Shriram Krishnamurthi
C. R. Ramakrishnan (Eds.)

Practical Aspects of Declarative Languages

LNCS 2257

4th International Symposium, PADL 2002
Portland, OR, USA, January 2002
Proceedings

Springer

Shriram Krishnamurthi
C.R. Ramakrishnan (Eds.)

Practical Aspects of Declarative Languages

4th International Symposium, PADL 2002
Portland, OR, USA, January 19-20, 2002
Proceedings

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Shriram Krishnamurthi
Brown University, Computer Science Department
Box 1910, Providence, RI 02912, USA
E-mail: sk@cs.brown.edu

C.R. Ramakrishnan
SUNY at Stony Brook, Department of Computer Science
Stony Brook, NY 11794-4400, USA
E-mail: cram@cs.sunysb.edu

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Practical aspects of declarative languages : 4th international symposium ; proceedings / PADL 2002, Portland, OR, USA, January 19 - 20, 2002. Shriram Krishnamurthi ; C. R. Ramakrishnan (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2002
(Lecture notes in computer science ; Vol. 2257)
ISBN 3-540-43092-X

CR Subject Classification (1998):D.3, D.1, F.3, D.2

ISSN 0302-9743
ISBN 3-540-43092-X Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by DA-TeX Gerd Blumenstein
Printed on acid-free paper SPIN 10846034 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 2257

Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Preface

Declarative languages build on sound theoretical bases to provide attractive frameworks for application development. These languages have been successfully applied to a wide variety of real-world situations including database management, active networks, software engineering, and decision-support systems.

New developments in theory and implementation expose fresh opportunities. At the same time, the application of declarative languages to novel problems raises numerous interesting research issues. These well-known questions include scalability, language extensions for application deployment, and programming environments. Thus, applications drive the progress in the theory and implementation of declarative systems, and in turn benefit from this progress.

The International Symposium on Practical Applications of Declarative Languages (PADL) provides a forum for researchers, practitioners, and implementors of declarative languages to exchange ideas on current and novel application areas and on the requirements for effective use of declarative systems. The fourth PADL symposium was held in Portland, Oregon, on January 19 and 20, 2002.

Thirty-seven papers were submitted in response to the call for papers. Each paper was reviewed by at least three referees. Eighteen papers were selected for presentation at the symposium. The symposium included invited talks by Veronica Dahl (Simon Fraser University) on "How to Talk to Your Computer so that It Will Listen"; Catherine Meadows (Naval Research Laboratory) on "Using a Declarative Language to Build an Experimental Analysis Tool"; and J. Strother Moore (University of Texas-Austin) on "Single-Threaded Objects in ACL2". Every member of the program committee went the extra mile to give constructive, detailed feedback on submitted papers. Additional reviewers were brought in to help the program committee evaluate the submissions. We gratefully acknowledge their service.

This workshop was co-located with the ACM Symposium on Principles of Programming Languages (POPL 2002). It was sponsored by COMPULOG AMERICAS, a network of research groups dedicated to promoting research in logic programming and related areas, by the Association for Logic Programming (ALP), the ACM, and the European Association for Programming Languages and Systems (EAPLS). We also thank Brown University, SUNY at Stony Brook, and the University of Texas at Dallas for generously making their resources available for the organization of the symposium. The support of many individuals was crucial to the success of the symposium. We thank John Launchbury (POPL general chair) and Kelly Atkinson (Conference Secretary) for general organizational help. We thank Gopal Gupta, the conference chair, for coordinating the organization of the symposium. We also thank Paul Graunke, who helped us develop and manage the software used to submit and review papers, and Samik Basu, who assisted in putting the final proceedings together.

November 2001

Shriram Krishnamurthi
C. R. Ramakrishnan

Program Committee

Sergio Antoy	Portland State University, USA
Gopal Gupta	University of Texas at Dallas, USA (General Chair)
Fergus Henderson	University of Melbourne, Australia
Joxan Jaffar	National University of Singapore
Andrew Kennedy	Microsoft Research, UK
Shriram Krishnamurthi	Brown University, USA (Program Co-chair)
Michael Leuschel	University of Southampton, UK
Kim Marriott	Monash University, Australia
John Peterson	Yale University, USA
Andreas Podelski	MPI, Germany
Enrico Pontelli	New Mexico State University, USA
C.R. Ramakrishnan	SUNY, Stony Brook, USA (Program Co-chair)
John Reppy	Bell Labs, Lucent Technologies, USA
Manuel Serrano	Université de Nice, France
Olin Shivers	Georgia Institute of Technology, USA
Paul Tarau	University of North Texas, USA

Referees

David McAllester
Marcello Balduccini
Manuel Carro
W.N. Chin
W. Rance Cleaveland
Steve Crouch
Paul Graunke
Stefan Gruner
Haifeng Guo
Pascual Julian Iranzo
Bharat Jayaraman
S.C. Khoo
H.C. Lau
T.Y. Leong
Andrew Lim
Bing Liu
Morgan McGuire
Henrik Nilsson
L. Robert Pokorny
Didier Remy
Vítor Santos Costa
Fernando Silva
Kish Shen
Tran Cao Son
Michael Sperber
Peter Thiemann
Guizhen Yang
Rong Yang
Roland Yap

Lecture Notes in Computer Science

For information about Vols. 1–2179
please contact your bookseller or Springer-Verlag

Vol. 2180: J. Welch (Ed.), Distributed Computing. Proceedings, 2001. X, 343 pages. 2001.

Vol. 2181: C. Y. Westort (Ed.), Digital Earth Moving. Proceedings, 2001. XII, 117 pages. 2001.

Vol. 2182: M. Klusch, F. Zambonelli (Eds.), Cooperative Information Agents V. Proceedings, 2001. XII, 288 pages. 2001. (Subseries LNAI).

Vol. 2183: R. Kahle, P. Schroeder-Heister, R. Stärk (Eds.), Proof Theory in Computer Science. Proceedings, 2001. IX, 239 pages. 2001.

Vol. 2184: M. Tucci (Ed.), Multimedia Databases and Image Communication. Proceedings, 2001. X, 225 pages. 2001.

Vol. 2185: M. Gogolla, C. Kobryn (Eds.), «UML» 2001 – The Unified Modeling Language. Proceedings, 2001. XIV, 510 pages. 2001.

Vol. 2186: J. Bosch (Ed.), Generative and Component-Based Software Engineering. Proceedings, 2001. VIII, 177 pages. 2001.

Vol. 2187: U. Voges (Ed.), Computer Safety, Reliability and Security. Proceedings, 2001. XVI, 249 pages. 2001.

Vol. 2188: F. Bomarius, S. Komi-Sirviö (Eds.), Product Focused Software Process Improvement. Proceedings, 2001. XI, 382 pages. 2001.

Vol. 2189: F. Hoffmann, D.J. Hand, N. Adams, D. Fisher, G. Guimaraes (Eds.), Advances in Intelligent Data Analysis. Proceedings, 2001. XII, 384 pages. 2001.

Vol. 2190: A. de Antonio, R. Aylett, D. Ballin (Eds.), Intelligent Virtual Agents. Proceedings, 2001. VIII, 245 pages. 2001. (Subseries LNAI).

Vol. 2191: B. Radig, S. Florczyk (Eds.), Pattern Recognition. Proceedings, 2001. XVI, 452 pages. 2001.

Vol. 2192: A. Yonezawa, S. Matsuoka (Eds.), Metalevel Architectures and Separation of Crosscutting Concerns. Proceedings, 2001. XI, 283 pages. 2001.

Vol. 2193: F. Casati, D. Georgakopoulos, M.-C. Shan (Eds.), Technologies for E-Services. Proceedings, 2001. X, 213 pages. 2001.

Vol. 2194: A.K. Datta, T. Herman (Eds.), Self-Stabilizing Systems. Proceedings, 2001. VII, 229 pages. 2001.

Vol. 2195: H.-Y. Shum, M. Liao, S.-F. Chang (Eds.), Advances in Multimedia Information Processing – PCM 2001. Proceedings, 2001. XX, 1149 pages. 2001.

Vol. 2196: W. Taha (Ed.), Semantics, Applications, and Implementation of Program Generation. Proceedings, 2001. X, 219 pages. 2001.

Vol. 2197: O. Balet, G. Subsol, P. Torguet (Eds.), Virtual Storytelling. Proceedings, 2001. XI, 213 pages. 2001.

Vol. 2198: N. Zhong, Y. Yao, J. Liu, S. Ohsuga (Eds.), Web Intelligence: Research and Development. Proceedings, 2001. XVI, 615 pages. 2001. (Subseries LNAI).

Vol. 2199: J. Crespo, V. Maojo, F. Martin (Eds.), Medical Data Analysis. Proceedings, 2001. X, 311 pages. 2001.

Vol. 2200: G.I. Davida, Y. Frankel (Eds.), Information Security. Proceedings, 2001. XIII, 554 pages. 2001.

Vol. 2201: G.D. Abowd, B. Brumitt, S. Shafer (Eds.), Ubicomp 2001: Ubiquitous Computing. Proceedings, 2001. XIII, 372 pages. 2001.

Vol. 2202: A. Restivo, S. Ronchi Della Rocca, L. Roversi (Eds.), Theoretical Computer Science. Proceedings, 2001. XI, 440 pages. 2001.

Vol. 2203: A. Omicini, P. Petta, R. Tolksdorf (Eds.), Engineering Societies in the Agents World II. Proceedings, 2001. XI, 195 pages. 2001. (Subseries LNAI).

Vol. 2204: A. Brandstädt, V.B. Le (Eds.), Graph-Theoretic Concepts in Computer Science. Proceedings, 2001. X, 329 pages. 2001.

Vol. 2205: D.R. Montello (Ed.), Spatial Information Theory. Proceedings, 2001. XIV, 503 pages. 2001.

Vol. 2206: B. Reusch (Ed.), Computational Intelligence. Proceedings, 2001. XVII, 1003 pages. 2001.

Vol. 2207: I.W. Marshall, S. Nettles, N. Wakamiya (Eds.), Active Networks. Proceedings, 2001. IX, 165 pages. 2001.

Vol. 2208: W.J. Niessen, M.A. Viergever (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001. Proceedings, 2001. XXXV, 1446 pages. 2001.

Vol. 2209: W. Jonker (Ed.), Databases in Telecommunications II. Proceedings, 2001. VII, 179 pages. 2001.

Vol. 2210: Y. Liu, K. Tanaka, M. Iwata, T. Higuchi, M. Yasunaga (Eds.), Evolvable Systems: From Biology to Hardware. Proceedings, 2001. XI, 341 pages. 2001.

Vol. 2211: T.A. Henzinger, C.M. Kirsch (Eds.), Embedded Software. Proceedings, 2001. IX, 504 pages. 2001.

Vol. 2212: W. Lee, L. Mé, A. Wespi (Eds.), Recent Advances in Intrusion Detection. Proceedings, 2001. X, 205 pages. 2001.

Vol. 2213: M.J. van Sinderen, L.J.M. Nieuwenhuis (Eds.), Protocols for Multimedia Systems. Proceedings, 2001. XII, 239 pages. 2001.

Vol. 2214: O. Boldt, H. Jürgensen (Eds.), Automata Implementation. Proceedings, 1999. VIII, 183 pages. 2001.

Vol. 2215: N. Kobayashi, B.C. Pierce (Eds.), Theoretical Aspects of Computer Software. Proceedings, 2001. XV, 561 pages. 2001.

Vol. 2216: E.S. Al-Shaer, G. Pacifici (Eds.), Management of Multimedia on the Internet. Proceedings, 2001. XIV, 373 pages. 2001.

Vol. 2217: T. Gomi (Ed.), Evolutionary Robotics. Proceedings, 2001. XI, 139 pages. 2001.

Vol. 2218: R. Guerraoui (Ed.), Middleware 2001. Proceedings, 2001. XIII, 395 pages. 2001.

Vol. 2219: S.T. Taft, R.A. Duff, R.L. Brukardt, E. Ploedereder (Eds.), Consolidated Ada Reference Manual. XXV, 560 pages. 2001.

Vol. 2220: C. Johnson (Ed.), Interactive Systems. Proceedings, 2001. XII, 219 pages. 2001.

Vol. 2221: D.G. Feitelson, L. Rudolph (Eds.), Job Scheduling Strategies for Parallel Processing. Proceedings, 2001. VII, 207 pages. 2001.

Vol. 2223: P. Eades, T. Takaoka (Eds.), Algorithms and Computation. Proceedings, 2001. XIV, 780 pages. 2001.

Vol. 2224: H.S. Kunii, S. Jajodia, A. Sølvberg (Eds.), Conceptual Modeling – ER 2001. Proceedings, 2001. XIX, 614 pages. 2001.

Vol. 2225: N. Abe, R. Khardon, T. Zeugmann (Eds.), Algorithmic Learning Theory. Proceedings, 2001. XI, 379 pages. 2001. (Subseries LNAI).

Vol. 2226: K.P. Jantke, A. Shinohara (Eds.), Discovery Science. Proceedings, 2001. XII, 494 pages. 2001. (Subseries LNAI).

Vol. 2227: S. Boztaş, I.E. Shparlinski (Eds.), Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. Proceedings, 2001. XII, 398 pages. 2001.

Vol. 2228: B. Monien, V.K. Prasanna, S. Vajapeyam (Eds.), High Performance Computing – HiPC 2001. Proceedings, 2001. XVIII, 438 pages. 2001.

Vol. 2229: S. Qing, T. Okamoto, J. Zhou (Eds.), Information and Communications Security. Proceedings, 2001. XIV, 504 pages. 2001.

Vol. 2230: T. Katila, I.E. Magnin, P. Clarysse, J. Montagnat, J. Nenonen (Eds.), Functional Imaging and Modeling of the Heart. Proceedings, 2001. XI, 158 pages. 2001.

Vol. 2232: L. Fiege, G. Mühl, U. Wilhelm (Eds.), Electronic Commerce. Proceedings, 2001. X, 233 pages. 2001.

Vol. 2233: J. Crowcroft, M. Hofmann (Eds.), Networked Group Communication. Proceedings, 2001. X, 205 pages. 2001.

Vol. 2234: L. Pacholski, P. Ružička (Eds.), SOFSEM 2001: Theory and Practice of Informatics. Proceedings, 2001. XI, 347 pages. 2001.

Vol. 2235: C.S. Calude, G. Păun, G. Rozenberg, A. Salomaa (Eds.), Multiset Processing. VIII, 359 pages. 2001.

Vol. 2236: K. Drira, A. Martelli, T. Villemur (Eds.), Cooperative Environments for Distributed Systems Engineering. IX, 281 pages. 2001.

Vol. 2237: P. Codognet (Ed.), Logic Programming. Proceedings, 2001. XI, 365 pages. 2001.

Vol. 2239: T. Walsh (Ed.), Principles and Practice of Constraint Programming – CP 2001. Proceedings, 2001. XIV, 788 pages. 2001.

Vol. 2240: G.P. Picco (Ed.), Mobile Agents. Proceedings, 2001. XIII, 277 pages. 2001.

Vol. 2241: M. Jünger, D. Naddef (Eds.), Computational Combinatorial Optimization. IX, 305 pages. 2001.

Vol. 2242: C.A. Lee (Ed.), Grid Computing – GRID 2001. Proceedings, 2001. XII, 185 pages. 2001.

Vol. 2243: G. Bertrand, A. Imiya, R. Klette (Eds.), Digital and Image Geometry. VII, 455 pages. 2001.

Vol. 2244: D. Bjørner, M. Broy, A.V. Zamulin (Eds.), Perspectives of System Informatics. Proceedings, 2001. XIII, 548 pages. 2001.

Vol. 2245: R. Hariharan, M. Mukund, V. Vinay (Eds.), FST TCS 2001: Foundations of Software Technology and Theoretical Computer Science. Proceedings, 2001. XI, 347 pages. 2001.

Vol. 2246: R. Falcone, M. Singh, Y.-H. Tan (Eds.), Trust in Cyber-societies. VIII, 195 pages. 2001. (Subseries LNAI).

Vol. 2247: C. P. Rangan, C. Ding (Eds.), Progress in Cryptology – INDOCRYPT 2001. Proceedings, 2001. XIII, 351 pages. 2001.

Vol. 2248: C. Boyd (Ed.), Advances in Cryptology – ASIACRYPT 2001. Proceedings, 2001. XI, 603 pages. 2001.

Vol. 2249: K. Nagi, Transactional Agents. XVI, 205 pages. 2001.

Vol. 2250: R. Nieuwenhuis, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. Proceedings, 2001. XV, 738 pages. 2001. (Subseries LNAI).

Vol. 2251: Y.Y. Tang, V. Wickerhauser, P.C. Yuen, C.Li (Eds.), Wavelet Analysis and Its Applications. Proceedings, 2001. XIII, 450 pages. 2001.

Vol. 2252: J. Liu, P.C. Yuen, C. Li, J. Ng, T. Ishida (Eds.), Active Media Technology. Proceedings, 2001. XII, 402 pages. 2001.

Vol. 2253: T. Terano, T. Nishida, A. Namatame, S. Tsumoto, Y. Ohsawa, T. Washio (Eds.), New Frontiers in Artificial Intelligence. Proceedings, 2001. XXVII, 553 pages. 2001. (Subseries LNAI).

Vol. 2254: M.R. Little, L. Nigay (Eds.), Engineering for Human-Computer Interaction. Proceedings, 2001. XI, 359 pages. 2001.

Vol. 2255: J. Dean, A. Gravel (Eds.), COTS-Based Software Systems. Proceedings, 2002. XIV, 257 pages. 2002.

Vol. 2256: M. Stumptner, D. Corbett, M. Brooks (Eds.), AI 2001: Advances in Artificial Intelligence. Proceedings, 2001. XII, 666 pages. 2001. (Subseries LNAI).

Vol. 2257: S. Krishnamurthi, C.R. Ramakrishnan (Eds.), Practical Aspects of Declarative Languages. Proceedings, 2002. VIII, 351 pages. 2002.

Vol. 2258: P. Brazdil, A. Jorge (Eds.), Progress in Artificial Intelligence. Proceedings, 2001. XII, 418 pages. 2001. (Subseries LNAI).

Vol. 2259: S. Vaudenay, A.M. Youssef (Eds.), Selected Areas in Cryptography. Proceedings, 2001. XI, 359 pages. 2001.

Vol. 2260: B. Honary (Ed.), Cryptography and Coding. Proceedings, 2001. IX, 416 pages. 2001.

Vol. 2264: K. Steinhöfel (Ed.), Stochastic Algorithms: Foundations and Applications. Proceedings, 2001. VIII, 203 pages. 2001.

Table of Contents

Using a Declarative Language to Build an Experimental Analysis Tool

Catherine Meadows

Naval Research Laboratory, Code 5543
Washington, DC 20375, USA

Abstract. In this paper we give a brief summary of our experience in using a declarative language, Prolog, to develop an experimental formal analysis tool, the NRL Protocol Analyzer, which was updated and modified over the years to incorporate new theories and techniques. We discuss the benefits of using such an approach, and also some of the downsides....

The application of formal methods to cryptographic protocol analysis is now an established field. The types of assumptions that need to be made, and the techniques for automatically proving properties of cryptographic protocols, are well known, at least for a certain subclass of problems. However, when we began working on this problem in the late 80's, this was definitely not the case. Only a few tools, such as Millen's Interrogator [6], and a few algorithms, such as those devised by Dolev, Even, and Karp, [1], existed. Although these could be used as a basis for my research, it was unclear where we would ultimately wind up. Thus, we needed to ability to build a tool that could be rapidly reconfigured to incorporate new techniques and models, and that updated over (possibly) over a long period of time.

The earliest version of the Analyzer [2] consisted of a simply of a state generation tool. The user specified a state, and the Analyzer would use equational unification to generate all states that immediately preceded it. The search strategy was largely guided by the user, and was input by hand. This was very tedious, but allowed me to collect data that could be used to build the next version of the Analyzer.

The second version of the Analyzer allowed some automatic guidance of the search. In particular, it was possible to write and then use the Analyzer to prove inductive lemmas that put conditions on infinite classes of states. The search could then automatically avoid states that were unreachable according to the lemmas. However, it was up to the user to figure out what lemmas needed to be proved.

As we continued to use the Analyzer, it was found that many of the lemmas obeyed certain canonical forms. This made it easier to automate the generation as well as the proof of lemmas. Thus, the current version of the Analyzer, although it still requires some input from the user, generates most lemmas automatically [3]. It also proves a much greater variety of lemmas than it did before, and supports

S. Krishnamurthi, C. R. Ramakrishnan (Eds.): PADL 2002, LNCS 2257, pp. 1–2, 2002.

a higher-level and more flexible specification language than earlier versions. The most up-to-date description of the Analyzer is given in [4].

Throughout this process, we found the use of a declarative language such as Prolog a great boon. The ease of writing and reading such programs made it easier to update the Analyzer incrementally, over long periods of time, and even with long periods of inactivity. On the other hand, we found that many of the special tricks that can be used to improve Prolog's performance worked against this, and as a result we intended to avoid this after a while. Because of this, and because of other design decisions that we made in order to make this incremental modification easier (the use of generate-and-test as a theorem proving strategy, for example), there are a number of cryptographic protocol analysis tools designed with more specialized applications in mind that outperform the Analyzer. However, we believe that the Analyzer is still one of the most flexible tools around, and it has been used in the analysis of more complex protocols (see for example [4,5]) than almost any other tool. Moreover, many of the newer tools make use of techniques that were pioneered by the NRL Protocol Analyzer.

In summary, we would definitely recommend declarative programming as a rapid prototyping tool, especially one which is expected to undergo major changes as a project progresses. On the downside, the very techniques that would improve such a program's performance appear to mitigate against its usefulness for rapid prototyping by making the program more opaque. However, this is a tradeoff that one might expect.

References

1. D. Dolev, S. Even, and R. Karp. On the Security of Ping-Pong Protocols. *Information and Control*, pages 57–68, 1982.
2. C. Meadows. A system for the specification and verification of key management protocols. In *Proceedings of the 1991 IEEE Symposium in Research in Security and Privacy*. IEEE Computer Society Press, May 1991.
3. Catherine Meadows. Language generation and verification in the NRL protocol analyzer. In *Proceedings of the 9th Computer Security Foundations Workshop*. IEEE Computer Society Press, 1996.
4. Catherine Meadows. Analysis of the Internet Key Exchange protocol using the NRL Protocol Analyzer. In *Proceedings of the 1999 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 1999.
5. Catherine Meadows. Experiences in the formal analyzer of the GDOI protocol. In *Proceedings of Verlassliche IT-Systeme 2001 - Sicherheit in komplexen IT-Infrastrukturen*, 2001.
6. J. K. Millen. The interrogator: A tool for cryptographic protocol security. In *Proc. 1984 Symp. Security and Privacy*, pages 134–141. IEEE Computer Society Press, 1984.

How to Talk to Your Computer so that It Will Listen

Extended Abstract

Veronica Dahl

Logic and Functional Programming Group, Computing Sciences Department
Simon Fraser University, Burnaby, B.C. V5A 1S6, Canada
veronica@cs.sfu.ca

1 Introduction

Currently, many developments revolutionize computing sciences: the maturing of logic programming/grammars, allowing us to communicate with computers in more human and higher level terms than ever before; the World Wide Web; and the possibility of speaking to computers through affordable software such as Naturally Speaking or Microsoft Speech Agent.

The time is ripe to try to integrate these developments, with the final aim of making speech itself the programming language of choice. This article discusses shorter term, more attainable objectives along the route to that final goal, from the perspective of our own personal research interests.

2 Research Directions

Interesting results have recently been obtained in the areas of natural language processing, virtual worlds, internet programming and mining, robotics, deductive knowledge bases, and the combination of these.

Such results indicate good promise for the following shorter term research directions:

1. Automatic creation and consultation of knowledge bases through natural language: We are developing a prototype system that will automatically initialize, update and query a database from speech commands, as well as generate spoken answers, based on initial results reported in [5]. This research branch partially addresses the need of integrating software for speech recognition, speech synthesis, and Artificial Intelligence programs. For the language component, we mainly rely on the grammatical form of Assumptive Logic Programming [34,10,9,14,6,30]. We shall also adapt our deductive database methodologies [17,13] to the task, and integrate them with our abductive reasoning methodologies for syntactic error recovery [2] and for exception handling [19]. We shall also incorporate our treatment of ellipsis and sentence coordination [12,7,15] into Assumption Grammar form, along

S. Krishnamurthi, C. R. Ramakrishnan (Eds.): PADL 2002, LNCS 2257, pp. 3–8, 2002.

the lines sketched in [34], to allow users as natural a dialogue with the computer as possible. This whole line of research is a milestone towards a longer term, more ambitious objective: programming through natural language.

2. Driving Robots through speech: This research branch was explored in a preliminary form in collaboration with Universite de Nice, for the high level routing of mini-robots [8,33] whose lower level operations were commanded in C. Possible applications include endowing robots with language understanding capabilities, including virtual robots (robots that move and execute commands in a visual world), such as explored in preliminary form with Andrea Schiel and Paul Tarau (Generating Internet-based Animations through NL controlled Partial Order Planners- SFU Internal report). Other exisiting prototypes of language-driven robots include the pet dog AIBO developed by Frédéric Kaplan at Sony CSL, Paris, and the Japanese robot of Dr Mizoguchi's team, which offers wine at social gatherings.
3. High level tools for accessing and interacting with the internet, aiming at endowing the web, that fantastic but often frustrating reservoir of knowledge, with intelligent communication capabilities such as:
   - Providing multilingual access to virtual worlds over the internet. A prototype system, LogiMOO [32] accepts interactions in various languages, translates each to a controlled English based interlingua (along the lines sketched in [11,31,21]), and reacts in the language of origin. Among the possible applications, those to distance learning have been outlined in [22], as well as those to robotics [8,33]
   - Knowledge extraction from internet documents. This branch of research, studied in [35,36], can be combined with the automatic creation of knowledge bases branch in order to produce domain-specific knowledge bases or concept classifications from web documents.

   Of particular interest for 3) is our recent research on code migration [26,23], [28,29], higher level internet tools [25,24], and resource discovery [35,36].
4. Automatic creation of taxonomies: This branch of our research is based on linguistic work [4] and has application to the two previous objectives as well as for instance to molecular biology, medical and forestry applications, etc. Our methodologies for type hierarchies will also be useful re. line of research 1), since an underlying ontology must be gleaned from the user's natural language specifications.

3 Related Work

The intersection between logic programming and the internet is a very new but rapidly growing field. Recent logic programming conferences typically include workshops or tutorials on the subject, and the journal Theory and Practice of Logic Programming has recently put out a special issue on this theme. A useful classification in terms of client-based systems, server-side systems, and peer-to-peer systems is given in [20]. Depending on the site where (most of) the processing happens, most systems fall into either client-side or server-side

systems, while peer-to-peer systems (such as our own [11]) tend to have fully symmetric interaction capabilities, and use abstractions such as message passing or blackboards, while retaining the Internet as their underlying communication layer. This allows them to implement multi-agent systems, where all participants must communicate on equal terms, bypassing the intrinsic asymmetry of the client/server model. The most natural incarnation of peer-to-peer systems is the metaphor of communicating Virtual Worlds. The only system we know of which uses logic programming for virtual world simulation is our own system LogiMOO [32,11,31], although many sophisticated web-based applications and tools have been implemented in CP/CLP languages. A very large number of research projects have recently started on mobile computations and mobile agent programming. Among the most promising developments are Luca Cardelli's Oblique project at Digital, mobile agent applications, and IBM Japan's aglets (`http://www.trl.ibm.co.jp/aglets`). Database interfacing through spoken language has been little explored, possibly because speech analysis and synthesis software is relatively new and not as advanced as it should be for truly practical uses. Written text, however, has long been used for database consultation [3] and for database updates (e.g. [16]). There is increasing interest from industry in the spoken language field. However, putting all the pieces of the puzzle together will require careful crafting. Within the logic-based database field, developments such as the uses of Inductive Logic Programming to automate the construction of natural language interfaces to database queries [37] could prove most valuable.

4 Expected Benefits

Providing more human like communication with computers and with the Internet might help bridge the gap between the humanistic and the formal sciences, towards an overall more balanced world. Linguistics, being the most formalized of the humanistic sciences, holds fascinating promise when interacting with Computing Sciences.

Speech-driven database creation and consultation and robot control or programming might give some relief from computer use related health problems (tendonitis; eye, neck and back strain, Carpal Tunnel Syndrome...) that the present typing/screen based model of computer use entails. Our proposed higher level tools for internet access and interaction will add a degree of intelligent communication to the web, that fantastic but frustratingly unimaginative repository of world knowledge; and our multilingual virtual worlds will hopefully remove geographic and language barriers, perhaps contributing to enhance understanding and cooperation among the people of this world.

Acknowledgements

Thanks are due to my collaborators in the various projects here described: Pablo Accuosto, Jamie Andrews, Joao Balsa, Koen De Boschere, Andrew Fall, Jia Wei

Han, Yan Nong Huang, Renwei Li, Luis Moniz Pereira, Lidia Moreno, Manuel Palomar, Gabriel Pereira Lopes, Stephen Rochefort, Andrea Schiel, Marius Scurtescu, Paul Tarau, Marie-Claude Thomas, Kimberly Voll, Tom Yeh, and Osmar Zaiane. Special thanks go to Paul Graunke for formatting the text. This research was made possible by NSERC research grant 611024 and NSERC Equipment grant 31-613183.

References

1. J. Andrews, V. Dahl, and P. Tarau. Continuation logic programming: Theory and practice. In *ILPS'95 Workshop on Operational and Denotational Semantics of Logic Programs*, November 1995.
2. J. Balsa, V. Dahl, and J. G. Pereira Lopes. Datalog grammars for abductive syntactic error diagnosis and repair. In *Natural Language Understanding and Logic Programming Workshop*, 1995.
3. V. Dahl. On database systems development through logic. In *ACM Transactions on Database Systems*, volume 7(1), pages 102–123, March 1982.
4. V. Dahl. Incomplete types for logic databases. In *Applied Mathematics Letters*, volume 4(3), pages 25–28, 1991.
5. V. Dahl. From speech to knowledge. In M. T.Pazienza, editor, *Information Extraction: towards scalable, adaptable systems*, volume 1714, pages 49–75. Springer-Verlag, 1999. LNAI (Lecture Notes in Artificial Intelligence).
6. V. Dahl, A. Fall, S. Rochefort, and P. Tarau. Hypothetical reasoning framework for natural language processing. In *8th IEEE International Conference on Tools with Artificial Intelligence*, November 1996.
7. V. Dahl, A. Fall, and P. Tarau. Resolving co-specification in contexts. In *IJCAI'95 Workshop on Context in Language*, July 1995.
8. V. Dahl, A. Fall, and M. C. Thomas. Driving robots through natural language. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 1904–1908, 1995.
9. V. Dahl and P. Tarau. From assumptions to meaning. In *Canadian Artificial Intelligence*, volume 42, Spring 1998.
10. V. Dahl, P. Tarau, P. Accuosto, S. Rochefort, and M. Scurtescu. Assumption grammars for knowledge-based systems. In *Informatica, Special Issue on Natural Language Processing and Agent Systems*, volume 22(4), pages 435–444, December 1998. (previous version in: Proc. NLDB'97, Vancouver, June 1997).
11. V. Dahl, P. Tarau, P. Accuosto, S. Rochefort, and M. Scurtescu. A spanish interface to LogiMOO—towards multilingual virtual worlds. In *Informatica*, volume 2, June 1999. (previous version in: Proc. International Workshop on Spanish Natural Language Processing and Language Technologies, Santa Fe, New Mexico, July 1997).
12. V. Dahl, P. Tarau, and J. Andrews. Extending datalog grammars. In *Workshop on Natural Language and Databases (NLDB'95)*, June 1995.
13. V. Dahl, P. Tarau, and Y. N. Huang. Datalog grammars. In *Joint Conference on Declarative Programming*, pages 19–22, September 1994.
14. V. Dahl, P. Tarau, and R. Li. Assumption grammars for natural language processing. In Lee Naish, editor, *Fourteenth International Conference on Logic Programming*, pages 256–270. MIT Press, 1997.

15. V. Dahl, P. Tarau, L. Moreno, and M. Palomar. Treating coordination through datalog grammars. In *COMPULOGNET/ELSNET/EAGLES Workshop on Computational Logic for Natural Language Processing*, pages 1–17, April 1995.
16. James Davison. A natural language interface for performing database updates. In *ICDE*, pages 69–76, 1984.
17. Y. N. Huang, V. Dahl, and J. Han. Rule updates in logic databases: A meta programming approach. In *3rd International Pacific Rim Conference on Artificial Intelligence,*, August 1994.
18. Y. N. Huang, V. Dahl, and J. W. Han. Fact updates in logic databases. In *Int. Journal of Software Engineering and Knowledge Engineering*, volume 5(3), pages 467–491, 1995.
19. R. Li, V. Dahl, L. Moniz Pereira, and M. Scurtescu. Dealing with exceptions in textual databases. In *NLDB*, June 1997.
20. S. W. Loke. *Adding Logic Programming Behaviour to the World Wide Web.* PhD thesis, University of Melbourne, 1998.
21. S. Rochefort, V. Dahl, and P. Tarau. Controlling virtual worlds through extensible natural language. In *AAAI Symposium Series "Natural Language Processing for the World Wide Web"*, March 1997.
22. S. Rochefort, V. Dahl, and P. Tarau. A virtual environment for collaborative learning. In *World Multiconference on Systemics, Cybernetics and Informatics (SCI'98) and 4th International Conference on Information Systems Analysis and Synthesis (ISAS'98)*, volume 4, pages 413–416, 1998.
23. P. Tarau and V. Dahl. Code migration with first order continuations. In *Joint Declarative Programming Conference AGP98,*, July 1998.
24. P. Tarau and V. Dahl. A coordination logic for agent programming in virtual worlds. In W. Conen and G. Neumann, editors, *Coordination Technology for Collaborative Applications - Organizations, Processes, and Agents.* Springer-Verlag, 1998.
25. P. Tarau and V. Dahl. A logic programming infrastructure for internet programming. In M. J. Wooldridge and M. Veloso, editors, *Artificial Intelligence Today—Recent Trends and Developments*, pages 431–456. Springer-Verlag, 1999. LNAI 1600.
26. P. Tarau and V. Dahl. High level networking with mobile code and first order and continuations. In *Theory and Practice of Logic Programming.* Cambridge University Press, March 2001. (This is the new and sole official journal of the Association of Logic Programming).
27. P. Tarau, V. Dahl, and K. De Boschere. A logic programming approach to coordination in virtual worlds. In *Workshop on Coordination languages, models, systems in the Software Technology Track of the Hawaii International Conference on System Sciences (HICSS-31)*, 1997.
28. P. Tarau, V. Dahl, and K. De Boschere. A logic programming infrastructure for remote execution, mobile code and agents. In *Post ICLP Workshop on Logic Programming and Multi Agents*, July 1997.
29. P. Tarau, V. Dahl, and K. De Bosschere. Remote execution, mobile code and agents in binprolog. In *Electronic Proc. Logic Programming Workshop in conjunction with the 6th International World Wide Web Conference*, pages 7–11, April 1997.
30. P. Tarau, V. Dahl, and A. Fall. Backtrackable state with linear affine implication and assumption grammars. In J. Jaffar and R. Yap, editors, *Concurrency and parallelism, Programming, Networking, and Security*, pages 53–64. Springer Verlag, 1996. Lecture Notes in Computer Science 1179.