen erfolgreicher Refutationen. Somit wird statt einer Schlüsselwort-Indexierung eine geeignete Klauselkopf-Indexierung benötigt. Stärker vom *document retrieval* abweichend, kann ein Klauselrumpf (die Regelprämissen) als eine Folge abstrakter (lokaler oder gloabler) Wissensbasis-*links* gesehen werden, die von der Wissensfolgerungs-Maschine in weiteren Resolutionsschritten verfolgt werden müssen. Selbst für erfolglose Refutationsversuche können die gefundenen Wissensbasis-*links* wertvoll sein, denn das Stellen von verfeinerten Anfragen an den möglichen eigenen Folgerungs-*Server* einer der neu entdeckten Wissensbasen, oder an ihre heruntergeladene Version, oder einfach das *Browsen* durch sie kann zu Problemumformulierungen und unerwarteten Lösungen führen.

Eine derartige Nutzung Web-verteilten Wissens könnte theoretisch über Schlüsselwortsuche implementiert werden (auf einer niedrigen Ebene sind auch Atome Zeichenketten). Praktisch ist die Realisierung eine interessante Herausforderung an die Logische Programmierung.[4]

3. Entwurfsfragen für eine Horn-Suchmaschine

Unten folgt eine erste Skizzierung einer möglichen Suchmaschine für Wissensbasen (KBs) im Web, kurz einer **KB-Suchmaschine**.

Diese wäre schon nützlich, wenn sie nur die formale (hornlogische) Syntax von Wissensbasiseinträgen besser berücksichtigen würde als heutige Text-Suchmaschinen. Hier sollen aber erste semantische und inferentielle Fähigkeiten hinzukommen.

Zunächst ist die **Typisierung** von Wissensbasen hilfreich, etwa durch eine (HORN-)KB-Kennzeichnung von bzw. in ihren HTML-Seiten, z.B. bereits im *header*, um bei der Suche nur relevante, d.h. KB-Seiten, zu betrachten (umgekehrt könnten 'Nur Text'-Suchmaschinen so KB-Bereiche überspringen). Eine Variante, zumindest für Datalog, ist der in HTML 3.2 vorhandene *tag* **table**, mit dem Prädikat-Definitionen als Tabellen dargestellt werden können (aber ein Übergang zum allgemeineren SGML könnte auch für diese Web-Verwendung vorteilhaft sein, indem *generische Identifikatoren* für KBs, Prädikate, Klauseln und vielleicht darunter genutzt werden). In jedem Fall ist es analog zu einem SQL-Datenbankschema sinnvoll, die Signaturen von n-ären Prädikaten zu deklarieren, um die Bedeutung der n Argumente festzulegen. Die hierfür verwendbaren *Sorten* können ihrerseits als unäre Prädikate in Sortenverbänden definiert sein (z.B. durch Regeln oder binäre **subsumes**-Relationen zwischen den Prädikaten). Die KB-Suchmaschine prüft nun vor der Verwendung

---

[3] Auch weitere nichtdeduktive Verwendungen solcher Wissensbasen sind sinnvoll, z.B. durch Lernalgorithmen auf der Basis Induktiver Logischer Programmierung [1]. Wenn einmal Wissen im Web (Horn-)formalisiert ist, könnte insbesondere die unten skizzierte Suchmaschine - während ihrer periodischen Indexierung der verteilten Wissensseiten - auch neue Zusammenhänge zwischen Wissenseinheiten in geographisch auseinanderliegenden Seiten vorschlagen, um vielleicht deren Autoren untereinander in Kontakt zu bringen; ein einfaches Beispiel eines Zusammenhangs wäre die Verwendung identischer Prädikatnamen, wie unten diskutiert.

[4] Umgekehrt wie bei der häufiger untersuchten Verwaltung von beliebigen Web-Seiten **durch** (extra)logische Programme (z.B. [6], [11]), werden hier Web-Erweiterungen **für** (horn)logische Programme mit beliebigen Implementationssprachen betrachtet. Beide LP-WWW-Zusammenhänge könnten prinzipiell auch gleichzeitig verfolgt werden.

einer gefundenen Prädikat-Definition zunächst deren Signaturkompatibilität zum evtl. seinerseits sortierten Ziel-Atom. Diese optionale Verwendung von Ordnungs-Sortierung zur **semantischen Vorfilterung** vermeidet elementare Prädikat-Mehrdeutigkeiten, wie man sie von den entsprechenden leidigen Schlüsselwort-Mehrdeutigkeiten heutiger Text-Suchmaschinen kennt. Damit ist über formalen KBs prinzipiell eine höhere *Präzision* erreichbar als über natürlichsprachlichen Texten. Expertengruppen könnten sich - wiederum mit Hilfe des Internets - nach dem Muster der gemeinsamen FAQ-Erarbeitung für Newsgroups auf ihr Spezialvokabular und ihren Sortenunterverband einigen, mit Schnittstellen zu verwandten Wissensgebieten. Allmählich könnte so die Mehrdeutigkeit von Prädikatnamen verringert werden.

Das **inferentielle Kernproblem** einer KB-Suchmaschine im Web entspricht der Problematik *ODER-paralleler Architekturen* wie Aurora [13], deren Ressourcen zwischen lokaler Suche (normalerweise sequentiell) und globaler Suche (parallel, kommunikationsintensiv) ausgewogen werden müssen. Die KB-Suchmaschine muß nämlich die Resolution auf einer (gefundenen) KB-Seite mit der Web-Bereitstellung von (weiteren) KB-Seiten geeignet verzahnen. Hier existiert auch ein Querbezug zum Thema *modulare/kontextuelle LP* [5]: Wann soll ein Ziel lokal in der aktuellen KB-Seite (Modul) bewiesen werden, wann soll global auf andere KB-Seiten (Module) übergegangen werden? Wann soll Backtracking über Seiten(Modul)-Grenzen hinweggehen? Während vor allem für rekursive Regeln eine Seiten-lokale Abfolge von Resolutionsschritten sinnvoll ist, muß sich der globale Suchprozeß ständig weiter parallel über die KB-Seiten ausbreiten (z.B. mit Algorithmen wie *Fish-Search* [14]). Lokale Nicht-Terminierungen können dann den globalen Erfolg einer Anfrage nicht gefährden: Selbst wenn man voraussetzen dürfte, daß alle Web-publizierten Wissensbasen einzeln terminieren würden, so könnten sich dennoch beim Zusammenspiel verteilter Wissensbasen Nicht-Terminierungen ergeben, z.B. durch zwei verteilte korekursive Regeln. Da die (modelltheoretische) Semantik einer Prädikat-Definition trotz Signatur-Deklarationen schwer zu fassen ist, wenn ihre Klauseln über mehrere Wissensbasen verteilt sind, bietet sich an, hier mehr oder weniger starke Einschränkungen zu vereinbaren: Das strengste **Lokalitätsprinzip** würde die Prädikat-Verteilung überhaupt verbieten, ein mittleres nur die Fakten-Verteilung erlauben und das schwächste auch verteilte nichtrekursive Regeln tolerieren. Weiterhin stellt sich die wichtige Frage, ob zur Vereinheitlichung der Semantik statt der Prolog-üblichen Verwendung von Backtracking auch die Klauseln **innerhalb** einer KB-Seite im Sinne von reiner Hornlogik ODER-parallel (breitenorientiert) interpretiert werden sollen. In jedem Fall würde man Lösungen eher Lycos-artig gruppenweise als Prolog-artig durch individuelles Nachfordern erwarten; in ihrer Reihenfolge könnten sich auch Sicherheitsgewichtungen, ausgehend von den verwendeten Fakten, widerspiegeln. Wegen der Offenheit und Dynamik des WWWs würde sich die Frage der logischen 'Vollständigkeit' von Lösungsmengen kaum stellen: Die KB-Suchmaschine muß sogar die noch laufenden (evtl. endlosen) Suchprozesse abbrechen, wenn keine (weitere) Lösungsgruppe nachgefordert wird; eine SQL-artige (Gesamt-)Mengenorientierung wäre also hier nicht sinnvoll.

Schließlich ist zu untersuchen, wie die **Indexierungstechniken** von Text-Suchmaschinen, Datenbanken und Prolog auf die KB-Suchmaschine übertragbar sind. Innerhalb einer KB-Seite können die den Prädikat-Definitionen ('Prozeduren') vorangestellten Signatur-Deklarationen die Rolle von indexierten HTML-*headings* spielen. Eine feinere KB-Indexierung bis zur Ebene einzelner Klauseln wäre analog zu einer vollständigeren Text-Indexierung denkbar, mit Alta Vista-ähnlichen Vor- und Nachteilen.

Die oben skizzierte KB-Suchmaschine arbeitet im wesentlichen *interpretativ* auf KB-Sourcen, die auch mit normalen Browsern benutzt werden können. Dies ermöglicht über die Black Box-Anfrage hinaus auch die direkte Inspektion und Montage von Wissensbasen nach dem Vorbild von Textbausteinen. Erst durch diese direkte Manipulierbarkeit (hornlogisch) standardisierten Wissens haben Wissensbasen eine gewisse Chance, zum Allgemeingut in der Art von EXCEL-Tabellen zu werden: Dazu ist nicht nur die Durchführung von Wissensbasis**anfragen** (Tabellen**kalkulationen**) wichtig, sondern auch die Unterstützung von **Aufbau**, **Änderung**, **Kombination** und **bidirektionaler Textkopplung** von Wissensbasen (Tabellen). Dies könnte durch die Hypertext-Mechanismen des Web unterstützt werden. So müßten etwa Untermodule einer Wissensbasis sowohl von der KB-Suchmaschine als auch interaktiv über einen speziellen HTML-*link* erreichbar sein. Wenn aus Effizienzgründen die *Kompilation* einer Hornlogik-Wissensbasis erforderlich wird, könnte versucht werden, die direkte Manipulierbarkeit zu simulieren, z.B. durch Source-Rückverweise und automatische Rekompilation bei Änderungen. Bei einer WAM-Kompilation bliebe gegenüber einer noch

effizienteren Native Code-Kompilation immerhin der erstmals von Java genutzte Vorteil der Pseudocode-Versendung über das WWW und seiner Client-seitigen Emulator-Ausführung.

4. Eine standardisierte Infomaster-Anpassung

Eine Möglichkeit zur Implementierung der KB-Suchmaschine bestünde in einer Anpassung der von Michael Genesereth und der Logic Group an der Stanford University entwickelten *Infomaster*^{TM}-Technologie [7]. Infomaster ist ein *virtuelles Informationssystem* für den Zugriff auf verteilte heterogene Datenbanken und Wissensbasen, implementiert in der Agenten-Kommunikations-Sprache ACL, einer Kombination aus KQML, KIF und Ontologien. Dazu werden *Facilitatoren* benutzt, die - in unserer Sprechweise - ACL-(konvertierte-)Ziele zerlegen (z.B. in konjunktive Teilziele), an Faktenbasen und andere Facilitatoren weiterleiten und verschiedene Repräsentationen ineinander übersetzen.

Einerseits kann die KB-Suchmaschine als eine Spezialisierung von Infomaster gesehen werden, in dem Sinn daß aufgrund des angestrebten Hornlogik-KB-Standards die Übersetzungsfunktion von Facilitatoren nur eingeschränkt benötigt wird (s.u.). Andererseits strebt Infomaster in Erweiterung seiner Verwendung für das Stanford Information Network an, Basistechnologie für ein Welt-Informations-Netzwerk (WIN) zu werden, das offenbar im Wettbewerb zum WWW aufgebaut werden soll; stattdessen soll hier die vorhandene allgemeine Web-Infrastruktur vernetzter HTML-Seiten direkt für Wissensbasen genutzt werden.

Somit entsteht für die Infomaster-Implementierung der KB-Suchmaschine die Notwendigkeit einer Übersetzung zwischen der gewählten Hornlogik-HTML-Syntax und ACL. Dafür wurden aber bereits erhebliche Vorarbeiten geleistet, z.B. mit den in unserer Gruppe entwickelten Übersetzern zwischen purem Prolog und einer Teilmenge von KIF (über Relfuns Prolog- und Lisp-artige Syntaxen). Diese Übersetzer wurden im Rahmen einer Stanford-Kaiserslautern-Zusammenarbeit bzgl. KIF und Facilitatoren bereitgestellt. Da sie - wie die Infomaster-Software - in Lisp programmiert sind, erscheint eine Integration einfach.

Aus der Sicht von Infomaster sind verteilte Hornlogik-Wissensbasen eine standardisierte Vereinfachung der dort zugelassenen Notations-Heterogenität. Umgekehrt vermeiden wir die folgenden zusätzlichen Problemen dieser Heterogenität:
a) Eine direkte Manipulation von gefundenen Teil-KBs nach dem Vorbild von Textbausteinen und EXCEL-Tabellen wird erschwert.
b) Die Übersetzung von gefundenen Teil-KBs in andere Repräsentationen muß nicht umkehrbar eindeutig sein, wodurch Rückübersetzungsprobleme bei der Fehlerbehandlung und Wartung entstehen.
c) Dynamische Übersetzungen führen zu schlechterem Laufzeitverhalten als eine einmalige, statische Wissens-'Standardisierung'.
d) Jede Wissensbasis, die in einem einzigen standardisierten Format (weiter-)gewartet wird, erspart die zusätzliche Pflege eines Paares von Übersetzern.

Natürlich erzielt die Auseinandersetzung mit heterogenen **Notationen** eine größere Allgemeinheit. Allerdings wird es, wie oben diskutiert, sogar innerhalb der standardisierten Hornlogik-Notation schwierig sein, die Heterogenität von **Vokabularen** (z.B. Sorten und Signaturen) zu verringern, welche in einer weltweit verteilten Anordnung zu erwarten ist. Die Realisierung der KB-Suchmaschine über eine angepaßte Infomaster-Technologie erlaubt eine eingehendere Untersuchung dieser Tradeoffs zwischen Standardisierung und Heterogenität.

5. Internet-Erfahrung und Intranet-Aussichten

Wegen der vergleichsweise einfachen Formalisierbarkeit von technischem Wissen liegt es nahe, Hornlogik-Wissensbasen zunächst für diesen Bereich aufzubauen. Im Projekt VEGA haben wir dies vor allem in Umwelt-Teilbereichen der Werkstoffkunde getan (z.B. [2]). Allerdings sind Wissensbasen wie `RTPLAST` im Web bisher im wesentlichen als reine ASCII-Dateien (in Relfun-Syntax) publiziert. Die Aufnahme weiterer Hornlogik-Wissensbasen ist geplant, z.B. zum Vergleich funktional-logischer Programme. Es ist aber noch offen, welche

operationale Variante der Hornlogik für die Publikation praktischer Wissensbasen im Web am besten geeignet ist: Die Erfahrung spricht etwa für eine Beachtung der textuellen Anordnung von Klauseln (ODER-Sequentialität) und ihrer Prämissen (UND-Sequentialität), während die verteilte Web-Suche eine zumindest ODER-parallele Interpretation nahelegt. Nach den hier notwendigen Festlegungen muß insbesondere eine genauere Spezifikation der HTML-Zusätze für Hornlogik-Wissensbasen und eine Überarbeitung der darauf operierenden KB-Suchmaschine geleistet werden, bevor mit einer prototypischen Implementation (etwa in Java) begonnen werden kann. Aus Akzeptanzgründen spielt natürlich die Effizienz der KB-Suchmaschine eine große Rolle; diese könnte sowohl durch Fortschritte in der LP (z.B. bessere ODER-parallele Strategien) als auch beim WWW (z.B. bessere Java-Effizienz) gesteigert werden.

Auch in der Form von *Intranets* zur **organisationsinternen** Informationsbereitstellung auf Basis der Web-Technologie bieten sich interessante Möglichkeiten für WWW-Wissensbasen. In Analogie zu einem zentralen *corporate memory* [9] wären die Intranet-Wissensbasen einer (über LAN oder WAN) vernetzten Organisation nämlich die Wissenskomponente eines *verteilten Unternehmensgedächtnisses*. Die Zeit und Geld kostenden Informationsdefizite in größeren Organisationen zeigen, daß auch für Intranet-Wissensbasen eine KB-Suchmaschine sinnvoll wäre, die hier aufgrund der kleineren Wissensvolumina schneller arbeiten würde als im offenen Internet. Wenn Organisationen dabei auch als internes Repräsentationsformat den extern vereinbarten (Hornlogik-)Standard verwenden, dann können sie Wissensbasen bedarfsweise zwischen ihrem Intranet und dem öffentlichen Internet hin- und herbewegen: Nach innen zur Übernahme bzw. Anpassung von neuem Know-how; nach außen zur Demonstration (von Teilmengen oder früheren Versionen) und evtl. zum (kryptographischen, Internet-Zahlungsverkehrs-)Vertrieb ihrer Expertise. Analog ermöglicht eine vereinheitlichte Repräsentation flexible Allianzen zwischen wissensintensiven Stiftungen oder Firmen.

6. Web-Wissensbasen: Eine Gemeinschaftsanstrengung

Abgesehen von den technischen Fragen, die im Hauptteil dieser Arbeit diskutiert wurden, hat der Aufbau von Wissensbasen im World Wide Web eine starke 'Gemeinschafts'-Komponente: Während das Interesse am Anbieten und an der Verwendung von Inter- oder Intranet-KBs wächst, sollten wir zu einem standardisierten Repräsentationsformat konvergieren. Da die Hornlogik bereits die Sprache der Wahl für viele nicht-Web-KB-Publikationen ist und sich im Kern vieler LP-Erweiterungen befindet, erscheint sie als der richtige Ausgangspunkt. Wenn Web-KB-Anbieter Prolog-Syntax benutzen, sind ihre KBs in das endgültige Format übersetzbar, das von der ins Auge gefaßten Suchmaschine für Hornlogik-KBs indexiert und abgefragt wird, da bereits eine hinreichende Zahl von 'überlieferten' Prolog-KBs existiert, die ebenfalls einen solchen Übersetzer benötigen. Der schwerere Teil beim Versuch wiederverwendbare Web-Horn-KBs aufzubauen wird wahrscheinlich in gemeinsamen Vokabular- oder 'Ontologie'-Entscheidungen bezüglich kompatibler Definitionen von Prädikatnamen bestehen. Die Sorten, die für die hierarchische Strukturierung von nichttrivialen Domänen benötigt werden, können jedoch der Einheitlichkeit halber als unäre Prädikate betrachtet werden, die wiederum durch (einfache) Hornregeln definiert sind. Sobald einmal eine Suchmaschine für typische Ziele genügend viele Klauseln antrifft, kann ein Selbstbeschleunigungseffekt beim Wachstum von Web-KBs eintreten, ähnlich dem, den wir z.Zt. für andere Informationskategorien im Web beobachten. Dies kann man durch Kopplung (essentieller) Hornklauseln mit natürlichsprachlichen Texten, die wie eine inline-Dokumentation agieren, weiter ermutigen: Wenn solche Texte von normalen Suchmaschinen gefunden werden, gewöhnen sich Web-Benutzer an die formalere Klausel-Repräsentation von Wissen.

Diese Arbeit hat versucht zu zeigen, daß die LP-Gemeinschaft für eine solche umfassende Web-Anstrengung prinzipiell gut vorbereitet ist. Unter den Ansätzen, eine Web-Infrastruktur für die Wiederverwendung von logischen Programmen oder KBs zu schaffen, kommt der vorliegende Vorschlag am nächsten zu dem unabhängig in [12] entwickelten: “Lightweight deductive databases” sind deduktive Datenbanken, die in Web-Seiten inkorporiert werden, um eine Quelle verteilter strukturierter Information bereitzustellen. Ihre Hauptmerkmale, *verteilte Wartung, Erweiterungsmöglichkeit* und *Wiederverwendbarkeit*, sowie Konzepte aus der modularen LP werden von unserem Vorschlag geteilt. Wie andere Suchmaschinen behalten wir jedoch *Server-seitige Verarbeitung* als Default bei. Zusätzlich betonen wir *Purität* in dem Sinn, daß keine zusätzlichen Operatoren, z.B. um Ziele in anderen Modulen aufzurufen, benutzt werden. Stattdessen sollte eine pure Hornlogik-KB unverändert in eine KB-

gekennzeichnete Web-Seite ladbar sein, wobei die KB-Suchmaschine die modulare Struktur der Seiten behandelt, ähnlich wie es Text-Suchmaschinen tun. Purität unterstützt auch *ODER-Parallelität* als Paradigma zur Ausnutzung Web-verteilter KBs, das durch 'Agenten-Replikation' - wie etwa in Aurora [13] - implementierbar ist. Schließlich halten wir es für wichtig, daß eine KB-Suchmaschine auch Lösungen 'erklären' kann, indem die für eine Deduktion benutzten Klauseln im Kontext ihrer KB-Seiten hervorgehoben werden.

Was wir insgesamt vorschlagen ist keine neue LP-Sprache für das Web, sondern die pragmatische Wiederverwendung von Hornlogik-KBs - in einer Prolog-artigen Syntax - durch eine neue Suchmaschine. Sobald einmal etwas Erfahrung mit der Web-Suche solcher KBs angesammelt ist, könnten Web-orientierte - aber immer noch pure - Erweiterungen der Hornlogik-Kernsprache sorgfältig eingeführt werden. Aber bevor wir in die akribischen Details - und Auseinandersetzungen - der Definition einer 'Web-LP'-Sprache gehen, sollten wir mit einer oder mehreren Suchmaschinen für eine feste (Horn-)Sprache experimentieren.

Literatur

[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*, Seiten 29-42, TMS, 1994.
http://www.dfki.uni-kl.de/~vega/relfun.html (Examples: RTPLAST)

[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*, 19/20:443-502, 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*, Seiten 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.