计算机科学1997 Vol.21 No.6

World Wide Web的索引与查询技术

The Index and Query Techniques of World Wide Web

阳小华　周龙骧

（中国科学院数学研究所　北京100080）　（中南工学院计算机系）

摘　要　With the explosive growth of World Wide Web, one of the most pressing issues is the so-called resource discovery problem. It leads to the development of information systems which index the Web documents and allow users to locate resources by specifying keywords. In this paper, we discuss the index and query techniques used in current WWW information systems and the future work of WWW search.

关键词　World Wide Web, information retrieve, index database

1　引言

WWW(World Wide Web)是一个由许多称为Web页的超媒体文档组成的集合，这些文档用HTML(Hyper Text Markup Language)书写，包含多种媒体对象和指向其它文档的指针(超级链接)。Web文档散布在世界各地的Web服务器上，每个服务器自主地管理自己的资源，没有统一的管理机制。由于WWW的迅速发展，其信息容量已超过Gopher和WAIS，成为全球最大的信息系统，因而要在

束、消息发送等，也是由TLA公式来表示的，最终性的证明一般都归约成 $P \leadsto Q$。因为并发系统的安全属性蕴含任何事情会发生，因此 $P \leadsto Q$ 的证明必须从程序的活性条件中导出。我们使用规则WF1来证明。只要做一些简单的替换就可以了：

$P \leftarrow n \in Nat \wedge x = n$　$N \leftarrow M$　$f \leftarrow \langle x, y \rangle$

$Q \leftarrow x = n+1$　$A \leftarrow M_1$

则该规则的假设变成

$(n \in Nat \wedge x = n) \wedge [M]_{\langle x,y \rangle} \Rightarrow ((n \in Nat \wedge x' = n) \vee (x' = n+1))$

$(n \in Nat \wedge x = n) \wedge \langle M_1 \rangle_{\langle x,y \rangle} \Rightarrow (x' = n+1)$

$(n \in Nat \wedge x = n) \Rightarrow Enabled\langle M_1 \rangle_{\langle x,y \rangle}$

所以规则的结论很容易得到

$\Box[M]_{\langle x,y \rangle} \wedge WF_{\langle x,y \rangle}(M_1) \Rightarrow ((n \in Nat \wedge x = n) \leadsto (x = n+1))$

在TLA中，一个并发程序和它的属性之间没有任何区别，对于 Φ 来说，我们与其把它看成是一个并发系统的描述，不如把它看成是一个要求程序所必须满足的性质。

3.4　用TLA描述低级程序实现高级程序

一个系统可以在许多抽象层次上进行描述，从高级的描述到低级的实现。如果S1允许的每一个外部可见的操作S2也允许，则我们说规范S1实现规范S2。为了证明S1实现S2，我们要证如果S1允许的操作序列$\langle (e0,z0),(e1,z1),(e2,z2),\cdots \rangle$，其中zi为内部状态，那么必然存在S2允许的内部状态 $yi \Leftrightarrow \langle (e0,y0),(e1,y1),(e2,y2),\cdots \rangle$。

只要找到一个函数f使$(ei,zi)=f(ei,yi)$，证明f映射S1的执行序列(可能有停滞不前步)到S2的执行序列，同时还保证了S2的活性属性。一个映射f是保序列和保活性属性的叫做一个精炼映射。

限于篇幅，本文不再讨论用TLA的公式和演泽规则来证明了。有关这方面的详细的内容，可参考本文的有关文献。

参考文献

[1] Leslie Lamport, The Temporal Logic of Actions, ACM Trans. on Programming Language and Systems, 16(3), 1994

[2] Martin Abadi and Leslie Lamport, The Existence of Refinement Mappings, Theoretical Computer Science 82, 1991

[3] 王元元，计算机科学中的逻辑学，1989

Web 上找寻所需的信息不是一件容易的事情。

为了帮助用户查找 Web 信息，出现了许多 WWW 索引信息系统，如 WebCrawler[1]、Yahoo[2]、AltaVista[3]、Excite[4]、Infoseek[5]、Lycos[6]、HotBot[7]、AliWeb[8]、Harvest[9]等。这些系统通过各种方法搜集散布在各个 Web 服务器上的信息，为文档建立索引，给用户查找信息提供有效的手段。

虽然许多 WWW 索引信息系统不仅支持对 Web 信息的索引与查询，还常常提供对 Internet 上其它资源(如 Newsgroups、Gopher、FTP 等)的索引与查询服务，但是为了简单清晰，本文只讨论在 WWW 上使用的索引和查询技术。

2 Web 索引

2.1 索引信息的获取方法

索引信息系统如何发现 Web 文档并为之建立索引呢?有两种方法，一是由用户主动向系统报告自己的资源，一是由系统的搜索工具去找寻用户的信息。许多系统都有 Add URL 功能，用户可以利用这个功能递交自己的文档信息。Yahoo 要求四个方面的信息：title、URL、category 和 comment。其中 title 是文档在 Yahoo 中的名字，category 是文档在 Yahoo 中的类别，也就是它所在的子目录，comment 是文档内容的概述(不超过20个单词)。事实上，Yahoo 只是把用户提供的信息加以整理，建立索引，它并不去访问 URL 所指向的文档。当然，Yahoo 也通过搜索去发现新的资源。

WebCrawler、AltaVista、Excite、Lycos 等系统只把 Add URL 功能作为给其 Web 自动搜索工具(Web robot)提供出发点的一个手段，另一个获取初始 URL 的手段是访问诸如 NCSA 的 What's New 等 Internet 信息发布场所。Web robot，也称作 crawler、spider、wanderer、worm 等，是一个能够沿着超链漫游 Web 文档集合的程序。给定一些 URL，Web robot 能够利用象 HTTP 这样的标准协议读取相应的文档，然后沿着文档中的超链访问新的文档，如此继续下去。因此，用户只要提供一个 URL，系统就能够追溯所有从该 URL 直接或间接可达的文档，并为之建立索引。由于 Web robot 能够快速地、周期性地浏览 WWW，获取最新的信息，所以基于 Web Robot 的索引信息系统可以建立并维持较大的索引信息库。AltaVista 的 Web 索引信息库有约3100万个 Web 页，Excite 的 Web 索引库有约5000万个 Web 页，Lycos 有近7000万个 URL 索引。相比之下，Yahoo 只有几十万个索引项，要小得多。所以，各种各样的 robot 在 WWW 索引信息系统中得到了广泛的使用[10]。

Robot 频繁地访问各个 Web 结点，索取大量的信息，给网络和被访问的 Web 服务器带来了较大的负担，如果 Web 结点管理员不喜欢某些 robot 的打扰，可以按照 SRE (Standard for Robot Exclusion)[11]提供的方法限制它们对结点资源的访问。Robot 的另一个缺陷是不加区分地索取所有可达的文档，为之建立索引。这样，不仅使索引数据库变得十分臃肿，而且丧失了文档之间的结构。有些 Web 文档是为了特定的目的动态生成的，生存期很短，为它们建立索引是毫无意义的。例如，每个索引信息系统都可以接受用户的查询，查询结果就是以 Web 文档的形式返回的，这些文档只在查询期内有意义。另外，WWW 上有许多服务，这些服务的表示形式就是 Web 文档的集合。对于用户来讲，只要知道服务的入口就可以享受整个服务。为组成服务的所有文档建立索引，不仅没有必要，而且还破坏了原有的结构。

为了减轻 robot 给网络和 Web 服务器带来的负担，AliWeb[12]采用了一个独特而有效的方法构造其索引信息库。每个希望自己的信息被 AliWeb 索引的 Web 服务器管理员要按照 IAFA (Internet Anonymous FTP Archives)模式为本结点的文档建立一个索引文件，一般命名为 site.idx，直接放在本结点公用的 HTML 目录下。Web 管理员要访问 AliWeb，填写表格，为本结点注册。AliWeb 周期性地访问注册结点，读取索引文件，汇集索引信息，构成自己的索引信息库。每个结点的索引文件由其管理员负责维护，以保证索引信息的有效性。AliWeb 的优点是简单有效，给网络和 Web 服务器带来的负担较轻。但是，由于需要 Web 服务器管理员的主动参与，其覆盖面较窄。

与 WebCrawler、AltaVista、Excite、Lycos 等系统维持集中而庞大的索引信息库不同，Harvest[13]采用了分布式的方法。Harvest 系统由多个子系统组成，收集信息的工作由 Gatherer 子系统承担。Gatherer 子系统由许多 Gatherer 构成，它们可以分布在各个 Web 结点上，周期性地扫描所在结点的 Web

文档，收集最新的索引信息，这些索引信息被称为Broker的程序汇总组成综合性的索引库。Harvest的Broker子系统包含许多Broker，每个Broker可以从一个或多个Gatherer那里收集索引信息，还可以从其它的Broker那里索取信息。Broker可以按Internet域或地理范畴分布，各司其职，协同工作，这样可以避免索引信息的重复收集。Gatherer和Broker的管理由称作Harvest Server Registry的Broker负责。考虑到WWW的庞大性，Harvest这种按Internet域或地理范畴建立分布式索引信息系统的想法是很有吸引力的。但是，由于WWW结点的自治性，实现起来却很困难。

2.2 索引信息的选取与组织

有些WWW索引信息系统，如Yahoo、AliWeb等，只是简单地汇集用户按一定格式提供的关于Web文档的信息，它并不去访问文档本身。但是，绝大多数基于robot的索引信息系统都要读取Web文档，分析其结构，提取索引信息。由于HTML文本本身有一定的结构，这些结构把文档分割成不同的组成部分，而各个部分的重要性是不一样的。因此，有些系统只从文档中提取比较重要的内容建立索引。WWWW(World Wide Web Worm)[14]的索引就只包括文档的标题(title)、地址(URL)和超级链接(包括其中的文本与URL)。

Robot跨越网络读取文档，却只使用了其中的一部分信息，大部分内容被抛弃，这无疑是一种浪费。另外，由于Web文档千差万别，几个特定的部分未必能完全概括其内容。因此，象AltaVista、Excite、Infoseek、Lycos、HotBot等索引信息系统都采用全文索引，其索引范围覆盖了文档除comment域以外的几乎全部内容。AltaVista、Infoseek等系统的索引甚至包括了虚词。在Infoseek中查询词组"to be or not to be"，有374个文档被选中。在Excite、Lycos、HotBot等系统中查询"to be or not to be"则得不到有意义的结果，因为它们的索引排除了诸如"the"、"or"、"a"、"to"等没有具体意义的虚词。

为了更好地索引Web文档，AltaVista、Infoseek等系统获取并保持了HTML文本本身的结构信息，如title、anchor、link等。同时，还允许用户使用元标识(META Tag)"keywords"为文档提供附加的索引信息，用元标识"description"为文档建立摘要。由于元标识可以帮助系统为文档建立更好的索引与摘要，但也可能被某些人用来误导系统，因此，许多系统对它持保留态度。

为了方便用户找寻信息，Yahoo把收集到的索引信息按主题分类组成一个层次型的目录系统。每条索引都属于某些子目录，也就是类别(category)。Yahoo允许用户在Add URL时选择索引所属的类别。Infoseek，号称为Web最大的目录系统，其超过5000万个的索引项也被纳入多层目录结构。与Yahoo不同的是，索引项在Infoseek目录系统中的位置是由其编辑根据文档的内容来决定的。WebCrawler也提供了称为WebCrawler Select的目录，以便用户浏览。

Excite把索引信息组织为14个频道(channel)，每个频道集成了Web Guide、News、Guided Web Tours、Excite Board&Chat、Exciting Stuff、Other Destination和Search the Web等栏目。Web Guide是一个子目录系统，分门别类地列出本频道的Web结点，Guided Web Tours引导用户浏览与本频道某些主题关系最密切的结点。Exciting Stuff为用户提供本频道最有意思的信息。Other Destination为用户访问本频道的相关信息提供了快捷手段。

Lycos的核心是18个面向主题的Internet活动中心，Lycos WebGuide。每个Lycos WebGuide包括News、Features、Lycos MiniGuide、Lycos Top 5% Sites、Top Ten Sites、Lycos Power Searches、Web Searches、Custom Searches、Pictures&Sounds Searches等栏目，其内容由Lycos Editorial Team不断刷新(每天数次)，以保证向用户提供有关本主题的最新和最有用的信息。

象Lycos WebGuide和Excite channel这样，把目录结构与新闻、特写、导游等内容结合起来，可以适应各种用户的不同需要，给用户提供尽可能多的帮助，使他们能够更方便地找寻WWW信息。但是，过多的栏目和服务也给普通用户掌握系统的使用带来了麻烦，同时也增加了系统的维护工作。事实上，简单清晰的目录系统是广大用户最欢迎的。

在WWW上有成千上万的Web文档，它们可以分为许多类型。象Yahoo、Infoseek这样的目录系统仅仅是按主题进行分类，而且是靠人工来维持的。事实上，Web文档除了HTML提供的标注结构外，还常常具有特定领域的、内在的深层结构，如产品目录就包含有产品价格、型号等结构信息。索引信息系

统如果能够识别、归纳出特定的深层结构模式，并将它与目录系统的自动构造结合起来，一定可以大大改善信息查询的准确性。Harvest 系统在这方面作了一些尝试，它使用的抽取信息的工具 Essence[15]可以识别出一些基本的文件类型，并可对不同类型的文档按不同的方式抽取信息。例如，Essence 可以从 Latex 文档抽取出 author、title 等信息。

3 Web 查询

3.1 查询与浏览

在 WWW 上找寻信息有两个基本途径：浏览与查询。所谓浏览，就是利用 Web 文档的超链结构在 WWW 上漫游，找到所需的资源。由于 WWW 是一个信息的海洋，通过漫游找寻信息不仅是一件很费力耗时的事情，还容易迷失方向。Yahoo 等目录系统通过为大量分散的 Web 文档建立层次型索引，使用户能够通过浏览目录找寻资源，大大地减轻了工作量。

所谓查询，就是用户给出文档的某些信息（关键字），由 Web 查询工具检索相应的索引数据库，确定文档的位置。由于 Web 文档数目巨大，缺乏良好的结构，用户难以准确地描述自己的需求，所以满足查询条件的文档往往很多，用户需要进一步地浏览或查询才能最终找到所需的资源。为了帮助用户尽快地在一大堆查询结果中发现所需的东西，Web 查询工具通常按照命中的文档与查询信息的相关程度分批顺序地返回有关信息（一般包括文档的标题、URL 和一些文档描述信息），文档与查询条件的相关性通常是由查询关键字在文档中出现的位置与次数决定的。许多查询工具还允许用户选择查询结果的格式和分批的大小。

3.2 查询工具

每个 WWW 索引信息系统都有自己的查询工具—Search Engine，它按照用户的查询要求，检索本系统的索引数据库，返回命中的文档信息。因为 WWW 的信息非常多，Web 结点散布全球，而且经常变化，单个索引信息系统的数据库很难涵盖所有的 Web 资源，有时人们不得不检索多个索引信息系统的数据库。为了满足这种要求，出现了独立于索引信息系统的查询工具—Meta-Searcher。

Meta-Searcher 没有自己的索引数据库，它给用户提供集成的查询界面，用户的查询（经过处理后）被转发给相应的索引信息系统，真正的查询过程是由索引信息系统的 Search Engine 完成的，查询的结果（经过 Meta-Searcher 处理后）返回给用户。由于不需要考虑索引数据库的建立与维护，Meta-Searcher 的开发者可以把注意力完全集中在查询上，设计更好的查询界面，提供更强的查询功能。另外，Meta-Searcher 是独立的查询工具，可以很方便地配置于客户端，避免诸如服务器过载之类的问题。下面，我们看几个具体的系统。

CUSI (Configurable Unified Search Index)[16]，一个可配置的 WWW 查询界面，提供诸如 Lycos、AltaVista、Harvest、Yahoo、AliWeb、WebCrawler 等索引信息系统的查询入口，用户每次可以选择一个 Search Engine 执行查询。CUSI 使用户可以比较方便、迅速地检索多个索引数据库，避免了在多个系统之间切换及重复输入查询关键字的麻烦。严格地讲，CUSI 只是多个 Search Engine 的索引，用户的查询未经任何处理就直接转送给相应的 Search Engine，Search Engine 返回的结果也不经处理直接提交给用户。类似的系统还有 All-In-One Search Page[17]、W3 Search Engine list[18]等。

MetaCrawler[19]，一个典型的 Meta-Searcher，它有统一的查询界面和查询语言，用户不需要选择 Search Engine，其查询要求经过 MetaCrawler 转换后并行地传送给 Yahoo、Infoseek、Lycos、WebCrawler、Galaxy[20]和 Open Text[21]，从这些系统得到的查询结果要经过 MetaCrawler 整理后再返给用户。这样，不仅查询的命中率提高了，而且用户也能够得到相关性较好的文档。由于各个系统并行工作，MetaCrawler 的反应还比较快。类似的系统还有 Savvy[22]等。

3.3 查询技术

Web 查询技术要解决两个问题：第一，要尽可能地把所有满足查询条件的资源都找出来，不要有所遗漏。第二，要尽可能地保证查询结果的准确性，不要把许多关系不大的东西都返回给用户。解决第一个问题主要依赖于索引数据库的建设，其覆盖面越广，能查询到的信息就越多。从这个角度来看，Meta-Searcher 比 Search Engine 要好一些。解决第二个问题主要依赖于查询语言的设计与实现。查询语言的功能越强，对所需资源的描述越详细，则查询结果的准确性越高。下面，我们来看一看当前的查询

工具所使用的主要技术。

·词组表示和布尔运算。几乎所有的查询语言都支持词组表示和布尔运算。一般用双引号""括住若干单词表示词组，只有当这些单词在文档中依次连续出现时才认为是匹配的。有的系统，如 WebCrawler，还引入了一个新的运算符 Near，表示要求参与运算的关键字在文档中出现的位置比较靠近（彼此相隔不超过25个单词）。布尔运算包括"与"、"或"、"非"三种，分别表示要求包含全部关键字，任意一个关键字及不能含有该关键字。

·范围查询。就是把检索限定在满足某些条件的 Web 文档的集合内。有的查询语言能够表示时间条件，使用户只得到较新的信息。Yahoo 可以把找寻的范围限定在最近3年，6个月、3个月、1个月、1周、3天和1天内加入的文档。AltaVista 可以只查询从某个时间到另一个时间之间的文档。HotBot 不仅可以把时间限定在数天、数月甚至数年之内，还能够将时间限制为某个确定的时间之前或之后。除了可以给查询加上时间限制以外，AliWeb、HotBot 等还能够把查询限制在一定的 Internet 域内，HotBot 也可以把查询限制在一定的地理范围之内。

Yahoo、Infoseek 等目录系统的用户常常把目录浏览与关键字查询综合起来使用。用户通过浏览确定文档所在的子目录，然后在子目录范围进行查询。Infoseek 还允许用户在前一个查询的结果范围内再进行深入的查询(search only under these results)。

AliWeb 把文档分为 Organization、User、Service、Document 等类型，用户在查询时可以指定所要的文档类型。

·结构查询与模式识别。HTML 文档有一定的结构，在查询时若能充分利用结构信息，将显著提高查询的准确性。AltaVista 可以要求查询关键字出现在 title、anchor、text、applet、object、link、image、URL 等特定的地方。Infoseek 可以把检索限定在 title、link、site、URL 等范围内。

除了 HTML 本身的结构外，Web 文档还有更深层次的模式结构。如何识别并使用这些模式，是提高查询质量的关键，在这方面已经有了一些努力。例如，Ahoy[23]可以根据输入的用户名，找出用户的个人主页(homepage)。Ahoy 通过 MetaCrawler 进行查询，然后根据其掌握的关于个人主页的模式，从得到的结果中识别出用户的个人主页。

·通配符与词形变化。许多查询语言支持通配符，如 AltaVista 允许用户用"*"匹配任意多个任意的字符，AliWeb 等允许使用正则表达式。Infoseek 等系统支持词形变化。例如，用户输入查询关键字"mouse"，Infoseek 会检测出所有包含"mouse"或"mice"的文档。如果用户要求把输入的词语当作人名进行检索，则 HotBot 会把相应的变化考虑进去。例如，当用户输入"Zeppo Marx"并要求系统把它看作人名时，HotBot 不仅会搜索"Zeppo Marx"，还会去搜索诸如"Marx Zeppo"、"Mr. Zeppo Marx"等变形。

·概念查询。ICE(Intelligent Concept Extr-action)是 Exite 独有的概念查询技术。当 Exite 为文档建立索引时，ICE 要弄清楚新的词语概念与其它词语概念是如何联系起来的。当用户查询时，ICE 不仅要找出与查询表达式匹配的文档，而且要找出所有包含与查询表达式概念相同的词语的文档。例如，用户查询"dog care"时，Exite 知道"pet grooming"是一个相关的概念。因此，所有包含"pet grooming"的文档都会被选中。

·模糊查询。有时，用户的目标在开始时是模糊不清的，只有一个大致的概念。在查询的过程中，目标才渐渐地清晰起来。为了满足这种需要，Exite 提供了"more like this"功能，使用户能够以查询的某个结果作为样本，进行新的查询。

·自然语言查询。WebCrawler、Infoseek 都支持简单的自然语言查询："plain English query"。系统能够理解用户用简单的英语表达的要求，执行相应的查询。作者向 Infoseek 询问"What is the best search engine of WWW"，得到了许多结果。

除了上述的方法以外，还有一些其它的查询技术。例如，Lycos 给用户提供了搜索图象、声音等媒体对象文件的手段。HotBot 允许用户声明 Web 文档中必须包含的对象文件类型，如 audio 文件、Java applets、ActiveX 文件等。Infoseek 等还可以找出所有包含指向某个 URL 的超级链接的文档。等等。

4 总结

WWW 是一个信息的海洋，单纯利用 Web 文档的超级链接遨游 Web，要发现所需的资源是一件很困难的事情。必须为数目巨大的 Web 文档建立索引，才能使在 WWW 上找寻信息成为可能，象 Ya-

hoo、Infoseek 等既能浏览又能查询的索引目录系统更受欢迎。由于 Web 资源众多又不断变化，而且涉及面非常广泛，从科技到娱乐无所不包，没有统一的语义模式，所以通过 Web Robot 定期搜集信息建立全文索引有其优越性。但是，更重要的是如何自动获取文档的结构信息，并将它与目录组织结合起来。这可能是今后的索引信息系统需要重点探索的问题。Meta tag 也许可以提供一些帮助，但这需要用户的广泛参与。

与数据库查询相比，Web 查询结果的质量要差得多。这一方面是由于 WWW 太大又没有数据库那样良好的结构，另一方面也说明现有的 Web 查询工具能力有限，特别是缺乏识别和使用深层语义模式的能力。因为 WWW 的用户及资源五花八门，遍布全世界，Web 查询工具要想使每个用户都可以准确地描述其需求是非常困难的。把查询工具与索引数据库分离，安置在客户端，可能是提高查询准确性的最好途径。这样，每个用户都可以按照自己的习惯配置查询工具，使之具有个人的独特风格和识别特定的语义模式的能力。考虑到 Web 资源的广泛性，可能只有用(受限的)自然语言才能较好地表述用户的要求。因此，一个能够理解(受限的)自然语言、具有语义模式识别能力、个人化的 Meta-Searcher 可能是比较理想的 Web 查询工具。

每个 Web 服务器都是一个自主的信息系统，如果它们都有良好的目录结构和查询功能，将大大增强用户获取信息的能力。在使 Web 服务器具备查询能力这个方面已有一些比较好的工作，如 WebGlimpse[24]，但还需要进一步深入。总之，要使 WWW 成为更加方便有效的信息系统，需要在索引数据库的构造、客户查询工具的设计及 Web 服务器的建设方面作出更大的努力。

参考文献

[1] http://webcrawler.com/

[2] http://www.yahoo.com/

[3] http://altavista.digital.com/

[4] http://www.excite.com/

[5] http://www.infoseek.com/

[6] http://www.lycos.com/

[7] http://www.hotbot.com/

[8] http://www.nexor.co.uk/public/aliweb/

[9] http://harvest.cs.colorado.edu/harvest/

[10] http://info.webcrawler.com/mak/projects/robots/active.html

[11] http://info.webcrawler.com/mak/projects/robots/norobots.html

[12] Martijn Koster. ALIWEB-Archie-like indexing in the WEB. Computer Networks and ISDN systems, 27(1994)

[13] http://harvest.transarc.com/afs/transarc.com/public/Harvest/technical.html

[14] http://guano.cs.colorado.edu/wwww/

[15] Darren R. Hardy, Michael F. Schwartz. Customized Information Extraction as a Basis for Resource Discovery. ACM Trans. on Computer Systems, 14(2)1996

[16] http://pubweb.nexor.co.uk/public/cust/cust.html

[17] http://www.albany.net/allinone/

[18] http://cuiwww.unige.ch/meta-index.html

[19] http://www.cs.washington.edu/research/metacrawler

[20] http://galaxy.einet.net/

[21] http://www.opentext.com:8080/omwlf-omw.html

[22] http://www.cs.colostate.edu/~dreiling/smartform.html

[23] http://www.cs.washington.edu/research/a-hoy

[24] http://glimpse.cs.arizona.edu/webglimpse

[25] Oren Etzioni. The World Wide Web: Quagmire or Gold Mine?. CACM, 39(11) 1996