ParaSite: Mining the Structural Information on the World-Wide Web

by

Ellen Spertus

S.B., Computer Science and Engineering, MIT (1990)
S.M., Electrical Engineering and Computer Science, MIT (1992)

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1998

Copyright ©Ellen Spertus, 1998. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part.

Author .................................................................
Department of Electrical Engineering and Computer Science
February 6, 1998

Certified by .............................................................
Lynn Andrea Stein
Class Of 1957 Career Development Associate Professor
Thesis Supervisor

Accepted by ............................................................
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students
ParaSite: Mining the Structural Information on the World-Wide Web
by
Ellen Spertus

Submitted to the Department of Electrical Engineering and Computer Science
on February 6, 1998, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

The World-Wide Web is potentially the world’s largest knowledge base but only if new information retrieval techniques are developed to take advantage of its unique characteristics, particularly the semi-structured information within pages, across pages, and in page names. Because these types of structure are represented in such different ways, a large number of specialized tools have been required to gather structural information. I provide a relational database interface to the Web called Squeal, which encapsulates these different types of structure in a uniform manner, allowing the user to query the Web in Structured Query Language (SQL) as if it were a database. A novel “just-in-time” interpreter automatically retrieves information from the Web as implicitly demanded by user queries, a technique which could be applied not just to the Internet but to other sources of data too large to be precomputed into a database. The level of abstraction provided by Squeal allows the user to easily create agents that make full use of the previously-untapped information on the Web. One such “ParaSite” is a simple structure-based recommender system that compares favorably to the best text-based system.

Thesis Supervisor: Lynn Andrea Stein
Title: Class Of 1957 Career Development Associate Professor
Acknowledgments

The University of Washington (UW) was my academic home away from home for the last two years. I am grateful to Oren Etzioni and Dan Weld for taking me into their Internet Softbot group. Other UW folk who made me feel at home were Alan Borning, Lauren Bricker, Steve Hanks, Frankye Jones, David (Pardo) Keppel, Ed Lazowska, Sean Sandys, my office-mates Marc Finucynski, Jack Lo, Brendan Muney, Kurt Partridge, and Xiaohan Qin, and of course Keith Golden.

My work benefited greatly from interaction at UW with Oren Etzioni, Marc Friedman, Keith Golden, Nick Kushmerick, Tessa Lau, Marc Langheinrich, Greg Lauckhart, Alon Levy, Neal Lesh, Greg Linden, Kurt Partridge, Mike Perkowitz, Rich Segal, Erik Selberg, Jonathan Shakes, and Stephen Soderland of the University of Washington. I’m particularly grateful to Rich for badgering me into realizing that the user need not explicitly request transfers from the Web to the SQL database.

Alberto Mendelzon, Gustavo Arocena, and George Mihaila of the University of Toronto have generously shared their time, code, and expertise. Their WebSQL system was a major influence on this work.

I also benefitted from code made freely available by Arthur Do (HtmlStreamTokenizer [8]), Santeri Paavolainen (MD5 [30]), Original Reusable Objects, Inc. (OROMatcher [29]), and Sriram Sankar, Sreenivasa Viswanadha, and Rob Duncan (JavaCC [36]).

This thesis is the culmination of many years as an MIT student. Faculty and staff who were especially helpful and encouraging over the years include Bill Dally, who introduced me to research and supervised my bachelor’s and master’s theses; John Guttag, for repeatedly helping me beyond the call of duty; Tom Knight, who always encouraged me with even my most random ideas; Marilyn Pierce, for her helpfulness and warmth; Jerry Sussman, who is MIT incarnate; and Bill Weihl, whom I could always go to for excellent advice. Other people who encouraged me over the years were Judy Goldsmith of the University of Kentucky and David Lewis of AT&T Research.

From the beginning of freshman year through the completion of three theses, Nate Osgood has provided me with invaluable emotional and intellectual support. I am privileged to have worked with such a wonderful person. I always enjoyed staying with him, Carol Collura and Norm Margolus, Kathy Knobe, my sister Andrea Spertus, and Lisa and Greg Tucker-Kellogg on my visits to MIT.

Philip Greenspun provided me with a much-needed education about databases. Other MIT folk with whom I had valuable discussions about my thesis include Michael Ernst (now at UW), David Karger, Brian LaMacchia (now at Microsoft), Richard Lethin (now at Equator Technologies Consulting), Henry Lieberman and Jim Miller.

I was generously funded by the National Science Foundation for three years of my graduate study and by the Intel Foundation for one particularly crucial year.

I am very grateful to my committee: Lynn Andrea Stein, Ken Haase, Tom Knight, and Pattie Maes. I was privileged to have such a strong and diverse committee, with Lynn’s strength in AI, Ken’s and Tom’s broad backgrounds, Pattie’s expertise with agents and collaborative filtering, and all of their interests in novel approaches to information retrieval. I also appreciate help from their assistants: Marie Lamb, Agnieszka Meyro, and Annika Pfluger.

I could not have done this without the support of my family. I am particularly grateful to my father and brother for stimulating my interest in computers and math.

I am not sure I could have done this without the tremendous support and assistance from my advisor Lynn Stein and my fiancé Keith Golden. Lynn took me on as a student under non-optimal circumstances and stuck with me despite the geographic obstacles and the many other demands on her time. She challenged me at just the right level, treating my ideas and work with respect but demanding that I then go and fully develop them. It is a rare teacher who manages to be both rigorous and compassionate. These meant even more to me than her vital intellectual contribution.

Keith Golden helped me in more ways than I could list, including discussing research with me, teaching me about AI, and showing me by example that a thesis could be completed gracefully and thoroughly, but most of all by loving me and believing in me. I feel that I’ll be able to do anything I want with him at my side and have a lot of fun too.
# Contents

1 Introduction ........................................ 11

1.1 ParaSites ........................................ 12

1.1.1 Link geometry .................................. 12

1.1.2 A recommender system ........................... 12

1.1.3 A home page finder .............................. 13

1.1.4 Summary ........................................ 13

1.2 A Database Interface to the Web .................... 14

1.2.1 Background ..................................... 14

1.2.2 Squeal .......................................... 15

1.3 Related work ....................................... 16

1.3.1 Semantic networks ............................... 16

1.3.2 Structure within documents ........................ 16

1.3.3 Structure within and across web pages ............ 16

1.4 Reader’s Guide ..................................... 17

2 Ontology ............................................. 18

2.1 Types of Information on the Web .................... 18

2.1.1 Uniform Resource Locators (URLs) ............... 18

2.1.2 Hypertext Markup Language ........................ 19

2.2 Database Relations ................................ 20

2.2.1 Basic Relations .................................. 20

2.2.2 Tag and attribute relations ....................... 22

2.2.3 Relations built on tags ........................... 24

2.2.4 Relations for second-hand information ............ 27

2.3 Summary .......................................... 30

3 Squeal ............................................. 32

3.1 Lexical Tokens .................................... 32

3.2 Expressions ....................................... 32

3.2.1 First-class data types ............................ 33

3.2.2 Passed-through data types ........................ 33

3.2.3 Special data types ................................ 33

3.3 Simple Statements ................................ 33

3.3.1 Basic statements ................................ 33

3.3.2 Function/procedure statements .................. 36

3.3.3 Control statements ............................... 36

3.4 Internal Statements: FETCH and MSELECT .......... 38

3.5 Squeal’s SQL core ................................... 41

3.5.1 User-Readable Tables ............................. 41

3.5.2 Automatic Tables ................................ 42

3.5.3 Derived Tables ................................... 42

3.6 Command-Line Interface ............................. 47
4 Implementation

4.1 Database state
4.2 Column
4.3 Tables
  4.3.1 Purpose
  4.3.2 Details
  4.3.3 Squeal internal tables
4.4 Exceptions
4.5 Representation of variables
  4.5.1 SymbolTable
  4.5.2 Bindings
4.6 Parser
  4.6.1 SimpleNode
  4.6.2 NodeWithRequiredName
  4.6.3 NodeWithOptionalName
  4.6.4 NodeContainingList
  4.6.5 NodeContainingParenthesizedList
  4.6.6 BinaryOperation
  4.6.7 Context
4.7 Output
4.8 FrontEnd
4.9 Miscellaneous
  4.9.1 namedArg
  4.9.2 Cell
  4.9.3 Junction
  4.9.4 Set
  4.9.5 SelectionResult
4.10 Functions and Procedures
  4.10.1 UserCallableFunc
  4.10.2 UserDefinedFunc
  4.10.3 UserDefinedProc
  4.10.4 JavaDefinedFunc
  4.10.5 SQLfunc
4.11 SearchEngine
  4.11.1 AltaVista
  4.11.2 Lycos
4.12 Computation
4.13 Selection
4.14 Utils
  4.14.1 String Manipulation
  4.14.2 SQL Server Access
  4.14.3 Conversion
  4.14.4 Node Manipulation

5 Applications

5.1 Sibs: Finding Similar Pages
  5.1.1 Basic Technique
  5.1.2 Optimizations
  5.1.3 Evaluation
5.2 A Home Page Finder
  5.2.1 Sample Run
  5.2.2 Support for nicknames
  5.2.3 Evaluation
5.3 Bo Peep: Finding Moved Pages
5.3.1 Technique 1: Climbing the directory hierarchy
5.3.2 Technique 2: Checking with pages that referenced the old URL

6 Conclusions

6.1 Lessons Learned
6.1.1 A Relational Database Model of the Web
6.1.2 Using SQL syntax to specify computation

6.2 Comparisons to Related Work
6.2.1 Structure within and across web pages
6.2.2 Theoretical analyses of the Web
6.2.3 Database interfaces to the Web

6.3 Future Work
6.3.1 Improvements to the System
6.3.2 Further evaluation

A SQL Database Schema for Squeal

B The Squeal Grammar

C Source Code for Home Page Finder

D User Evaluations of Recommender Systems
D.1 American Airlines
D.2 Geodesic Systems
D.3 Rogue Market
D.4 Art Bell
D.5 Activision
D.6 Happy Puppy
D.7 Economist
# List of Figures

1-1  A geometric representation of material related to computer science and Iowa  

1-2  Cache relation of the database to the Web 

1-3  Structure of Data Transfer in the ParaSite System 

2-1  HTML specification of internal document structure 

2-2  Appearance of HTML internal document structure 

2-3  Transcript illustrating the basic relations 

2-4  Transcript demonstrating the tag and att relations 

2-5  Rules for managing list structure 

2-6  Example of struct values of list at list tags 

2-7  A SQL definition of link in terms of other relations 

2-8  Transcript demonstrating the link relation 

2-9  Transcript demonstrating the rlink and rcontains relations 

3-1  Lexical tokens 

3-2  Grammar for expressions 

3-3  Grammar for statement 

3-4  Grammar for LET statements 

3-5  Grammar for DFUPCN, DFPROC, CALL, and HELP 

3-6  Transcript demonstrating Squeal functions and procedures 

3-7  Grammar for INPUT, OUTPUT, and QUIT Statements 

3-8  Transformation of Squeal user query into internal statements 

3-9  Grammar for FETCH statement 

3-10 Interpretation of simple FETCH statements 

3-11 Grammar for squeal queries 

3-12 Pseudocode for transform 

3-13 Pseudocode for findBound 

3-14 Description of merge 

3-15 Pseudocode for refDependences 

3-16 Changed portion of findBound for derived tables 

3-17 Usage for Squeal 

4-1 Member variables for Column 

4-2 Class hierarchy of tables 

4-3 Member variables for Table 

4-4 Methods defined for Table 

4-5 Methods defined for SymbolTable 

4-6 Methods defined for Bindings 

4-7 Class hierarchy of parser-generated nodes 

4-8 Methods defined for SimpleNode 

4-9 Class hierarchy based on java.io.PrintWriter 

4-10 Sample log output 

4-11 Methods in FrontEnd
4-12 Member variables for SelectionResult ................................. 59
4-13 Class hierarchy for functions and procedures ........................... 60
4-14 Methods defined for UserCallableFunc ................................... 60
4-15 Java-defined function strcat ........................................... 61
4-16 Code for SQLfuncRlink .................................................. 62
4-17 Methods defined for SearchEngine ....................................... 63
4-18 String-manipulation methods defined for Utils ......................... 66
4-19 SQL server access methods defined for Utils .......................... 66
4-20 Conversion methods defined for Utils .................................. 67
4-21 Node manipulation methods defined for Utils ......................... 68

5-1 Code for SimPagesBasic .................................................. 70
5-2 Transcript of run of SimPagesBasic ...................................... 71
5-3 Modification to SimPagesBasic to require presence of keyword .......... 71
5-4 Follow-On to SimPagesBasic to return hosts ............................ 71
5-5 Follow-On to SimPagesBasic to only return pages pointing to one or more of the original pages ................................................. 72
5-6 Code for SimPageListHeader ............................................. 73
5-7 Listings returned by Excite query on “Roger Ebert” ....................... 74
5-8 Listings returned by Excite “more like this” query ....................... 75
5-9 Code for Sibs .............................................................. 75
5-10 Instructions for evaluation of recommender systems ...................... 77
5-11 Sample evaluation page for recommender systems ....................... 79
5-12 Top level code for home page finder ................................... 87
5-13 The “names” table ....................................................... 90
5-14 Top level code for HomePageWithNicknames ............................. 90
5-15 Blurb returned from HotBot in response to the query “Lenore Blum 1943” 91
5-16 Bo Peep: Code to climb one up in the directory hierarchy .............. 92
5-17 Views of parse and valstring for a parent directory and child file .... 92
5-18 Simple Implementation of Bo Peep .................................... 93
5-19 Transcript of Bo Peep .................................................... 94
5-20 Example of broken Link .................................................. 94
5-21 Implementation of Bo Peep2 ............................................. 95
5-22 Transcript for Bo Peep2 .................................................. 96
## List of Tables

1.1 Sample relation `link` representing hyperlinks ........................................ 14

2.1 The `valstring` relation ........................................................................ 21
2.2 The `urls` relation .................................................................................. 22
2.3 The `parse` relation ................................................................................ 22
2.4 Example `parse` relation ......................................................................... 22
2.5 The `page` relation .................................................................................. 24
2.6 The `tag` relation .................................................................................... 24
2.7 The `att` relation ..................................................................................... 24
2.8 The `header` relation ............................................................................... 26
2.9 The `list` relation .................................................................................... 26
2.10 The `link` relation .................................................................................. 27
2.11 The `rcontains` relation ......................................................................... 30
2.12 The `rlink` relation ................................................................................ 30

3.1 User-callable functions defined at Squeal start-up .................................. 36
3.2 Defining columns for tables ..................................................................... 38
3.3 Relations allowing `FETCH` conjunctions or disjunctions ....................... 39
3.4 SQL commands supported by Squeal ....................................................... 41
3.5 Categories of tables ............................................................................... 42
3.6 Relations between derived tables and their parents ................................ 43

4.1 The `creation` relation ............................................................................ 50
4.2 The `computation` relation ..................................................................... 51
4.3 Sample `computation` entry ..................................................................... 52
4.4 Classes of nodes created by parser ......................................................... 55
4.5 Streams used by Squeal .......................................................................... 57

5.1 Top 5 pages returned by Excite and ParaSite with Ebert seed URL .......... 74
5.2 Performance of Excite and ParaSite on 25 seed URLs ............................. 76
5.3 Averages of ratings by seed URL ............................................................. 78
5.4 User ratings of Ebert recommendations ................................................... 79
5.5 Top 5 pages returned by Excite and ParaSite with Austin weather seed URL 80
5.6 User ratings of Austin weather recommendations .................................... 80
5.7 Top 4 pages returned by Excite and ParaSite for AMD seed URL .......... 81
5.8 User ratings of AMD recommendations ................................................... 81
5.9 Top 4 pages returned by Excite and ParaSite for GSotD seed URL .......... 82
5.10 User ratings of Geek Site of the Day recommendations .......................... 82
5.11 Top 5 pages returned by Excite and ParaSite for MapQuest seed URL .... 83
5.12 User ratings of MapQuest recommendations ........................................... 83
5.13 Top 5 pages returned by Excite and ParaSite for GSotD seed URL .......... 84
5.14 User ratings of KnotPlot recommendations ............................................ 84
5.15 Pages returned by ParaSite for KnotPlot with `tolinks=40` ..................... 85
5.16 Pages returned by ParaSite for MapQuest with `tolinks=40` ................. 85
5.17 Results of Home Page Finder on names from Aha’s list

D.1 Averages of ratings by seed URL with page numbers

D.2 Top 5 pages returned by Excite and ParaSite for American Airlines seed URL

D.3 User ratings of American Airlines recommendations

D.4 Top 4 pages returned by Excite and ParaSite for Geodesic Systems seed URL

D.5 User ratings of Geodesic Systems recommendations

D.6 Top 5 pages returned by Excite and ParaSite for Rogue Market seed URL

D.7 User ratings of Rogue Market recommendations

D.8 Top 5 pages returned by Excite and ParaSite for Art Bell seed URL

D.9 User ratings of Art Bell recommendations

D.10 Top 5 pages returned by Excite and ParaSite for Activision seed URL

D.11 User ratings of Activision recommendations

D.12 Top 4 pages returned by Excite and ParaSite for Happy Puppy seed URL

D.13 User ratings of HappyPuppy recommendations

D.14 Top 5 pages returned by Excite and ParaSite for Economist seed URL

D.15 User ratings of Economist recommendations