

Dynamic Thesaurus and Dynamic Discovery of Distributed eLearning Materials

Michael Schlenker, Julika Mimkes, Eberhard R. Hilf

Institute for Science Networking Oldenburg, Germany
Ammerländer Heerstr. 121, D-26129 Oldenburg
schlenker@isn-oldenburg.de
mimkes@isn-oldenburg.de
hilm@isn-oldenburg.de

Abstract

A specialized search engine for scientific documents has been adapted by us to look for eLearning material available on the World Wide Web. Starting from known professional scientific institutions worldwide, listed in the PhysDep service, our system looks for probably relevant material. The method used for the search is explained and first results of the application on the domain of eLearning are shown.

Keywords: eLearning ; resource discovery; search engine

1 Motivation and Aim

eLearning will promote the mobility of students, allowing to choose classes or modules at different universities at the same time. Local lecturers will integrate distributed third party learning material into their lectures. Both will boost the international competition between universities.

What is needed as a first step for a future informed decision by students as well as lecturers is a service providing an overview of virtually all available relevant distributed eLearning material. Only if relevant and professional quality material is found and known, it will be used.

This calls for professional services discovering and cataloguing all available relevant material.

The current situation in eLearning is split in two parts, very similar to the situation in scientific publishing. On the one hand is the generally well structured and annotated material in professional learning management systems, on the other hand the more or less scattered and unstructured material placed on normal web pages. Learning management systems are the way to go. Open access standards and interfaces probably soon will see more widespread use, which will ease

the retrieval. But in the meantime large amounts of distributed eLearning material stay inaccessible.

Our system tries to find such material, material which is useful for eLearning but not yet available through standard metadata interfaces. This probably is the case for about 95 percent of the current material.

1.1 Background

We have started from the about 1.000 checked, reviewed links on distributed physics lecture material available from LiLi¹ including known collections of eLearning material from the distributed physics institutions². We then as a first step applied a HARVEST gatherer to crawl and fetch further material from the listed institutions. This search engine collected now already 22.270 links for physics.

Using a distinct field, in this case physics, and restricting to material linked by professional physics institutions, does assure virtually noise-free relevant material in contrast to general purpose search engines.

1.2 Techniques used

We developed and adapted a web-crawler, SHRIMPS (Simple HTTP Robot to Improve the PhysNet Service), an early version of which was designed by Svend Age Biehs for scientific documents, to automatically discover eLearning material on the web.

SHRIMPS extracts and analyzes web-pages and URLs using a given set of terms, assumed to be specific for eLearning materials. Starting from the about 5.000 home pages of Physics Departments

¹<http://www.physik-multimedial.de/lili/golili/lili.php>

²PhysNet-education <http://de.physnet.net/PhysNet/education.html>

worldwide it digs only those to any given depth, in contrast to general purpose machines. It stores the path information for the pages with their content for analysis and filtering and decides if a page is interesting based on the path to reach it and parts of its content.

SHRIMPS then presents a filtered list of candidates for relevant material in this context. Those lists can then either be certified by experts for inclusion into a database or directly fed into a full text indexing system like HARVEST as the next step.

For the international community of operators of the distributed webservice PhysNet <http://www.physnet.net> a tool has been set up so that each of them can operate and customize SHRIMPS independently and run it over regional subsets of physics institutes. This is important because of the mismatch of fast national and slow international connections, especially in some developing countries. Another important fact is the dependence of the system performance on the detection of cultural and language differences, which is obviously easier with operators embedded in the relevant cultures. These distributed gained results are then fed into the global PhysNet HARVEST system, operated for the EPS (European Physical Society). This sets up full text index files and feeds these into a set of about 15 worldwide distributed brokers which then can answer queries.³

2 Method

2.1 How to find Material

The discovery method used here is an adaption of a technique we used successfully for the discovery of scientific publications on institutional webservers.

Starting from the hypothesis that relevant material in the World Wide Web is in principle reachable by browsing web pages and following links, we analyzed the paths a human would take to access the relevant material. Material contained in locked databases and learning management platforms requiring a login for every contact to the material inside are obviously not reachable by this method.

We consider a wide variety of material to be possi-

bly relevant. This includes lecture notes, applets, link lists, examples, exercises, and tutorial style information. We also included course module descriptions into this collection, because those often link lecture material.

A human typically starts at a web page of a known authority, be it google or the home page of a recognized scientific institution. Examining the web page for links a human then follows links whose titles look interesting. For publications these are for example links named PUBLICATIONS, THESES etc. . The typical user browses the website from there, looking for content inside those general regions of the web site identified as relevant. Users recognize relevant material usually if they see it, reading the title and a description presented. Screenshots, typical stylistic conventions may help in this recognition process.

Dissecting this process reveals the sequence of events.

1. Fetch a web page.
2. Examine its title, its general content.
3. Decide if it is relevant.
4. Examine the provided hyperlinks.
5. Follow and examine every interesting hyperlink.

1. Just like a human using a normal web browser our system needs a way to fetch the pages. This is easy, we just have to take care that our system does not overload foreign servers. Thus we follow general web robot etiquette⁴.

2. The second step, examining title and general content of the page is harder. First we have to decide what actually is the title of the page. The (X)HTML standard[2] provides the HTML element `<title>` for the page title, but it is not always used correctly. The next common convention would be a heading `<h1> . . . <h6>` provided by the standard. There exist other title styles, graphical titles which may be linked to a descriptive text for the visually impaired, normal text that is just formatted in large and bold fonts, or even complex objects such as Flash animations or java applets. For a human visiting the page with a modern web browser this all appears in a similar way, for a visually impaired person it is often non-usable. The readability of web pages for the visually impaired is

³ The list of mirrors and operators can be seen at <http://de.physnet.net/PhysNet/crew.html>

⁴Wait for some seconds between page fetches, adhere to the robot exclusion protocol[1].

quite similar to the readability of a web page for a computer, as we need it.

3. If a meaningful title and content of the page can be extracted in plain text form, it is analyzed for relevance. Here we follow general information retrieval tradition and look for known good or known bad terms in the contents of the page. As an additional hint for the relevance of a page, we examine the file name of the page, and the title trail leading to the page.

4. Then we extract all hyperlinks from the fetched page. This crucial step is harder than expected, as more and more advanced web design techniques are used. The increasingly popular use of JavaScript⁵, Java and Flash animations allow the creation of web page navigation systems that do not even contain a single traditional HTML hyperlink. This makes them nearly unaccessible if no alternative navigation is provided. The use of `<frame>` elements or clickable graphical maps are obstacles that require special treatment to be recognized as valid links by a web crawler.

5. The final step before iteration is the classification of the links extracted in the previous step. This is done by comparison of the link title with a known good and known bad list of words. Some types of links are always followed, for example links from `frame` elements, as they behave like proxies for the actual content we are interested in.

Our relevance classification scheme is based on a statistical wordlist, created from our known good collection of physics material. For scientific publications we used the wealth of titles and abstracts of physical journal articles available⁶ with the OAI-PMH interface⁷ of large document repositories like ArXiv⁸ and IOPP⁹. We mainly looked for multi-word terms as these proved to have much higher weight for the assumption of relevance than single words.

For eLearning material we tried to reuse this thesaurus, and train the classifier with the known good text and link-data collected from the LiLi¹⁰ project. The training sets available for eLearning are smaller

by two orders of magnitude, so the resulting thesauri of terms are probably not as good.

As we do not assume that the program gets every relevant resource available at the web site, we added a second step to the process. We simply assume that pages near a page classified as relevant are probably relevant as well. So we added a proximity harvesting to the system, using the *Harvest*¹¹ search engine with very small search depth setting and collect the adjacent pages for the full text index.

This pollutes the index with some possibly not highly relevant material, but greatly increases coverage of relevant sources.

2.2 Implementation

Taking this model we created SHRIMPS a computer program that tries to browse sites looking for relevant material like a typical experienced human would. The whole system consists of three basic parts.

All of the system is unicode¹² based, so it is adaptable to different locales easily. This allows usage on a wide range of character systems like cyrillic, latin and others, if experts are available for language specific customisation of the filters.

The *SHRIMPS agent* is the workhorse of the system. It combines a web crawler, fetching pages from the WWW, with a link analyzer, parsing web pages, and a decision system, which decides on the path that will be followed. The *SHRIMPS agent controller* is a small middleware component based on SOAP¹³ and the transaction safe embedded *Metakit*¹⁴ database enabling communication with the central management server and managing runtime behaviour and configuration of multiple agent instances. The *SHRIMPS server* is the central management system, it manages the job lists, collects the returned data from the agents and allows configuration and monitoring of the system via a SOAP webservice or browser based user interface. The server is based on the Tcl webserver *tclhttpd*¹⁵.

A first prototype of the system was developed in the

⁵A scripting language, embedded into most current web browsers.

⁶We used PhysDoc-OAD <http://de.physnet.net/PhysNet/physdoc.html>

⁷The Open Archive Initiative Protocol for Metadata Harvesting. <http://www.openarchives.org>

⁸<http://www.arxiv.org>

⁹Institute of Physics Publishing <http://www.iop.org>

¹⁰<http://www.physik-multimedial.de/lili/golili/lili.php>

¹¹<http://www.sourceforge.net/projects/harvest>

¹²<http://www.unicode.org>

¹³The webservice standard from the W3 protocol working group. <http://www.w3.org/2000/xml/Group/>

¹⁴<http://www.equi4.com/metakit>

¹⁵<http://www.sourceforge.net/projects/tclhttpd>

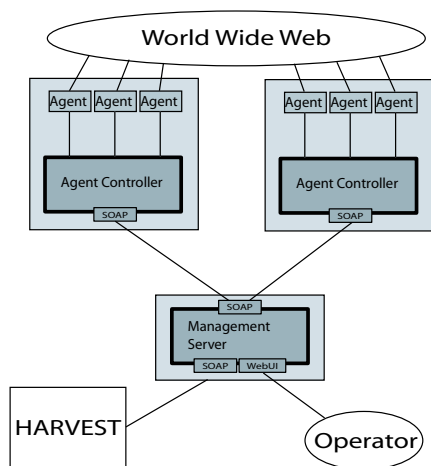


Figure 1: Hierarchy of the SHRIMPS system. The agent-controllers and their spawned agents are located on the same host and communicate via pipes. The management server communicates with the agent-controllers over the SOAP interface provided. User interaction and connection to the distributed HARVEST search engine is provided by the management server via a browser based user interface and a SOAP interface.

*Perl*¹⁶ language. It demonstrated the power of the concept, but proved to be hard to maintain and did not scale well to a distributed architecture.

A rewrite of major parts of the system in *Tcl* (Tool Command Language)¹⁷ was undertaken to make the system maintainable, scalable and portable to our distributed target environment. *Tcl* is a well established, highly portable scripting language often used in distributed/agent based computing [4], [5]. A decision against Java and .NET was made, as the development costs for those are usually up to a tenfold higher in terms of programmer time in contrast to typical scripting language solutions.

2.2.1 Shrimps agent

This part of the system handles the actual fetching and classification of the web pages. Instances of this agent are started as single processes by the agent controller on demand, when jobs have to be processed. Each agent processes a single website, and returns its results via the agent controller back to the central server system. Usually multiple agent instances are

working in parallel to enable a high scalability of the services. The agents themselves are rather simple and small systems and rely on their runtime environment, provided by the middleware, for transaction management and job control. In principle the system could handle specialized agents for different tasks with the same middleware component in place.

2.2.2 Shrimps middleware

The middleware component is mainly the agent controller. It communicates with the central server via a SOAP based publish/subscribe system based on a tuplespace/Lindaspace[6]. The agent controller acts as a dynamically configurable factory for the agent instances and can schedule agent execution based on local restrictions like server load or access restrictions. The agent controller can be deployed in a single file using a so called *Tclkit*¹⁸, on a wide range of platforms, without changes or configuration work on the local system. A basic infrastructure is in place to establish automatic updates of the middleware and the agents in a transaction safe way, so software updates can be installed without local administrator intervention. The middleware and agents are rather lightweight compared with a typical java virtual machine. The whole system fits into about 1.5 megabytes disc space including its virtual machine and all dependent libraries.

2.2.3 Shrimps management server

The management server provides the user interface to the operators of the service. It is a database backed embedded webserver handling user interface and the SOAP interface to the agent controllers. The system supports multiple concurrent users with their own disjunct workspaces. Users can create jobs by specifying list of websites to search. The browser based user interfaces provides allows configuration of the classification system, the monitoring of running jobs and supports result review. Post processing of their results is supported by an additional XUL¹⁹ based Mozilla plugin allowing them to review their findings with a single mouse click per page reviewed. A human reviewer with a fast internet connection can easily do a basic relevance review in under ten

¹⁶<http://www.perl.org>

¹⁷<http://www.tcl.tk>, [3]

¹⁸<http://www.equi4.com/tclkit>

¹⁹<http://www.mozilla.org/projects/xul/>
<http://www.mozilla.org/projects/xul/>

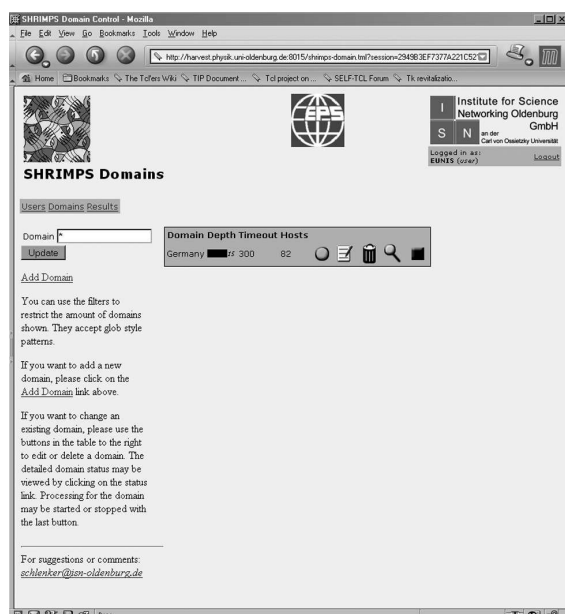


Figure 2: The web based user interface of the SHRIMPS management server. The screenshot shows the domain control UI, where new jobs can be created, started, stopped and the configuration of jobs can be changed.

seconds per page with this system. The system creates *Harvest* configuration files for breadth-first proximity harvesting of the resources on request.

3 Results

A first study was conducted with some german and british web servers and the physical thesaurus used for scientific physics publications on the assumption that lecture material is similar in structure. The classification terms for known good and known bad terms were adapted to the field of eLearning by adding terms like lecture material, virtual laboratory and their german equivalents.

We tested with 82 research institution web servers taken from PhysDep²⁰. 17 servers did not yield results, the remaining 65 servers returned a total of 1.809 unique URLs. The amount of results returned varied greatly from a single page to slightly above hundred URLs for a single entry point. This is expected and we observed similar effects when applying the system to scientific documents.

²⁰<http://www.physnet.de/PhysNet/physdep.html>

Classification	Number of URLs
Irrelevant material	924
eLearning relevant material	540
Personal homepages of lecturers / researchers	217
Scientific Publication lists	35
Other (unreachable at time of review)	93
Total	1809

Table 1: Distribution of primary URLs collected by the SHRIMPS system from 65 german and british research institutions, with about 30 percent relevant eLearning material.

From the 17 servers that did not yield results, 5 were laboratories that did no education, 8 had more or less broken entry pages using JavaScript to provide navigation without fallback. Four servers did redirect the agents to different domains already listed and were not included for that reason. This amount of failure is expected and is in the range experienced with scientific publications.

After manual review of the links fetched, we can see three large groups of links: Course or module descriptions, many of them with links to lecture notes, exercises or other lecture related material(30%). Personal homepages of lecturers/researchers that sometimes but not always also link to their lecture material (12%). A third large group are ordinary publication pages, laboratory yearbooks, lab journals or other introductory material from the sites searched.

Other frequently occurring pages in the results were the laboratory security handbooks, admission information for students and undergraduate or postgraduate program descriptions. Other topics occurring were job and PhD position descriptions. Really irrelevant are only (50%).

The total amount of URLs considered as relevant for eLearning after manual review is at about 30 percent of the total URLs collected. This looks like a small figure, but is far better than the percentage of relevant material that would be collected by brute-force indexing of the whole website, where one gets such trivia as the phonebook and lots of irrelevant organisational information.

A second study was conducted with some links to

eLearning material contained in the LiLi database. There we found some links to rather good material that wasn't reachable from the home page of the institution. Manual inspection of the whole site found no way to discover the material even though we knew it was there, so its outreach is limited to informed circles.

4 Analysis of results

Taking a closer look at the results shows some weaknesses in the current setup of the system. We found from studies with scientific publications that those are often listed and linked from the personal homepages of the staff members, so we assumed it would be worthwhile to check this assumption for lecture material and other eLearning material. The large amount of staff members listed in the result set without adjunct lecture material leads us to the conclusion that the special relationship existing between scientific publications and the individual researcher is much stronger as between an individual lecturer and his lecture material in comparison.

This conclusion is easily explainable with the different reputation inferred at present by lecturing and research publications. Most researchers care a great deal how their publications are received from the scientific community, the whole field of scientometrics is concerned with the impact of the publications.²¹

This however will change in the future in as much as lecture material of individual staff members gets accessible on the web and thus open to inspection and even refereeing by colleagues.

The weakness showed that our filters have to be further tuned for the much more heterogenous field of eLearning. Searching for scientific publications we could easily exclude large parts of the typical university home pages as irrelevant, as there were much clearer conventions used where to put publication lists. This seems not to be the case with eLearning material or more general material targeting students, at least at present.

This will probably change in the future in as much as the universities get organized and deploy university wide structures for their webservers eLearning content.

There seems to be no general guideline as to where

lecture material is deposited. Universities with learning management systems often put their lecture material and course details like the presentation slides behind locked doors so to speak. One has to be a member of the university network to access it or at least has to register to access the material. Our system can do next to nothing in such a situation unless a generic standard for robot logins (or metadata exchange) with such systems emerges. The OAI-PMH is in discussion as a solution for this problem of metadata harvesting.

One reason are of course the present copyright regulations, which allow access to many resources and their adaption to local classes only within small and closed groups like university intranets.

Most often we found eLearning relevant material adjacent to course descriptions. The other common place to find it were departmental or working group specific sites with educational toys or resources for schools. But even there no clear trend could be seen. Sometimes those links branched off from departmental sites, sometimes from some workgroup site and sometimes from the homepage of an individual lecturer or tutor.

But in general we are quite satisfied with the first results. We managed to identify a significant part of eLearning material available and openly reachable with this approach. The error rate is quite high, but we are optimistic that we can reduce the amount of irrelevant pages further by better tuning of the thesaurus and the url filters in use. The proximity harvesting we do after the initial candidate selection process completes usually increases the amount of relevant data by a tenfold.

5 Outlook

The discovery and retrieval of eLearning material is only the first but necessary step of what is needed for an effective service and the reuse of quality material. Essential is the training of the creators of eLearning material. The use of machine readable internationally standard metadata for the characterisation and description of their material, its application range in teaching, its technical requirements, has to be introduced. This includes easy to use interfaces to learning material metadata, instead of locked databases and learning management platforms. One example of such an interface is the already mentioned OAI protocol, which is used very succesful in the domain of scientific publications and digital libraries. Some

²¹One widely know example is the *Science Citation Index*.

learning platforms and eLearning sites have already implemented a OAI-PMH interface, or recommend its use[7] (LON-CAPA[8] advertises its OAI service, but tests show it non functional). In addition, due to the vastly varying quality and usability of eLearning material on the web, easy to use refereeing and certification quality filters have to be imposed. It has to be discussed which refereeing method of experts fits best to this application: blind refereeing as for journal papers, or open annotations, separately by staff members including usage experience feedback, and by students. LiLi tries to serve both.

eLearning material is much more complex in structure, application scenarios, user groups and technical requirements than scientific documents, and the sheer number of sources will quickly outnumber them. Respecting this, the design, development, implementation, experience, and communication of eLearning material management services is a most important upcoming field of work.

References

- [1] Martijn Koster. *A Method for Web Robots Control*
<http://www.robotstxt.org/wc/norobots-rfc.html> .
- [2] *XHTML 1.0 The extensible HyperText Markup Language*
<http://www.w3.org/TR/xhtml1/> .
- [3] John K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley, (1994).
- [4] Robert S. Gray. *Agent Tcl: A transportable agent system*. In Proceedings of the CIKM Workshop on Intelligent Information Agents, Fourth International Conference on Information and Knowledge Management (CIKM 95), Baltimore, Maryland, December 1995
<http://agent.cs.dartmouth.edu/papers/gray:agenttcl.ps.Z> .
- [5] Oswald Drobnik Anselm Lingnau, Peter Dömel. *An HTTP-based Infrastructure for Mobile Agents*. In Fourth International World Wide Web Conference Proceedings, pages 461–471, Boston, Massachusetts, December 1995.
- [6] Sudhir Ahuja, Nicholas Carriero, David Gelertner. *Linda and friends*: Computer **19**, (8) 26–34 (1986).
- [7] *IMS Digital Repositories Interoperability - Core Functions Best Practice Guide*
<http://www.imsglobal.org/digitalrepositories/> .
- [8] *LearningOnline Network with a Computer Assisted Personalized Approach (LON-CAPA)*
<http://www.lon-capa.org> .