



Insuma Distributed Search Engine

An Insuma GmbH White Paper

2002

Insuma software is a distributed search engine. It is used for full-text indexing and subsequent retrieval of documents in intranets or the Internet. The software is optimised for handling multiple types of data coming from multiple sources. A typical use scenario is a company with multiple branches connected by an intranet. In this case a copy of the software is installed into every branch. Documents residing in each given branch are indexed by the corresponding copy of the software, and searches for them are performed locally within the branch (see Figure 1). But it is always possible to initiate a search throughout all (or a subset of) branches simultaneously, as shown in Figure 2. The returned result set contains documents from all the searched branches.

1 Architecture

The software contains of three major components: the *Broker*, the *Collection*, and the *Crawler*. They communicate among themselves using the Object Management Group (OMG) CORBA protocols. Any number of components can run on the same or different computers in any combination.

Brokers present the users with a Web interface. They accept the user queries and forward them to one or more Collections for processing. Then they collect the partial results from the Collections and arrange them into a final list that is shown to the user.

Collections keep parts of the document index. They respond to the queries coming from the Brokers.

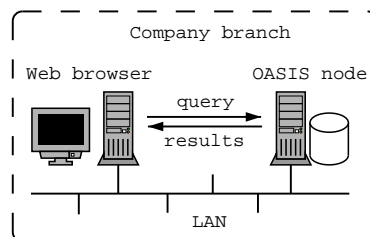


Figure 1: Local search scenario

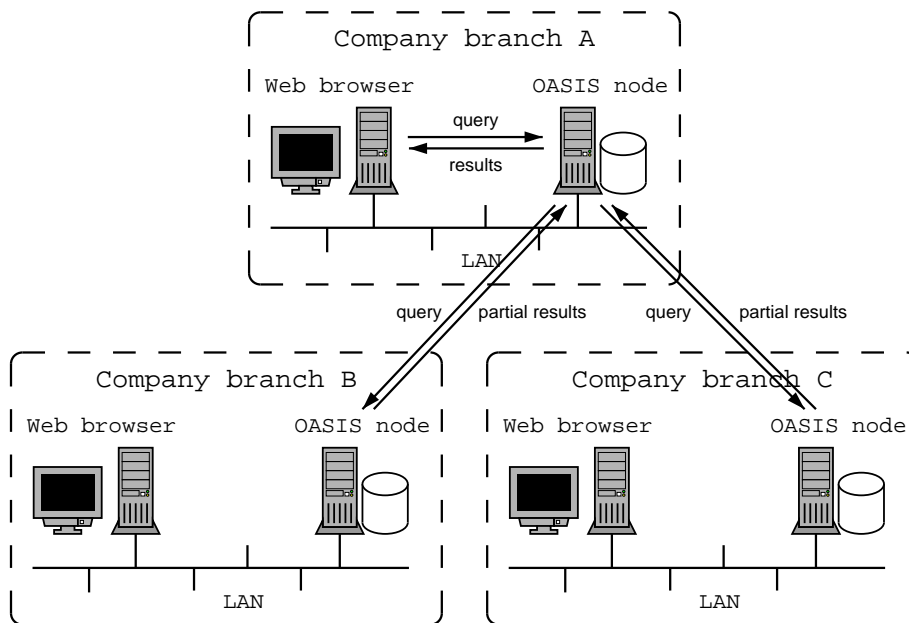


Figure 2: Distributed search scenario

Crawlers can be used for creation and updating the content of collections. These are Internet robots that download Web pages. Crawlers hunt only for documents related to a particular topic. To define a topic the system administrator provides the Crawler with several dozens of example documents that are considered relevant by a human expert.

The architecture has a number of advantages compared with the centralised approach:

- Good scalability. It is always possible to split the index in a larger number of nodes, i.e., run more Collections on a larger number of computers.
- Ease of customisation. The interface between the Collection and the Broker is defined in CORBA terms, which makes it platform and programming language independent. Legacy systems can be integrated into the company document management solution by writing CORBA wrappers conforming to the interface specification.
- Robustness. There is no single point of failure in the system. Local searches within a company branch can be performed even when the external link is down.
- Increased precision when multiple types of documents are indexed. Known ranking algorithms perform better when the indexed documents have similar length and structure. When, for example, e-mails and 100-page reports are indexed by two different Collections, more precise results are returned.
- Better security and access control. Users may be given different permissions for different Collections (e.g., the users from Accounting department may be denied access to the documents from the Development department).

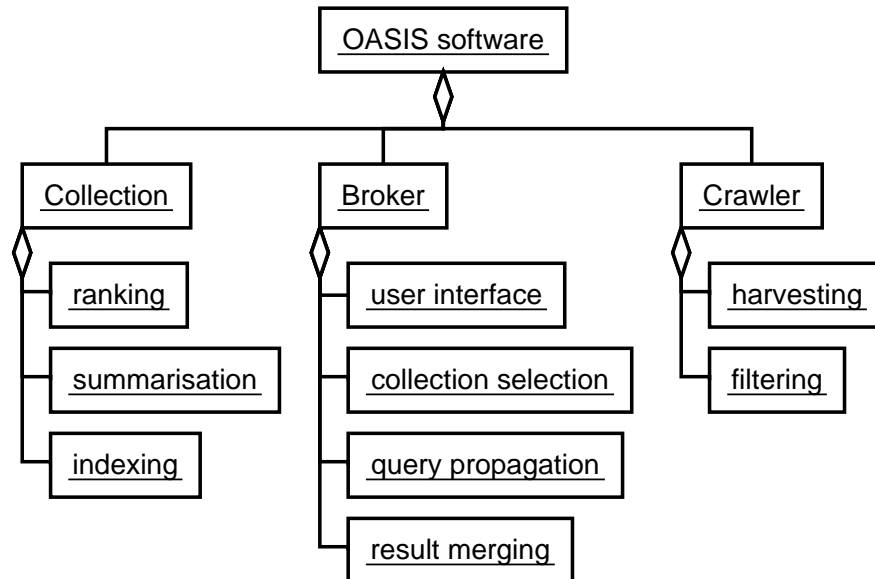


Figure 3: Insuma software functionality

A detailed view of the Insuma software architecture is presented in Figure 3.

The Collection functionality can be further subdivided.

Ranking is a process of selecting the documents that must be presented to the user as an answer to the user's query. The selection is done using the *index*, a previously constructed database of document properties.

Summarisation is a process of making a short extract or summary of a document that enables the user to decide whether she wants to see the whole document (i.e., click on the link provided) or not.

Indexing is a background process of creating the *index*, which is used by ranking and (possibly) summarisation. This part not directly visible to the end users.

Broker functionality can be also subdivided.

User interface translates between the natural language user queries and the formal query and result representation employed by the rest of the system.

Collection selection is a process of making the decision which collections to use for serving a particular query. This task becomes crucial when the number of collections is large (e.g., hundreds or thousands).

Query propagation means actual forwarding of the queries to the selected collections. This process can involve a number of format translations of queries to accommodate the differences between collections.

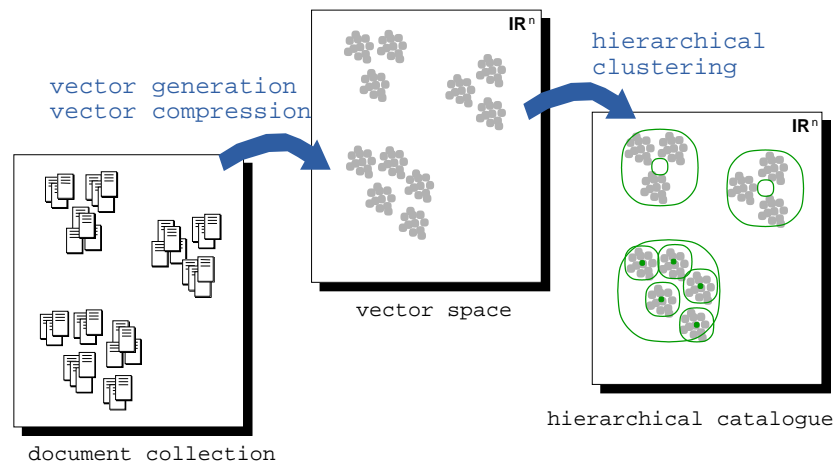


Figure 4: Hierarchical clustering with Neural Networks

Result merging is a process of arranging partial results from collections in a final result set.

Crawler can be broken down into two intercommunicating parts.

Harvesting is a process of fetching documents from the Web.

Filtering is a process of estimating whether the fetched documents match the collection topic, and correspondingly whether they must be included in the collection or discarded.

2 Advanced features

Several important techniques used in the Insuma search engine deserve mentioning.

Document clustering Traditional search engines present the user with a ranked list of documents. The documents in the beginning of the list are supposed to be more suitable for the user's information needs. This approach has inherent limitations. Multiple copies or versions of the same document often occupy the top 100 positions in the list. When the document types are diverse, it is difficult to decide what is more relevant to the user: a 50-page manual or a 50-line e-mail containing the same keywords.

Insuma software uses a Neural Network-based clustering algorithm for production of the final result set. The algorithm (working at the Broker during *result merging*) splits the documents (returned by Collections) into several groups. The documents in each group are more similar to each other than the documents from the other groups. The similarity estimates are based upon term frequencies. A simplified graphical representation of the process is shown in Figure 4.

Only a small number of representatives from each group (or cluster) is shown to the user. Thus the user can get an overview of a large result set by inspecting only several

documents. In particular, this solves the above-mentioned ranking problems. Duplicates are eliminated, and e-mails and manuals get into different clusters and thus are both presented to the user.

Relevance feedback Users tend to submit very short queries. Log analysis has shown that 80% of the queries consist of one word only. This information is not sufficient to describe the user's information needs. Thus it is essential to get richer input from the user, but the interface must remain very simple.

Insuma software relies on relevance feedback to accomplish this. After the users get results for their first query, they can mark the documents that meet their needs better than the others and press the 'Search again' button. The new results are more focused.

The relevance feedback works better in combination with document clustering. The documents returned to the user are guaranteed to differ from each other significantly, so the user can always choose which ones are better.

Relevance feedback is supported by all components of the Broker: *user interface*, *collection selection*, *query propagation*, and *result merging*.

Collection selection When the number of collections is very large (e.g., hundreds), propagation of all queries to all collections becomes unscalable. Only several collections must be selected for each given query.

The Insuma software uses statistical descriptions of collections to achieve this. The pre-computed descriptions (also known as forward knowledge) are stored in the LDAP directory. The Broker performs *collection selection* by querying the LDAP directory with the query terms. The statistical information returned by LDAP is then analysed and the propagation decision is made.

The algorithms employed allow scaling to tens of thousands of collections. This number is sufficient for indexing the whole Internet.

Topic-oriented Web harvesting The InsumaScout product performs focused search for documents matching the topic defined by the system administrator. To define a topic, the administrator gives the Scout several dozens of documents that match the topic. The Scout analyses these sample documents and builds a corresponding filter. The filter is used to estimate how well does any given document match the topic.

The Scout uses advanced search strategies to follow the links that are likely to point to relevant documents first. Adaptive strategy is used to reorder the queue of unfetched URLs according to the relevance of the fetched ones.

This approach makes a qualitative difference compared to the traditional recursive harvesting. It is impossible to download all the documents in the Web and then remove the ones that do not pass the filter. But it is possible to find a substantial share of documents present on the Web that do pass the filter.

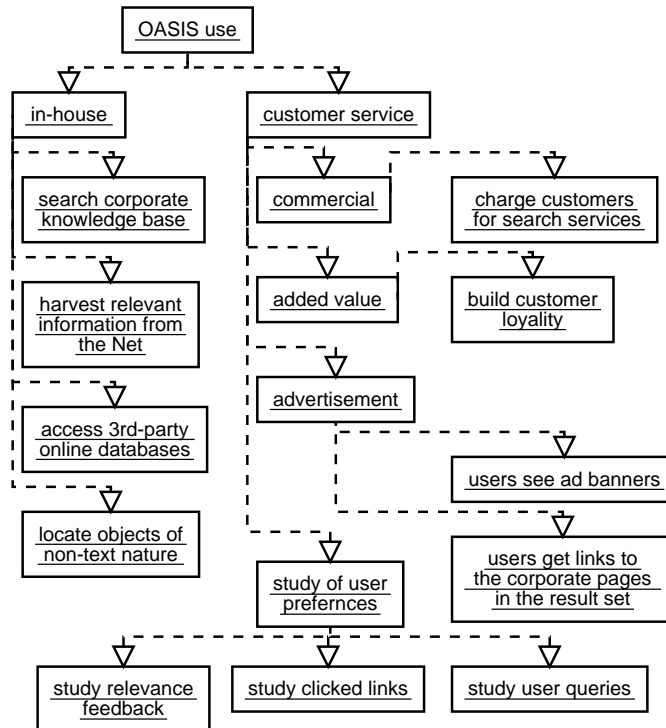


Figure 5: Insuma use scenarios

3 Implementation

The software is written in ANSI C and Python. It was tested under Solaris and Linux, but should compile under any sane Unix with minimal changes. The required libraries are all open-source and free.

The most significant of the third party products used is the CORBA object broker implementation from Xerox, called ILU. It compiles under a very wide range of platforms including MS Windows and support mappings into multiple programming languages including C, C++, Java and Python.

The use of these tools and libraries makes the product easily portable and customisable.

4 Use scenarios

The technology used in the Insuma software has rather generic nature, so it is hardly possible to envisage every possible use of it. Only some of the possible scenarios of use of the Insuma software are presented in figure 5. They fall down into two categories: in-house use (i.e., an organisation installs and runs some parts of the Insuma software for the use of the employees on the organisation) and customer service (i.e., an organisation expects to gain some benefits from third parties using the service). Any combination of the above-mentioned use scenarios is possible, e.g., a company can use the search

service internally and for indexing its public Web pages. Short descriptions of some of the use scenarios are presented below.

Search corporate knowledge base. Every large company generates a large amount of paperwork during its life cycle. Typically considerable heterogeneity is present with respect to the document formats used (different versions of MS Word, WordPerfect, MS Excel, Adobe PDF, HTML, textual fields of database records, e-mails) and document locations (file servers, workstations, off-line archives, database servers). The documents contain valuable information about customers, procedures, business cases, etc. Effective and efficient access to such information helps in using this information, thus allowing the organisation to avoid recurrent mistakes, save effort, and provide better service. In case of companies with many branches connected by an intranet a distributed search architecture is most appropriate.

Added value service. An Internet service provider or a Web portal may want to provide its users with free search services, in order to create added value and differentiate itself on the market. For example, a chain of Internet shops can be made searchable, with one-stop searches performed in one or any number of shops at the same time.

Harvest relevant information from the Web. Companies are often interested in gathering publicly available information in the areas related to their business. The Web is a major source of such information. The companies need a *focused* harvesting that brings relatively little junk. The Crawler is a right tool to do the job.

Access third party on-line database. There exists a large number of commercial on-line databases and news feeds (on business, politics, electronic components, research papers, etc.). Large companies typically subscribe to several such databases. Normally these databases are provided by unrelated organisations and have different interfaces, require different logins, and each of them must be queried individually. As a result, employees waste time re-typing the same query into several prompts. A one-stop search over all of the available databases can be provided. It can also include the in-house company archives and free Internet sources.

Users get links to the corporate pages. especially to the customers that actively seek these. A traditional way for companies is to put advertisement banners on search engine sites, that are presented either randomly or in connection with a certain predefined set of keywords appearing in the user query. This method has disadvantages: the banners reference only a small number of the company pages, and are often irrelevant to the query. Users tend to ignore or automatically filter out such banners as a result. It is much more desirable for a company to be able to receive the user's query and provide links to the most relevant pages of their site as a part of the result set returned by the search engine. The company may achieve this by running a *collection*.

Study of user preferences. The companies involved in the business-to-customer e-commerce value information on users' areas of interest, patterns of Internet use, etc. This activity is legitimate and reasonable provided that the users' privacy is vigorously protected. Any search engine is capable of recording the user queries,

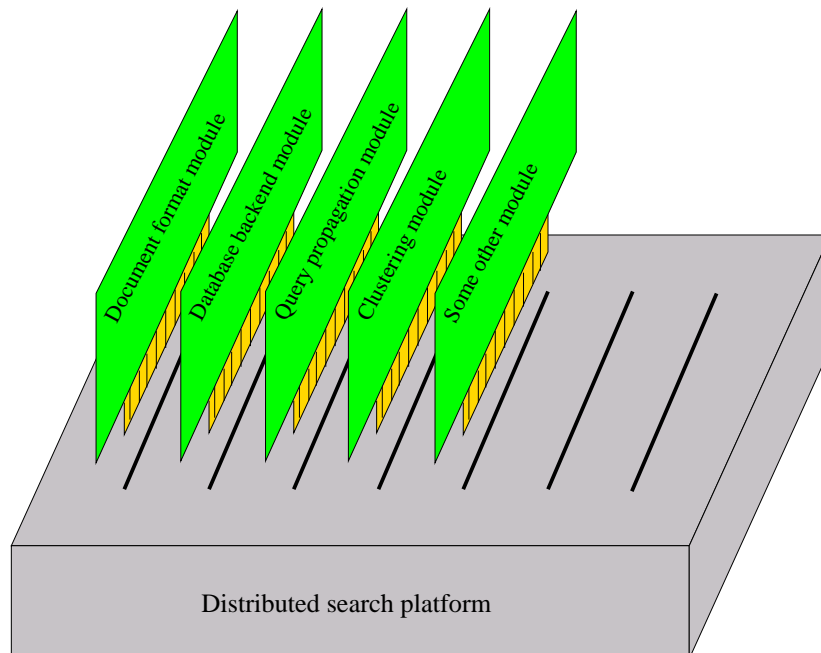


Figure 6: Insuma platform and modules

and many Internet companies buy such log data. Since Insuma is a distributed search system, the load per node is lower, and more resource-intensive query processing can be done. It includes tracking the visited links (the user gets a link to a script at the Insuma site, and gets automatically redirected to the site containing the actual page; but the broker is able to record the fact that the link was clicked). Also, Insuma supports so called *relevance feedback*, a technique allowing the user to mark the pages in the result set that are more relevant to his needs and resubmit the query. Thus the search engine is informed which pages are liked by the users, and also associate this information with the user queries.

Charge customer for search service. This model can be used for providing access to proprietary databases.

Locating objects of non-text nature. Some cases are known when the techniques developed for text search are applicable to other domains. For example, the *AltaVista* software is used for finding matching DNA sequences in the *Human Genome* project.

The above list is definitely open, please contact us if you have some other use for the Insuma technology in mind.

5 Customised solutions

Insuma software consists of a generic public domain platform and customised modules (see Figure 6). The modules communicate to the platform over CORBA, and thus

can have a different license. Proprietary modules (or plugins) are available from the Insuma GmbH company. Some of these modules can be purchased immediately, and the others are tailored to order. The modules can be grouped by functionality as shown in Figure 3. They are described below.

- Customised HTML templates for the Web interface. The layout of the user interface can be changed according to the company's specification. The changes include company logo, Web page headers and footers, banners, shortcut links to some sections of the company Web site, etc. The changes do not effect the functionality of the broker.
- Customised Broker. The functionality of the broker is extended or changed according to the agreed specification. For example, the broker is not called directly using the CGI interface, but is rather invoked by some client software and returns the search result in some format other than HTML.
- Subject-specific thesaurus. The search precision can be improved by using thesauri that contain synonyms, abbreviations, etc. The most effective thesauri are designed with some subject area in mind, e.g., Banking or Law. Third-party thesauri can be integrated into the search engine, or new ones can be created from scratch.
- Improved morphological support. The default public domain modules use public-domain dictionaries for morphological analysis. More complete and accurate proprietary dictionaries are available. When they are used, the search results are more precise and complete.
- Integration with customer data bases (e.g., MS SQL, SAP). This is done using CORBA wrappers. The integration eliminates the need for data export before indexing, thus making index always up-to-date and saving storage space and processing resources. It makes it possible to use field names of the database in the queries, thus increasing search precision.
- Indexing of document formats other than HTML and plain text. Some formats can be supported immediately (Postscript, Adobe PDF, MS Word, WordPerfect), others require customised development.
- Advanced relevance feedback. Proprietary algorithms are used to increase precision and recall for repeated queries.
- Statistical analysis of the user queries and search results. The module generates reports that include the most popular keywords in the queries, the pages that were ranked as relevant in the largest number of cases, the pages that were never shown as search results, unknown keywords entered by the users, etc. These reports provide a foundation for improvement of the indexed Web sites and if the search engine configuration.
- Analysis of links followed by the users. It provides additional information related to the previous item.
- Advertisement banners.

- Advanced neural network-based clustering of the result set. It increases the precision and recall of search results compared to the built-in public domain algorithms.
- Advanced document summaries. The result pages contain more informative annotations to the presented URLs. The keywords from the queries are highlighted, the document title and author identified wherever possible, etc.
- Automated classification of large document sets using the hierarchical neural network-based clustering (HRCL). This allows to impose a tree-like topical structure on a large unstructured set of documents (e.g., all documents residing on a server).

6 Conclusion

The Insuma search system is an implementation of an open distributed search architecture. The availability of public specifications of the protocols and interfaces used, as well as of the public domain source code of the platform, puts the users in control. The users are no longer locked into a single vendor solution.

The system is highly modular and thus customisable to the user needs. Insuma GmbH offers its services in making such customisations. It also provides support and consulting if such customisation is done in-house. The use of CORBA allows for easy creation of cross-platform environments, including integration with legacy systems.

Please contact Insuma GmbH for integration of the latest search technology in your business process.

Contact details

Adresse: Insuma GmbH
Sand 13
72076 Tübingen
Germany
Web: www.insuma.de
E-mail: info@insuma.de
Phone: +49-(0)7071-29 78997
Fax: +49-(0)7071-29 5062

References

- [1] U. Heuser, A. Babanine, W. Rosenstiel: *HTML Documents Classification using (Non-linear) Principal Component Analysis and Self-Organizing Maps*, Proc. of the Fourth International Conference on Neural Networks and their Applications (Neurap'98), March 11-13, 1998, Marseilles, France, pp. 291-295

- [2] U. Heuser, W. Rosenstiel: *Internetsuche und Neuronale Netze - Stand der Technik* Research Report WSI 98-10, Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, D-72076 Tübingen, Germany, ISSN 0946-3852, 64, pp., 23. September 1998
- [3] M. Bessonov, U. Heuser, I. Nekrestyanov, A. Patel: *Open Architecture for Distributed Search Systems*, Proc. of the Sixth International Conference on Intelligence in Services and Networks (IS&N'99), April 27-29, 1999, Barcelona, Spain, In: H. Zuidweg, M. Campolargo, J. Delgado, A. Mullery (Eds.) *Intelligence in Services and Networks*, Lecture Notes in Computer Science 1597, pp. 55-69, Springer-Verlag, 1999
- [4] U. Heuser, W. Rosenstiel: *Das Hierarchische Radius-basierte Competitive Learning (HRCL) im Vergleich mit statistischen und neuronalen Clusteranalyseverfahren* Research Report WSI 99-08, Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, D-72076 Tübingen, Germany, ISSN 0946-3852, 52, 52 pp., 12. Mai 1999
- [5] *OASIS: Distributed Search System in the Internet*, Patel, A., Petrosjan, L., Rosenstiel, W., ISBN 5-7997-0138-0, 1999, 570 p.
- [6] A. Patel, M. Blinov and M. Bessonov: *Reference model and functional architecture for information availability*. Computer Standards and Interfaces, vol. 21, issue 3, August 1999, pp. 273-285, Elsevier Science, Holland.
- [7] U. Heuser, W. Rosenstiel: *Automatic Generation of Local Internet Catalogues using Hierarchical Radius-based Competitive Learning* In: W.Horn (ed.): *ECAI 2000*. Proc. of the 14th European Conference on Artificial Intelligence, IOS Press, Amsterdam, 2000, pp. 306-310
- [8] U. Heuser: *Automatische Internet-Katalogisierung mit Hilfe des Hierarchischen Radius-basierten Competitive Learnings*, (engl.: *Automatic Generation of Internet Catalogues using Hierarchical Radius-based Competitive Learning*), PhD thesis, University of Tübingen, Faculty of Informatics, June 2000, Published at: Logos-Verlag Berlin, ISBN: 3-89722-561-1, 226 p.