

IST R&D PROJECT
SHARED-COST RTD PROJECT
HUMAN LANGUAGE TECHNOLOGIES
PROJECT OFFICER: YVES PATERNOSTER



clarity 

Newer Version of Clarity Distributed System

D6-3

7 SEP 2003, DEPT. OF COMPUTER SCIENCE
/WP6/VERSION 1.0

GEORGE DEMETRIOU

IST Project Number	IST-2000-25310	Acronym	Clarity
Full title	CROSS LANGUAGE INFORMATION RETRIEVAL AND ORGANISATION OF TEXT AND AUDIO DOCUMENTS		
EU Project officer	Yves Paternoster		

Deliverable	Number		Name	
Task	Number		Name	
Work Package	Number		Name	
Date of delivery	Contractual		Actual	
Code name			Version 1.0	draft <input type="checkbox"/> final <input checked="" type="checkbox"/>
Nature	Prototype <input checked="" type="checkbox"/> Report <input type="checkbox"/> Specification <input type="checkbox"/> Tool <input type="checkbox"/> Other:			
Distribution Type	Public <input checked="" type="checkbox"/> Restricted <input type="checkbox"/> to: <partners			
Authors (Partner)	George Demetriou, Heikki Keskustalo, Bemmu Sepponen, Patrick Herring, Kristofer Franzen, Jussi Karlgren, Fredrik Olson, Rob Gaizauskas, Mark Sanderson			
Contact Person	Mark Sanderson			
	Email	M.Sanderson@sheffi eld.ac.uk	Phone	+44 (0) 114 22 22648
			Fax	+44 (0) 114 27 80300
Abstract (for dissemination)	The requirement for Cross-Language Information Retrieval (CLIR) systems to access and analyse data in different languages means that they often need to integrate information from a variety of heterogeneous sources and software services at disparate sites. In this deliverable we describe the architecture of the Clarity system, a CLIR system for English, Finnish, Swedish and Baltic Languages. Clarity can be classified as a <i>distributed</i> CLIR system and its architecture comprises three layers: (i) an <i>application layer</i> that provides the query translation, document retrieval and language processing services, (ii) an <i>interface layer</i> that includes the user interface, and (iii) a <i>communication layer</i> that is based on Web services and acts as a <i>middleware</i> between the user interface and the services of the application layer. The architecture has been designed to speed up software development, to support service interoperability and information sharing and to promote user interactivity.			
Keywords	Cross language information retrieval, distributed architectures, Web services			

1 INTRODUCTION

The requirement for Cross-Language Information Retrieval (CLIR) systems to access and analyze data in different languages means that they often need to integrate information from a variety of heterogeneous sources and software services at disparate sites. In this deliverable we describe the architecture and the software components of the Clarity system, a CLIR system for English, Finnish, Swedish and Baltic Languages.

Clarity can be classified as a *distributed* CLIR system and its architecture comprises three layers: (i) an *application layer* that provides the query translation, document retrieval and language processing services, (ii) an *interface layer* that includes the user interface, and (iii) a *communication layer* that is based on Web services and acts as a *middleware* between the user interface and the services of the application layer.

The architecture has been designed to speed up software development, to support service interoperability and information sharing and to promote user interactivity. Cross-language experiments for English and Finnish for the 2002 iCLEF task (Petrelli et al 2002) have confirmed that Clarity's architecture is capable of facilitating practical and efficient CLIR (the architecture used for those experiments was even less advanced than the one described in this document).

Clarity's architecture was designed with the following aims:

- To promote application interoperability and seamless integration of system functions no matter where the back-end applications of these functions reside.
- To support the system's scalability as more languages and features are gradually added to the interface (e.g. the Baltic languages Latvian and Lithuanian).
- To allow for independent development of system modules at project partner sites, namely Sheffield, Tampere, Stockholm and Riga.

Full documentation for the Clarity architecture and the associated services is provided through the project's website at:

https://www.dcs.shef.ac.uk/research/groups/nlp/clarity/partners_only/tech_doc.html

2 CLARITY SYSTEM OVERVIEW

Clarity is implemented as a *distributed architecture* based on *Web services* (W3C 2002a), i.e. services that are available on the Internet, use standardised messaging formats, such as XML, and enable communication between applications without being tied to a particular operating system or programming language.

Distributed CLIR architectures have been researched and developed in the past in the SPIRIT (Fluhr et al 1996), and TITAN (Hayashi 1997) projects. Both these architectures have similarities with Clarity's, but their focus is more on managing distributed document collections and on methods for merging the retrieval results rather than managing distributed software processes. Clarity is to our knowledge one of the first, if not the first system that applies modern aspects of Web services to model distributed applications for CLIR.

Clarity is designed as a three-layer system (shown in the diagram of Figure 1) with the *user interface* as a front-end (*Interface Layer*), the data sources and services on the back-end (*Application Layer*) and a *middle layer* that separates the interface from the system services and provides the communication between them (*Communication Layer*).

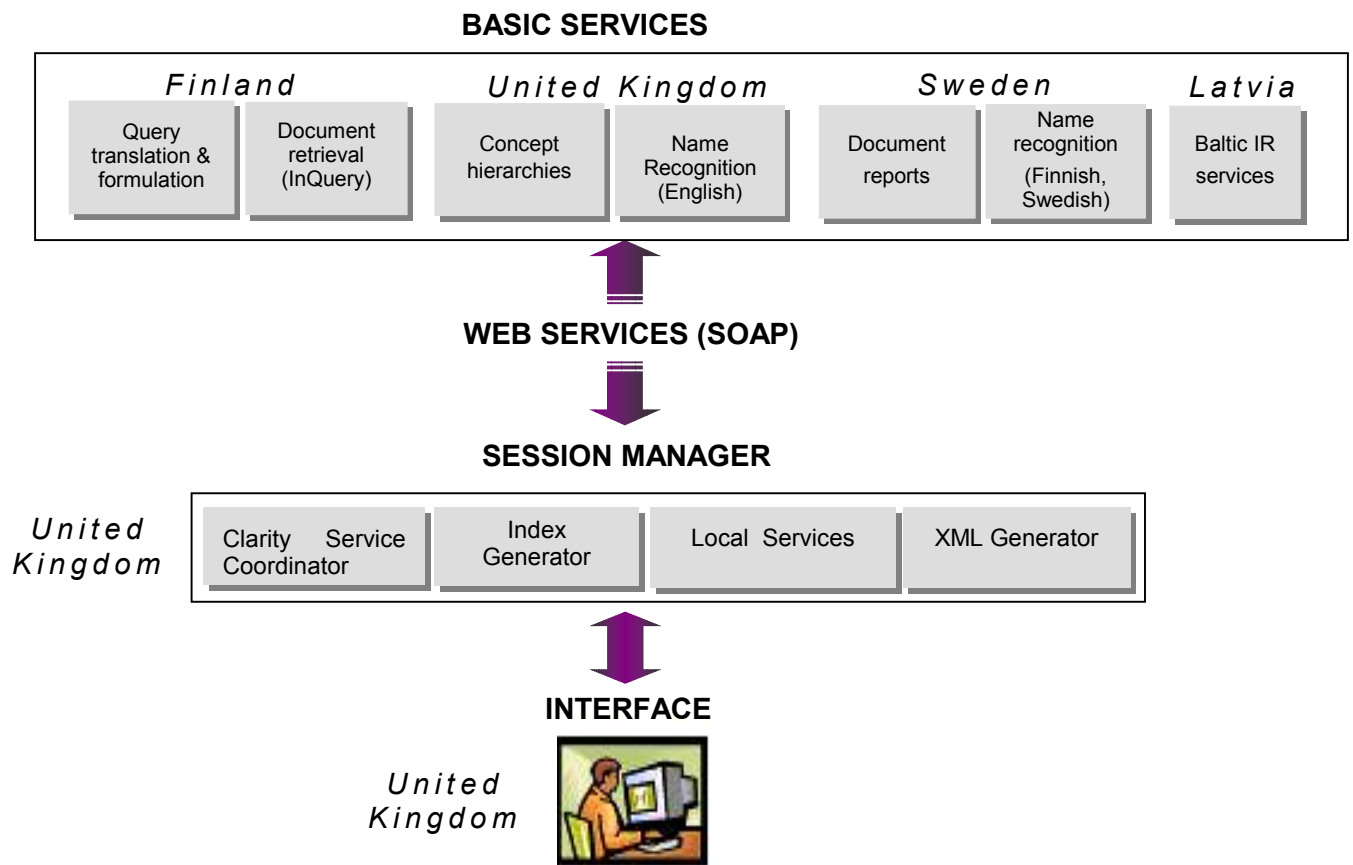


Figure 1: Clarity architecture overview.

Clarity supports the following functions:

- Query translation for English, Finnish, Swedish, Latvian and Lithuanian.
- Document retrieval and merging for multiple languages.
- Identification of query terms in each retrieved document.
- Translation of target language document titles.
- Extraction of the document's 'best passage' i.e. document excerpt with maximal 'density' of query terms.
- Concept hierarchies for a subset of the target languages involved.
- Document reports with filtering functionality based on stylistic analysis and proper name recognition.

3 THE APPLICATION LAYER

The Application Layer incorporates a range of services for efficient CLIR interaction:

- **Query translation and formulation services:** These include services for the translation and formulation of the source query.
- **Term processing services:** Services for processing terms such as term translation, normalisation and n-gram matching.
- **Document retrieval services:** services that perform retrieval of documents and provide associated functions such as the identification and highlighting of query terms, extraction of best passages, etc.
- **Document organization and presentation services:** Services for organizing and presenting foreign language documents to support the users in their analyses.

3.1 QUERY TRANSLATION AND FORMULATION SERVICES

The translation of the query is done with the UTACLIR query translation system (Hedlund et al 2003). For term translation, UTACLIR uses Globaldix, a multilingual dictionary for 18 languages and 650K words. Multilingual dictionaries often have poor coverage for proper names, multi-word phrases or compound terms. The current version of UTACLIR compensates for such inadequacies with the use of clever query formulation mechanisms.

Proper names and other terms not to be translated can be specified in the query by annotating them with a special character ('@'). For multi-word phrases, UTACLIR takes advantage of the proximity InQuery operator #uwN ('unordered window') that specifies that any set of (synonym) terms enclosed within #uwN(...) must appear within a window of words specified by N in no specific order. As an example of query translation and formulation with Finnish as the target language, UTACLIR will generate the following result for the English query "seat belt" (the quotes indicate phrase boundaries):

```
#sum( #uw2( #syn( istumapaikka istuinosa takamus) #syn( vyö  
hihna vyöhyke) ))
```

The #uw2 operator indicates that any combination of terms between the synonym set for 'seat' (istumapaikka, istuinosa, takamus) and the synonym set for 'set' (vyö, hihna, vyöhyke) must occur within a window of two words in the Finnish documents (no order is assumed). The size of the unordered window is adjusted automatically by UTACLIR according to the number of terms in the phrase (up to a maximum of five words per phrase).

UTACLIR implicitly performs the morphological normalisation of terms before translation. For compounds, i.e. multiple words written together as a single unit, if the

compound is untranslatable as a whole word, UTACLIR tries to split it into constituents, translate each constituent and form a proper query structure on the basis of these translations.

The translation routes that are currently available for source and target languages are summarised in Table 1.

Source language	Target language
English	Finnish
English	Swedish
English	Latvian
Finnish	English
Finnish	Swedish
Swedish	English
Swedish	Finnish

Table 1: Translation routes in Utaclir.

3.2 TERM PROCESSING SERVICES

Translation services for single terms make use of the Motcom and Globaldix dictionaries plus the Duden dictionary table for CLEF runs in 2000 and 2001. The underlying translation mechanism is similar to UTACLIR but the output is in different form as there is no need to wrap the translation with InQuery operators.

Lexical normalisation, i.e. the morphological normalisation of a term to its basic form, is an essential part of query translation but can also be used in other operations such as the extraction of document frequencies (for generating the concept hierarchies). Lexical normalisation is provided by the two-level morphological analyser TWOL by Lingsoft (Koskenniemi 1983).

To deal with the problem of query terms that may not be in the indexed databases, an algorithm that performs fuzzy pattern matching has been developed. The algorithm applies a novel n-gram matching technique and matches up to three best alternatives found in the database index. For example, for the query term 'ericsson', the algorithm

produces alternatives such as ‘ericsson’, ‘erickson’ and ‘eriksson’; these terms can subsequently be used for query expansion (Pirkola et al 2002).

3.3 DOCUMENT RETRIEVAL AND ASSOCIATED SERVICES

Clarity uses the InQuery (version 3.1) retrieval engine developed at the University of Massachusetts (Callan et al 1992). The text collections used include the CLEF collections for English (Los Angeles Times 1994 – 113K documents), Finnish (Alma Media 1994-95 – 55K documents) and Swedish (1994-95 – 142K documents).

For each document, associated information is provided such as the document id, language code and document score. In addition, a set of related services have been developed to give the user an insight into the content of foreign language documents. One such service is the translation of document titles. Clarity does not attempt full translation of the title as a whole, because the available resources do not have this capability. Rather, the translation of the title is attempted on a term-by-term basis as a way of giving the user an idea of the document’s subject in her/his own language.

A second service is the identification of query terms in the documents. This serves to allow users identify those documents that contain query terms and may be relevant. The matched terms appear in the same order as they occur in the document body. Different morphological word forms are not listed but the terms appear in the forms they occur in the translated queries. This choice does not limit the effectiveness of the term matching component because via another InQuery mechanism, the system displays the inflectional or compound forms of the matched terms highlighted in the document body.

A third service is the extraction of a *best passage* for each document. By ‘best passage’ is meant a document fragment of fixed length with the maximum density of matched query terms. The best passage serves as a good, representative excerpt of the document body that users can read in order to decide if they would like to browse the whole document. Of course, since there is no way to know in advance that this excerpt is indeed a good summary of the document, it should be treated as an ‘answer-indicative sentence’ (Wilkinson 2001).

Because the scores of documents from different languages may not accurately reflect the ranking of the documents (with respect to relevance), the merging mechanism for results from multiple languages is done by proportional representation of documents for every language (i.e. each target language gets the same number of documents).

Document retrieval is currently available for English, Finnish, Swedish, Latvian and Lithuanian collections. More details about the aforementioned services at Tampere can be found at the documentation URL:

http://www.uta.fi/~ccheke/clarity_readme.html

3.4 DOCUMENT ORGANISATION AND PRESENTATION SERVICES

Several of the bottlenecks of interactive information access systems in general become even more aggravated in the case of cross-lingual data. Users interact with information access systems at several points, among them formulating their information

need, selecting documents from a set, and perusing documents individually (see e.g. Oard, 2000). Clarity aids users by translating queries automatically, but does not translate target language documents, assuming instead that users are conversant if not fluent in the target language. The step in between, interaction with a set of documents, the reduction of a large retrieved set to a smaller set, is misleadingly similar across languages, but judging the quality and pertinence of documents in another language is non-trivial and risks confounding the unwary user (Hansen and Karlgren, in press). Most information access systems present retrieved items solely in the form of a list sorted by descending probability of relevance as understood by the system. Clarity instead supports the selection process by providing users with an overview of retrieved results. In Clarity the retrieved list is the default first view of retrieved items - but as a complement, the results can be presented in the form of a concept hierarchy or alternatively a textual retrieval report.

In concept hierarchies, documents are clustered with respect to a hierarchy of concepts that are derived from the set of retrieved texts; the generated structure is presented as a set of hierarchical menus according to the statistical principle of 'subsumption' (Sanderson and Croft 1999). The appealing characteristic of 'subsumption hierarchies' is that they can be automatically extracted without the need for prior knowledge or training data.

Compared to hierarchies extracted for monolingual IR, cross-language concept hierarchies have to overcome possible problems with term translations that may hamper the quality of the hierarchies. One problem is the problem of limited translation coverage for target language terms. Poor translation coverage can have a statistical effect on the subsumption algorithm because the number of input terms to the algorithm tends, in some cases, to be significantly lower than the one expected from the monolingual case¹. The second problem is the problem of multiple translations. Multiple translations are produced for a number of reasons, such as because the source language term may have multiple senses, the translation dictionary includes a list of translation synonyms for the term or the term may be a compound in which case each of its constituents is translated individually. The occurrence of multiple translations can clutter the appearance of the concept hierarchy and can confuse users in their relevance judgements.

Currently, the languages available for concept hierarchy generation are English and Latvian.

Documentation about the 'subsumption' software for the concept hierarchies is provided at:

https://www.dcs.shef.ac.uk/research/groups/nlp/clarity/partners_only/con_hier_description.pdf

¹ In our experiments it was found that about 30% of Finnish terms could not be translated to English.

Documentation about the web services at Riga and Sheffield can be found at:

<http://clarity.tilde.lv/ConceptHierarchies/ConceptHierarchies.aspx>

and

https://www.dcs.shef.ac.uk/research/groups/nlp/clarity/partners_only/CiquestSOAP1_1.pdf

The multi-document report service in CLARITY is designed to give an overview of the retrieved set of documents. The multi-document report describes the retrieved set in terms of various extracted features of individual items: number or frequency of search terms (see previous sections); alternative keywords, names and other data extracted from the documents; language; and text genre (e.g. news article, opinion piece, interview) or style of text (e.g. argumentative, subjective, personal) as described in Karlgren (1999). These informational elements can be activated as filters to help users reduce the retrieved set to manageable proportions before inspecting individual documents or items.

An XML Data Type Definition (DTD) for the document report service at Stockholm is provided at:

<http://sics.se/~stny/UBI/DTDs/dtd.html>

4 THE COMMUNICATION LAYER

The role of the Communication Layer is to act as a communicator and a data integrator between the Clarity services and the interface. The Communication Layer consists of three main parts: (1) the *Web services*, modelled (i.e. wrapped) around the basic services of the Application Layer, (2) the *Web service clients* which are used to request and receive information from the Web services, and (3) the *Clarity session manager* which serves as an information broker between the Clarity web services and the interface.

4.1 THE CLARITY WEB SERVICES

A *Web service* is an application that is accessible over the Internet via standardised protocols such as HTTP, SMTP, etc. and its public interface (i.e. methods, arguments and return values) is described in XML. One method to codify the process of sending XML messages between the different services is the Simple Object Access Protocol (SOAP) (W3C 2002b) (the other is XML-RPC). SOAP is an XML-based protocol for data exchange between computers by representing both data and remote method invocation requests in the same XML document. SOAP has been developed as a platform and language independent protocol to enable client applications to access remote services as remote procedure calls.

Clarity follows the standard paradigm of Web service architectures with service providers (i.e. the web services), service requestors (i.e. the service clients) and a registry of services for documentation and technical support (albeit not provided as a public registry). The services in the Clarity Application Layer have been modelled as SOAP services that are accessed by SOAP clients in the Communication Layer. A typical SOAP interaction consists of a SOAP request by the client and a SOAP response by the service. The client request must include the name of the method to be invoked along with any parameters. The client response contains the answer to the initial request. A simplified SOAP interaction for translating a query from English to Finnish in Clarity is shown in the example of Figure 2.

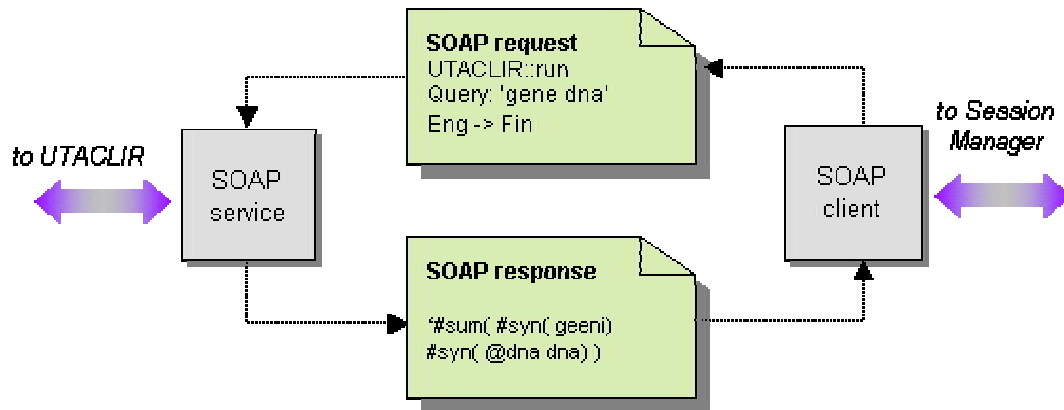


Figure 2: Example of SOAP interaction for query translation in CLARITY.

Both the request and the response are encoded in XML using the SOAP specification for data typing and exchange. The SOAP request specifies the method to be invoked (UTACLIR::run) and the method arguments i.e. source language (Eng), the target language (Fin), and the query ('gene dna'). Once the XML request is parsed at the server end and the request is forwarded to the backend translation application, the result (i.e. the translated query) is collected and a new XML message is formulated by SOAP as a response to the original request. This is sent back to the client service where it is parsed and its content is fed to the Session Manager for further use.

4.2 THE CLARITY SESSION MANAGER

The *Clarity Session Manager* is in the heart of the Communication Layer and caters for all of Clarity's data communication and interoperability requirements. The Session Manager is a *middleware* that communicates with both the SOAP services and the Clarity interface. 'Middleware' is taken here with the loose definition of 'software that performs specific operations to integrate disparate system components in order to act as a single application to another system'.

The Session Manager consists of four component modules:

1. **Service Coordinator:** The role of this module is the planning, coordination and completion of tasks expressed as interface requests. Planning refers to the sequencing of necessary actions to accomplish a particular task. The list of main tasks for the Clarity scenario interactions includes: *query translation*, *document retrieval*, *concept hierarchies*, and *multi-document reports with filtering functionality*. Each of these tasks is mapped to a series of actions and actions for tasks of different types can be combined in order to achieve a collective effect. For example, the document retrieval task implies that query translation must be done first because the translated query needs to be passed as a parameter to the retrieval service. Coordination refers to the synergy of steps for completing a task such as the routing of transactions to the appropriate Clarity services, the collection of data from these transactions, the integration of the data from different services, the storage of data as temporary files in the local file system,

and so on. Completion refers to the actual task implementation and the generation of expected responses to the interface requests.

2. **Interface Content Generator:** This component serves to generate the necessary data for the user interface. Due to the fact that the interaction between the Clarity Session Manager and the user interface is stateless, specialised indices of document meta-data are used to communicate the data to the interface. The indexed meta-data include all the information the interface might need such as the document language, the document title and its translation, the query terms found in the document, the location of the document body in the file system etc.
3. **Local Services Component:** for efficiency reasons a very small number of services of the Application Layer have been 'cloned' in the Communication Layer; this is due to the fact that methods that use limited resources but are used repetitively, such as the normalisation and translation of terms in documents, may cause significant bottlenecks due to the high number of transactions required over HTTP for a single task². These services are intended to be replaced by Web services in the near future.
4. **XML Generator:** the information generated during a Clarity interaction is stored locally in XML; XML serves to provide a uniform representation of data collected during a CLIR session.

The Session Manager documentation is accessible at:

https://www.dcs.shef.ac.uk/research/groups/nlp/clarity/partners_only/clarity_sess_man_doc.html

² As an example, for realistic sizes of documents sets thousands of terms may need to be morphologically analysed, translated and transported over HTTP per request.

5 THE INTERFACE LAYER

The Interface Layer includes the Clarity interface. The interface component reads the meta-data index generated by the Session Manager and dynamically reformats the corresponding information to HTML for display on a web browser.

The design of the Clarity interface was user-centred and is described in detail in Petrelli et al (2002). The two main user tasks are 'Translate' and 'Search'. 'Translate' refers to the query translation process. For each query term, the users are presented with all possible translations in the specified target languages and they can make selections of the translations they think they best represent the source language terms. 'Search' refers to the actual document retrieval task. This task results in a ranked list of document titles with additional document information so that the user can browse the list first and then select a document she/he would like to read, print, or save it to a file for later use.

Documents are retrieved and displayed in batches of 10 so that the retrieval is time efficient, and the user is not overwhelmed with the amount of document information on screen. Of course, the interface allows for the retrieval of the next batch of documents, or the backtracking to the previous batch, and so on. The document information presented for the 'Search' task is:

- Document language
- Document length
- Document rank
- Translated title
- Source and target query terms occurred in this document
- Best passage

Users can also choose two additional functions 'Overview' and 'Report'. 'Overview' is the generation of concept hierarchies from the retrieved document collection whereas 'Report' refers to multi-document reports with filtering functionality. Both functions have been designed to provide extra browsing and filtering capabilities in order to help the user understand the content for foreign language documents and get guidance to the more relevant parts of the document collection.

The Clarity interface is available at <http://clarity.shef.ac.uk> under 'Demos'.

The documentation for the latest interface version can be always found via <http://clarity.shef.ac.uk> under 'Project documentation' and then 'Technical'.

The URL for the documentation of interface version S2 is:

https://www.dcs.shef.ac.uk/research/groups/nlp/clarity/partners_only/clarityCGIvS2_doc.html

6 SYSTEM PRACTICALITIES

Since many of the variety of information access tasks in which information systems in general can be engaged are highly interactive, the users of the systems can sometimes be expected to have a limited tolerance with respect to time responses to their queries. Clarity was intended from the beginning to be a CLIR system that could support real experiments by users without resorting to techniques such as ‘canned’ translated queries, offline document retrieval or ‘simulated’ interactions. As it often happens with practical systems like Clarity, our experience during the development stage revealed that, although the architecture can support large scale experiments, modifications to the access of some services could provide speed improvements that might greatly enhance Clarity’s usability.

While the two main tasks of query translation and document retrieval are accessed only once and cause no significant delay during a user interaction (i.e. during a single ‘search’), a number of associated auxiliary services may have to be accessed repetitively for some tasks. In order to minimise bottlenecks and optimise system performance, three strategies were employed.

The first strategy was text preprocessing. For example, the translation of document titles involved the generation of indexes of pre-translated titles for each translation route (e.g. Finnish to English, Swedish to Finnish etc.) in advance.

The second strategy was to group a number of related services into the same Web service. For example, the retrieval of document bodies, the translated titles, the query terms found in each the document and the best passages were grouped as a single service.

The third strategy was to ‘clone’ services in order to take advantage of local database management or other software facilities instead of accessing them by SOAP. Such ‘cloned’ services include the normalisation and translation of terms (not query translation) for the concept hierarchies and services that provide term frequency information. These facilities are accessed as local services to the Session Manager.

7 PERFORMANCE EVALUATION

The focus of this section is not on the effectiveness of Clarity in terms of precision and recall, but rather, in terms of response times with respect to parameters such task completion, number of documents, number of languages involved and multiple users. To evaluate Clarity's speed of processing at several stages, we measured the time taken for the system to complete the different functions necessary for a CLIR interaction. The experiments were performed on a Pentium III 700Mhz PC with 1Gb of RAM running Linux OS under the normal multi-user, multi-tasking workload in a University research lab.

The functions were grouped together into the following main tasks:

- A. Query translation (SOAP service).
- B. Task A plus: document retrieval, title translation, identification and highlighting of target language query terms in documents and extraction of best passages (SOAP service).
- C. Task B plus: identification of source language terms in documents and generation of content indices for the user interface (implemented at Session Manager level).

Note that task (B) includes task (A) and task (C) includes both (A) and (B).

The average length in words of the queries set for this experiment was 6.25 and document retrieval was performed for one target language (Finnish) with English as the source language. The queries used for the experiments were based on four iCLEF topics as described in Petrelli et al (2002). The breakdown of the times for these tasks is provided in Table 2.

Number of documents retrieved	Clarity response times (secs)		
	Task A	Task B	Task C
10	2	3	4
20	2	5	7
50	2	8	11
100	2	15	18
150	2	23	26
200	2	29	32

Table 2: Clarity's response times with respect to particular subtasks.

As can be seen from Table 2, the time required for query translation (A) is constant for any number of texts retrieved as is to be expected. Clarity's document retrieval times

(B) are tolerable for up to 100 documents. For more than 100 documents the system gets slower but the response times are not overly excessive. Benchmarking tests revealed that most of the time is spent on the retrieval of translated titles, the identification and highlighting of query terms and the file management processes for the web services and less on the retrieval of documents by InQuery. The generation of interface indexes (C) by the Session manager takes only a fraction of the whole time and seems to be more or less constant with respect to document retrieval (B).

In a second experiment we investigated the impact of the number of target languages on the response times for different data sets. The languages used for this experiment were English, Finnish and Swedish and the number of retrieved documents was equally distributed among all target languages. The results are presented in Figure 5 which illustrates that the addition of more languages has an effect on system performance.

The relative increase in the response time for two target languages compared to one target language averages 65% and for three languages compared to two languages averages 33%. Take into account that for any added target language the Web services have to do one more query translation and invoke different operations for document retrieval and identification of query terms. Why the addition of a third language adds

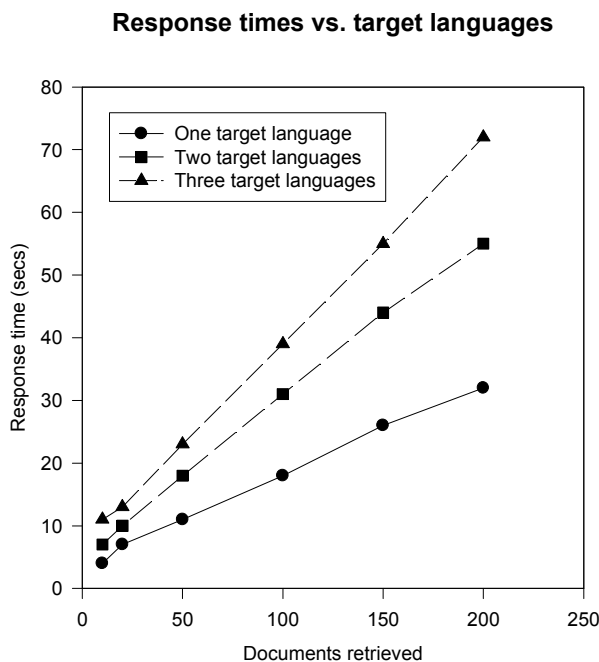


Figure 3: Clarity's response times with respect to the number of target languages.

less time relative to the addition of a second language, can be explained by the fact that one of the three languages currently happens also to be the source language in which case the query and the titles do not have to be translated.

In a third experiment we investigated the impact of concurrency due to multiple client requests running on the system at the same time. This experiment was designed to test

the effectiveness of Clarity for multi-user and multi-tasking usage and each client request simulated a different user interaction. The results are shown in Figure 6.

It can be seen that the number of concurrent clients has an impact on system performance which becomes greater as the number of clients increases. Apart from the extra workload required from more clients, part of this time degradation is due to peculiarities to the particular SOAP toolkit used for implementing the services which operates in a First-In-First-Out (FIFO) manner.

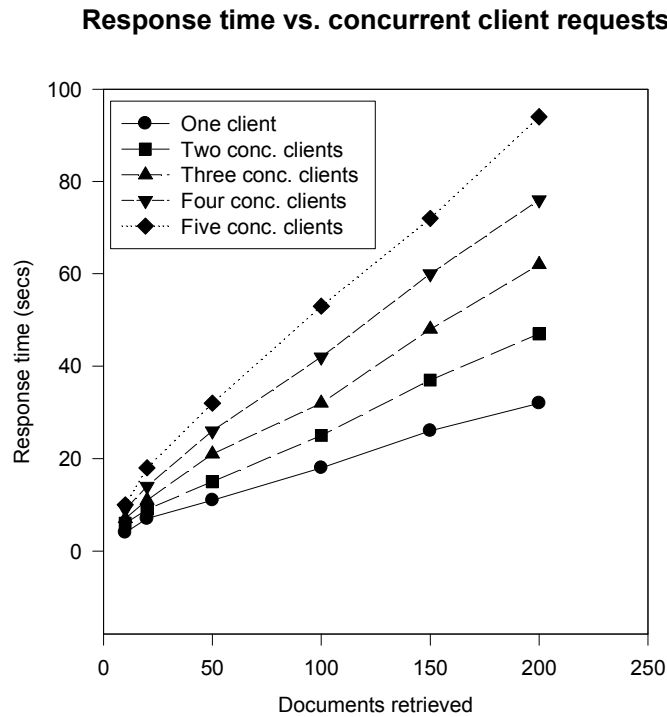


Figure 4: Clarity's response times with respect to the number of concurrent client requests.

Current work is underway for optimising the Clarity's performance and new statistics about the system response will be provided in the final report.

8 REFERENCES

Callan J. P., Croft W. B. and Harding S. M. (1992). The INQUERY retrieval system. In Proceedings of the 3rd International Conference on Database and Expert System Applications, Valencia, Spain, September 1992.

Fluhr C., Shmit D., Ortet, P., Elkateb, F., Gurtner, K., and Semenova, V. (1996). Distributed multilingual information retrieval, MULSAIC Workshop, ECAI96 Conference, Budapest, 12-16 August.

Hansen, P. and Karlgren J. In press. "Effects of Foreign Language and Task Scenario on Relevance Assessment". Submitted to special issue of IP&M on Cross-Language Information Retrieval.

Hayashi Y., Kikui G., and Susaki S. (1997). TITAN: Cross-linguistic Search Engine for the WWW, Working Notes of AAAI Spring Symposium on Cross-Language Text and Speech Retrieval, Stanford, CA, 58-65.

Hedlund, T., Airio, E., Keskustalo, H., Lehtokangas, R., Pirkola, A., Järvelin, K. (2003) Dictionary-Based Cross-Language Information Retrieval: Learning Experiences from CLEF 2000-2002. Accepted for publication in Information Retrieval.

Karlgren J. (1999), Stylistic Experiments in Information Retrieval. In: Natural Language Information Retrieval. Tomek Strzalkowski, (ed.), Kluwer.

Koskenniemi K. (1983). Two-level Model for Morphological Analysis. In Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 8-12 August 1983, Karlsruhe, 683-685.

Petrelli, D., Beaulieu, M., Sanderson, M., Demetriou, G., & Herring, P. (2002). Is Query Translation a Distinct Task from Search? Proceedings of the 2002 CLEF Workshop.

Oard, D. (2000). Evaluating Interactive Cross-Lingual Information Retrieval: Document Selection. Presented to first CLEF Workshop, September 2000. Lisbon.

Pirkola A., Keskustalo H., Leppänen E., Käsälä, A-P., Järvelin K., (2002). Targeted s-gram matching: a novel n-gram matching technique for cross- and monolingual word form variants. Information Research, 7 (2).

Radwan K., Foussier F. and Fluhr C. (1991). Multilingual access to textual databases. Proceedings of RIAO'91, Intelligent Text and Image Handling, Barcelona, April 2-5, 475-489.

Sanderson, M. and Croft, W.B. (1999). Deriving concept hierarchies from text. In Proceedings of the 22nd ACM SIGIR Conference, 206-213.

Wilkinson R., Wu Mingfang and Fuller M. (2001): Searcher Performance in Question Answering. Proceedings of SIGIR 2001, 375-383

World Wide Web Consortium – W3C (2002a). Web Services Activity. At: <http://www.w3.org/2002/ws/>.

World Wide Web Consortium – W3C (2002b). SOAP Version 1.2. At: <http://www.w3.org/TR/3/REC-soap12-part0-20030624/>.