ERCIM-99-W001
SICS

European Research Consortium
for Informatics and Mathematics

ERCIM

Eighth DELOS Workshop

User Interfaces in Digital Libraries

Stockholm, Sweden, 21-23 October 1999

CONTENTS

Introduction

The 8th DELOS Workshop on User Interfaces in Digital Libraries was held in Stockholm, Sweden, 21-23 October 1998. The DELOS Working Group is an action of the ERCIM Digital Library Initiative (http://www.area.pi.cnr.it/ErcimDL/). During the workshop, a "mini-workshop" was held, together with participants from the 4th ERCIM UI4All Workshop, another ERCIM Working Group, that was held between the 19-21 October 1998. 21 participants, including guest speakers attended the workshop. 11 presentations were made during the workshop.

The DELOS working group is funded by the ESPRIT Long Term Research Programme and has been promoted by ERCIM with the objective of supporting research in areas related to digital libraries. In particular the DELOS objectives are

- to stimulate research activities in areas which are relevant for the efficient and cost-effective development of digital library systems,
- to encourage collaboration between research teams working in the field of digital libraries and
- to establish links with on-going projects and activities in the field of digital libraries in industry and other public and private institutions.

A Digital Library is the integration of several different components and will include a range of content and services. It will also include a large and diverse group of users. It is important to develop an understanding of the overall tasks and interactions users are engaged in when entering a Digital Library. We interact constantly with our environment through different communication mechanisms and processes. *Information seeking and retrieval* in Digital Libraries is but a special case of such a process. Analysis and evaluation of user, systems and interactions are needed to successfully build future Digital Libraries. Information retrieval research tends to abstract away from both the general aspects of interaction and view actions at the interface as isolated events, and from the special requirement information access tasks pose on interface design.

Presentations

The workshop started with a "Challenge paper" (Preben Hansen and Jussi Karlgren, SICS). The paper summarized some important research issues raised 25 years ago related to information retrieval (IR) and user interfaces (UI) and its relevance for the workshop. The authors found that some of the questions raised then are still valid today, such as the characteristics of the user, the task, the information content and medium, the computer and IR techniques and the role of evaluation and feedback in the redesign cycle, among others. However, there has also emerged new research areas such as multimedia content, multimodal interaction, multilingual information and users and distributed systems and collections.

Our guest speaker, Professor Nicholas Belkin, Rutgers University, provided a "road-map" on important issues for Digital Libraries in his paper "Understanding and supporting Multiple Information Seeking Behaviours in a Single Interface Framework". First, Belkin presented a definition of a Digital Library and what functions need to be supported in such a framework. Based on the knowledge that people engage in multiple information seeking strategies (ISS) and multiple types of interactions with information within an information-seeking episode, Nick Belkin described his and his group's work within the third TIPSTER research program. The goal of the project is to identify and classify different ISSs, characterize sequential structures of ISSs, identify specific combinations of IR techniques appropriate for different ISSs, and construct and evaluate system which adapts to support different ISSs in the course of a single information seeking episode

Constantine Stephanidis, ICS-FORTH, our second guest speaker, raised some critical issues for interaction design in digital libraries in the light of HCI and Digital Libraries. Among the main issues and challenges mentioned were diverse user groups, variety in the context of use, and technological proliferation. The author also proposed a way to deal with the design of Digital Libraries containing 3 phases: Design processes and techniques, Implementation, and Evaluation.

The first two presentations focused on multilingual aspects. Nuno Miguel Antunes Freire, INESC, described a digital library project containing theses and dissertations, based on the LDAP technology, including Multilingual Classification Systems to allow a cross-language information retrieval service. Aarno Jarno Tenni (VTT) described the use of controlled languages for query translation in a legislative document retrieval system in the paper *Automatic Translation in Multilingual User View to Legislative Databases*.

Two papers addressed the issue of evaluation of information systems. The paper "*WWW Interface Design, Driven by Heuristic Evaluation: The EINS-Web Project* presented by Silvana Mangiaracina, CNR, describe the experience of the evaluation and design of the EINS-Web user interface, using heuristic evaluation. Demosthenes Akoumianakis, ICS-FORTH, discussed principles for constructing user interfaces as multiple metaphor environments.

Kuldar Taveter (VTT) presented an agent-based system of semantical information retrieval IR system based on concepts and domains in his paper *Intelligent Information Retrieval Based on Interconnected Concepts and Classes of Retrieval Domains*. Maria Francesca Costabile (Bari University) discussed the issue of visualization and describe a technique for visualizing meta-information in the paper *Information Visualization in the Interaction with Digital Libraries.*

Three papers concerned the implementation of user interfaces. Two of them were based on the Dienst System. *Implementing a Common User Interface for a Digital Library: the ETRDL experience* presented by Donatella Castelli (CNR) discussed the design decisions and experiences underlying the design of the interface. *AQUA: An advanced user interface for the Dienst digital library system* presented by László Kovács (MTA SZTAKI). Finally, based on empirically defined stylistics based genres and clustering, a interactive information retrieval interface with multi-dimensional presentation of search results which was developed in a joint project between SICS and Telia Research, was presented by Johan Dewe and Niklas Wolkert, now from Netsolutions AB.

Some important issues discussed during the workshop:

1. *Information seeking and retrieval*. One important issue in Digital Library research is issues related to HCI and distributed information seeking and retrieval (ISR). We should consider information seeking and retrieval as embedded activities within Digital Libraries.
2. *Evaluation*. Currently, a lot of different applications are being developed and used. We need techniques and methods to analyse, and evaluate different systems as well as different users, their behaviour, and tasks, when interacting with distributed information resources. This also includes developing new techniques and methods. It was also recognized that it was important to evaluate the ideas behind the systems developed.
3. *Support for interaction*. We need to support interactions with information, such as texts and multimedia in information seeking activities. This include access to multilingual information. It is also important to construct conceptual models and theories in order to understand the interactions in distributed Digital Libraries.
4. *Modalities*. Future Digital Libraries will encompass alternative modalities for representations of information seeking activities.

5. *Integration*. Integrating the user interface closely with the functionalities of the system is desirable: usefulness cannot be added after the functional design.

We take this opportunity to thank the participants of the workshop and especially those who made presentations, for the many interesting discussions that took place and for making the workshop such a success. We also thank the ERCIM office for their valuable contribution to the organization of the workshop and the publication of these proceedings.

Preben Hansen and jussi Karlgren
Swedish Institute of Computer Science, Human-Computer Interaction and Language Engineering Lab, Sweden.

Interaction and Interactivity:
User interfaces for Digital Libraries.

A Challenge paper for the 8th DELOS Workshop,
Stockholm, Sweden, 21-23 October, 1998.

Preben Hansen and Jussi Karlgren
SICS

INTRODUCTION

We interact constantly with our environment through different communication mechanisms and processes. *Information seeking and retrieval* in Digital Libraries is but a special case of such a process. Information retrieval research tends to abstract away from both the general aspects of interaction and view actions at the interface as isolated events, and from the special requirements information access tasks pose on interface design.

Many questions from early information retrieval research are still valid and no less important today. In a paper from 1971, Bennett outlines some parameters concerning interactive information retrieval and interface design:

- The characteristics of the searcher
- The conceptual framework presented to the searcher
- The role of feedback
- Operational characteristics such as command language, display, response etc.
- The constraints of the computer and IR techniques
- The effect of the IR system on the user interface for search
- Introduction of search facilities to the user
- The role of evaluation and feedback in the redesign cycle.

In a later paper (1972), he adds:

- The task to be performed,
- The user, and
- The information content

to the properties of information system design needs to take into account. This list has not aged.

CHALLENGE QUESTIONS

Challenge question 1:

What have we learnt about interactive information retrieval in the past 25 years?

There are new and emerging research areas that have bearing on our field, that in part did not exist twenty-five years ago:

- Multimedia content
- Multimodal interaction
- Multilingual material and users
- Distributed systems
- Theoretical frameworks, tools, and methods to analyse users' tasks and work context and to understand the dynamic and changing nature of tasks over time

- The move from static to dynamic evaluation and to having a balanced approach to evaluation (both system and user)
- The move from laboratory evaluation to beta testing by real users and customers
- The relationship between evaluation and design, that is, what implications the analysis and evaluation of user preferences, strategies, and tasks will have on user interface design

Challenge question 2:
What new aspects of interactive information retrieval have appeared in the past 25 years?

Challenge question 3:
What will we learn about interactive information retrieval in the next 25 years?

REFERENCES

Bennett, J. L. (1971). Interactive bibliographic search as a challenge to interface design. In: D. E. Walker (ed.) *Interactive bibliographic search: The User/Computer Interface*. Pp. 1-16

Bennett, J. L. (1972). The user interface in interactive systems. *ARIST*, 7, pp. 159-196.

Understanding and Supporting Multiple Information Seeking Behaviors in a Single Interface Framework

Nicholas J. Belkin
School of Communication, Information & Library Studies
Rutgers University
New Brunswick NJ 08901-1071 USA
nick@belkin.rutgers.edu

(From slide presentation)

What is a Digital Library?

An *institution* which performs and/or supports (at least) the *functions* of a *library* in the context of *distributed, networked collections* of information objects in *digital* form.

Functions of a Library

- Selecting
- Collecting
- Preserving
- Organizing
- Representing
- Providing access to
- Ensuring knowledge of
- Disseminating

Information objects
- Mediating between
- Supporting interaction between

Information users and (collections of) information objects

Structure of Today's Talk

- Description of our approach to Information Retrieval (IR)
- Description of our TIPSTER Phase III project
- Some results on classification of interactions with information
- Support for four "prototypical" interactions
- Some results from our TREC-7 studies

Project Hypotheses

- People engage in multiple information seeking strategies (ISSs) and multiple types of interaction with information during the course of a single information seeking episode
- Different combinations of IR techniques will be appropriate for supporting different ISSs

Goals of the Project

- Identify and classify the interactions with information that people (i.e. a group or community) engage in
- Develop individual systems each of which supports one of a range of such behaviors and test them
- Integrate the different systems within a single system and interface framework which supports graceful movement from one kind of interaction to another, and test it

In Other Words

- Identify and classify a range of different ISSs
- Characterize sequential structures of ISSs
- Identify specific combinations of IR techniques appropriate for different ISSs
- Construct and evaluate system which adapts to support different ISSs in the course of a single information seeking episode

To Accomplish Our Goals

- Study a group of information-intensive workers in their normal task environments; content analysis and classification of interactions with information
- Choose representative range of interactions; stipulate system designs to support each; test each against baseline system
- Integrate different support designs within common, object-oriented framework; implement common interface; test against baseline system

Underpinnings of our Approach

- People engage in many kinds of interactions with information, in single and across multiple information-seeking episodes
- Each kind of interaction requires a different kind of support, but support for all should be provided in a single system
- Each kind of support can be understood as a different combination of the different ways to implement the different IR processes
- An object-oriented framework provides a structure which allows multiple combinations of techniques

Project Theory

- Dimensions of ISSs
- IR as support for interaction with information
- Structured information interaction
- Combining from classes of IR techniques to develop specific systems for support of specific ISSs

What's Been Done

- Specific community has been studied; faceted classification of interactions with information has been developed; characteristics of information-seeking interactions have been identified
- Four prototypical kinds of interactions with information related to information-seeking identified; functional specification of support for each designed
- One support system implemented, with preliminary evaluation; one support system partially implemented
- O-O framework implemented, subsequently discarded

The Problem

- Design an information system which supports different information seeking behaviors
- Identifying and characterizing information seeking behaviors
- Identifying and testing techniques for supporting those behaviors

The Approach

- Identify a group of people who interact with information as a part of their normal task environment
- Observe and record their interactions with information, in relation to their tasks, goals and intentions

Classifying Information Behavior

- Interview information-intensive workers in task environments about their information behaviors
- Content analysis of interviews to identify
  - Resources interacted with
  - Tasks leading to interactions
  - Intentions of interactions
  - Types of interactions
  - Dimensions of interactions

The Approach

- Analyse the data to identify common dimensions of interaction with information
- Analyse the data to relate tasks, goals and intentions to specific information behaviors
- Analyse the data to relate system functionalities to information behaviors
- Analyse the data to identify patterns of information behaviors

The Methods

- Last stab:
  - at beginning of work day, give subjects an “activity notes” log to fill out during day
  - at close of day, interview on today’s tasks and activities and on other normal tasks and activities
- Pilot results:
  - good data, not too much work, not too obtrusive. USE IT!!

Subjects

- 14 managers, engineers, technical staff at various parts of Boeing, Seattle
- Opportunistic sample, selected with both variety and enthusiasm in mind
- 3 female, 11 male

Subject Speciality Distribution

- Sysadmin
- Team leader
  - Web infrastructure
  - Web standards and use
  - Technology assess. & development
- Web infrastructure
- Web design
- Technical writer
- Strategy formulation
- IS support manager (2)
- Manager
  - enterprise architecture deployment
  - technical communication
  - technical library
  - technology assessment

Data

- Job descriptions
- Activity notes

- Transcriptions of audiotaped interviews (1 1/2 - 2 hours/interview)

Data Analysis

- Initial identification of tasks which subjects perform in accordance with their positions and job descriptions
- Content analysis of transcribed interviews, interpretation aided by activity notes forms

Interviews Coded According to

- activity/resource interacted with
- task
- intention
- type of interaction
  - communication
  - information
- common dimensions of interaction type
- kinds of interaction
- dimensions specific to kinds of interaction
- criteria for interaction

Communication Behaviors

- Classified according to
  - Medium
    - speech, text, ...
  - Mode
    - face-to-face, mediated, ...
  - Mapping
    - one-to-one, one-to-many, many-to-many

Information Behaviors

- Create
- Disseminate
- Organize
- Preserve
- Access
- Evaluate
- Comprehend (e.g. read, listen)
- Modify
- Use (e.g. interpret)

Objects Interacted With

- Level (e.g. information, meta-information)
- Medium (e.g. image, written text, speech)
- Quantity (e.g. one object, set of objects, database of objects)

Common Dimensions of Information Interactions

- Information Object
  part -- whole
- Systematicity
  random -- systematic
- Degree
  selective – exhaustive

Criteria of Interaction

- time
- topic
- person
- importance
- alphabet
- authority
- accuracy
- and so on

Dimensions of Access

- method
  scan -- search
- mode
  recognize – specify

Prototypical Information Interactions

- Finding a known(?) information object
- Recognizing useful information objects by scanning through an information resource
- Evaluating the usefulness of information objects
- Determining the content / structure of a set or collection of information objects

Known Item Support

- Query by form (or example)
- Best, or partial (e.g. soundex) match (or range) on all fields
- Data fusion

- Field indexing
- Query reformulation support

Evaluating Information Items

- Scrollable displays of information objects
- Query-based summaries or focused display of information objects
- Clustered organization
- Saving and labelling

Learning Contents of Resources

- Labelled a priori classes with numbers of information objects in class
- Labelled ad hoc query-based classes
- Dynamic reclassification
- Examples

Finding Useful Items by Recognition

- Minimal unstructured query
- Automatic query expansion
- Classification of retrieval results
- Label clusters, display at levels of hierarchy
- Interaction with clusters and objects
- Summarize information objects
- Save and label objects
- Relevance feedback (+ & -)

Experimental Protocol

- Four experimental systems, one designed to support each type of information interaction
- Baseline system, best current generic IR system
- Tasks designed for each information interaction type
- Within subjects design, for each system, compare performance in baseline and experimental system

Baseline System

- Unstructured query, minimal syntax
- Stop word / stop phrase
- Field specification
- Term suggestion device
- Best match (based on InQuery)
- Positive and negative relevance feedback as term suggestion

System to Support Finding Useful Items in Unfamiliar Domain by Recognition

- Carried out within the TREC-7 Interactive Track
- Task is: Given a general topic area, find documents that identify the different aspects, instances, points of view with respect to that topic; save one document which represents each different aspect,...
- Point is to find and save one document per aspect, *not* all of the documents
- Example: “What are the different ways to treat high blood pressure?” Saving one document that lists them all satisfies the task; saving ten documents, each of which discusses one of the ten different ways, does too.

Preliminary Results for inq-R

- Relevance feedback (positive) as term selection device understood as useful and usable for the task
- No difference in performance between positive only vs positive + negative relevance feedback conditions
- Relevance feedback supports exploration
- Relevance feedback supports recognition
- Labelling supports task
- Best passage feature supports recognition
- MORE TO BE REPORTED AT TREC-7

Interaction Design in Digital Libraries : Some critical issues

Constantine Stephanidis

Foundation for Research and Technology-Hellas (FORTH)
Institute of Computer Science (ICS)
Science and Technology Park of Crete
Heraklion, Crete
GREECE

(From slide presentation)

HCI and Digital Libraries

- critical technological fields in the emerging Information Society
- identification of synergies towards a common, international R&D agenda

Main issues

- how can they be designed, implemented and evaluated
- what functional & non-functional quality attributes need to be addressed
- how can diversity be accounted for
- what software components are needed
- what architectural models need to be followed
- etc.

Characteristics of DLs

- Distributed (across the Internet)
- Large volumes of data
- Multimedia content
- New virtualities

→ new range of computer-mediated human activities

HCI challenges in the context of DLs

- Diverse user groups
  - orientation
  - navigation
  - information overload, etc.

- Technological proliferation
  - terminals (portables, network attachable equipment)
  - novel input/output devises (e.g., wearables)
  - new interaction platforms (e.g., Java)
  - etc.

- Variety in the context of use
  - desktop versus nomadic use
  - user tasks in the DL domain
  - physical and social environment

Meeting the challenges

- Designing for the broadest possible end-user population
  - studying the dimensions of diversity
    - users with different abilities, requirements and preferences
    - context of use
    - terminals, novel platforms and network-attachable devices
  - context and intend
    - alternative styles for instantiating user tasks

        - conveying context through assigning interaction objects to dialogue states
- Shifting the focus of implementation
    - specification-based framework versus programming
    - tools for developing interactive software

Recent contributions from ICS-FORTH (AT&HCI Lab)

- Unified user interface development method
    - Comprehensive methodology for integrating universal accessibility and interaction quality as part of the user interface development life-cycle
- Unified user interface development platform
    - USE-IT: Design environments
    - PIM: Platform Integration module
    - G-DISPEC: 4G Specification language
    - I-GET: Integrated Development Environment
- HCI International ’97 Tutorial
  http://www.ics.forth.gr/proj/at-hci/html/tutorials.html

Common themes (from the ECDL ’98 Proceedings)

- User interface adaptation
- User interface agents
- User modelling components
- Metaphors, including visualisations
- Virtual reality
- Multilinguality

- Information retrieval
- Multimodality
- Evaluation

A perspective

Phases:

- design
- implementation
- evaluation

Phases (1/3)

- Design processes & techniques
  - Human-centred design (ISO 13407) fosters
    - usability focus
    - iterative evaluation-feedback loops
    - techniques to attain the above
  - Is “usability” (as approached today) enough ?
  - What about quality (functional & non-functional attributes) ?
  - Do existing UCD techniques cope with design pluralism ?
  - Do existing UCD techniques offer process-oriented support ?
  - e.g. unfolding, capturing and maintaining design rationale

Phases (2/3)

- Implementation
  - Approaches

- programming
- specifications

- Tool requirements
  - support for collaboration
  - inter-operability
  - sharing knowledge and experiences, etc.
  - ➲ What is the role of software architectures ?

Phases (3/3)

- Evaluation
  - ➲ Does evaluation ever lead to innovative designs, or does it simply help identify design defects ?
  - ➲ How can one evaluate some of the non-functional quality attributes which are critical to DLs ?
    - inter-operability
    - modifiability
    - reusability
    - portability
    - scalability

Our focus

- Bridging across the two communities
- Awareness raising
- Exchange of experience and establishment of common ground
- Developing a common vocabulary
- Focusing on a common research agenda

- EC 5th Framework Programme
- ERCIM WGs
- Joint session of the ERCIM Delos Workshop and ERCIM WG UI4ALL (Stockholm, 21 October 1998)
- White Paper - “HCI in DLs: An International R&D Agenda”

Possible questions

1. Is “usability” (as approached today) enough ?
2. What about quality (functional & non-functional attributes) ?
3. Do existing UCD techniques cope with design pluralism ?
4. Do existing UCD techniques offer process-oriented support ?
   e.g. unfolding, capturing and maintaining design rationale
5. What is the role of software architectures ?
6. Does evaluation ever lead to innovative designs or does it simply help identify design defects ?
7. How can one evaluate some of the non-functional quality attributes which are critical to DLs ?

Integration of Multilingual Classification Systems with the Dienst digital library system

Nuno Freire

Instituto de Engenharia de Sistemas e Computadores (INESC)
Rua Alves Redol, 9, 1000 Lisboa, Portugal
Email: Nuno.Freire@inesc.pt

Abstract

This document aims to provide a presentation of the current stage of the digital theses and dissertations project. This project started in the beginning of August 1998 and comprises the processing of thesis and dissertations. We are currently working in the integration of Multilingual Classification Systems with the Dienst digital library system. One of the key functions of the classified space will be the information space normalization and to allow a cross-language information retrieval service.

INTRODUCTION

We intend to promote a digital circuit in co-operation with the university libraries to cover the digital processing of theses and dissertations. A national trial for an optional process for digital versions, parallel to the actual paper-based one, is under development with the collaboration of a group of universities.

This project has the following objectives:

- To improve graduate education by allowing students to produce electronic documents, use digital libraries, and understand issues in publishing
- To increase the availability of student research for scholars and to preserve it electronically
- To lower the cost of submitting and handling theses and dissertations
- To empower students to convey a richer message through the use of multimedia and hypermedia technologies
- To empower universities to unlock their information resources
- To advance digital library technology

A customised network of servers based in the DIENST technology and running in Unix machines will support the system. The system is known to work on Linux and Solaris platforms, although any system running Apache web server and a PERL interpreter should support it.

Client-side requirements are a typical "web browser".

Local DIENST servers will be installed at the local university libraries, from which the documents are captured to the central server using HTTP.

We are currently trying to solve the problem of correct document classification. Usually, the librarian classifies the document but, if we want to obtain a correct document classification, the classification should be performed by the author. This is the reason why we are integrating the classification systems with Dienst and creating user interfaces to allow both the author and the user, to submit, browse and search the collection, using the classification schema.

THE CLASSIFICATION SYSTEMS

We use the existing classified systems stored in LDAP due to its hierarchical and distributed proprieties. This space will be used to normalise users and information spaces and to solve multi-language information retrieval.

1.1 LDAP interface (Distributed directory)

All information about the classified systems is stored in a directory consisting of an X.500 directory, using an LDAP implementation. This decision allows a fast and hierarchical access to this information.

X.500 is a standard directory service that defines an information model, a namespace, a functional model and also an authentication framework. An X.500 directory is based on entries, which are collections of attributes as defined in RFC 1779 [1]. Each entry has a type (or class), typically defined by one or more mnemonic strings, and can have one or more values. The attributes required and allowed in an entry are controlled by a special object class attribute in every entry. The information is supposed to be structured in a tree, accessible by servers possibly distributed over a network.

X.500 defines the Directory Access Protocol (DAP) to access the service, a full, complex and heavy OSI protocol supporting operations in three areas: search/read, modify and authenticate. The search is possible at any level, based in a filter query involving attributes and returning requested attributes from each matching query.

The problem of the excessive complexity of the DAP protocol has been addressed by the Network Working Group of IETF, that has been proposing the Lightweight Directory Access Protocol (LDAP) as an alternative for the Internet. LDAP is a client-server protocol that runs directly over TCP/IP and was conceived to remove some of the burden of X.500 access from directory clients, such as taking out some of the less-often-used service controls and security features.

LDAP is being positioned as the directory standard for the Internet, with leading industry players like Microsoft, Netscape, IBM, Lotus, Novell and Banyan supporting it or intending to support it in the near future [2]. There are also plans to develop LDAP access for several database and index machines, such as Glimpse.

LDAP stores this information and we built interfaces to browse easily classification. We propose a LDAP implementation on a Linux machine with a freeware version offered by this University of Michigan [3]. This LDAP implementation has three main components:

- *Server*: We will run our server on a Linux machine as a stand-alone daemon. However, to provide more flexibility and fault tolerance, it will be distributed and replicated by other servers within the National academic networks (a feature supported by LDAP).
- *Client library*: a powerful C language API for accessing and using LDAP, with LDAP clients and a backend to handle database operations. With these tools we will build a user interface to browse easily the classified space and give also an interface for administration of this space.
- *Gateway*: a special web interface is available for directory and server administration.

The document-classified space is stored as shown in figure 1. At the top level there are entries representing the available classification systems, in the next levels there are entries representing general terms, and so on. At the lowest level there are entries representing subject descriptions. Each entry contains a unique identifier and a term in English, Portuguese and, in the future, translations to other languages.

Access to perform writing inputs is only given to Human authority. Users can only browse this space.

This classified space is important for normalisation purposes.

1.2 Cross-language information retrieval service

The existence of classified systems in different languages will allow a cross-language information retrieval service. When the document is classified, the system indexes the terms and the corresponding identifiers. So, if the user searches for relevant documents using the classification systems to formulate his query, the system will retrieve all documents with the selected classifications. The documents are retrieved independently from the language in witch they were classified. The system searches for the selected terms and respective identifiers.

In this project we use English and Portuguese for the document classification systems, but this framework is designed for an easy extension to other languages as well. With this tool we can provide a information retrieval service for differed languages only with the requirement of having classified systems translated to different languages.

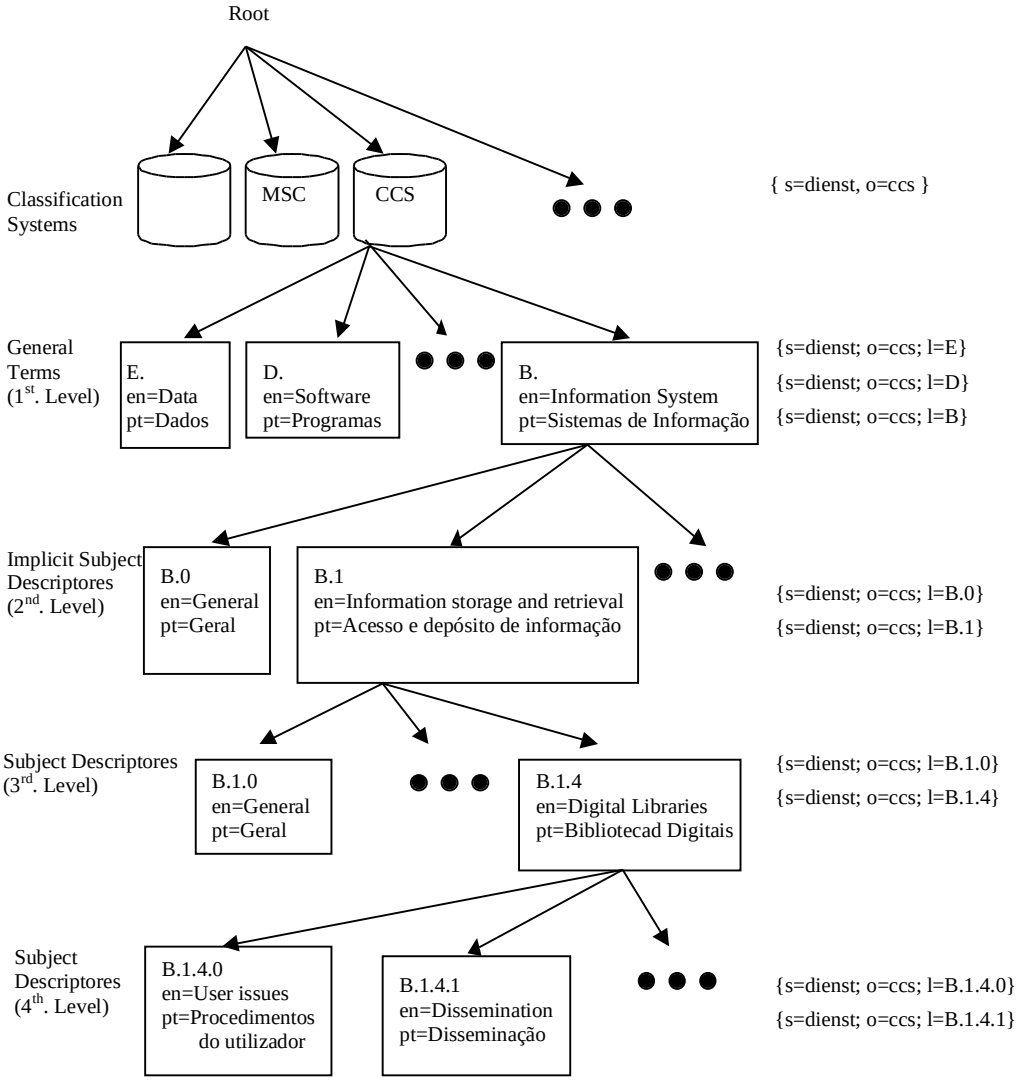


Figure 1: Classified spaces are stored and maintained in a LDAP space.

SYSTEM ARCHITECTURE

A user can access the system with one of two purposes: to submit a new document to the collection, or to search/browse in the collection. In both the situations it is possible to take advantage of the Classification Server.

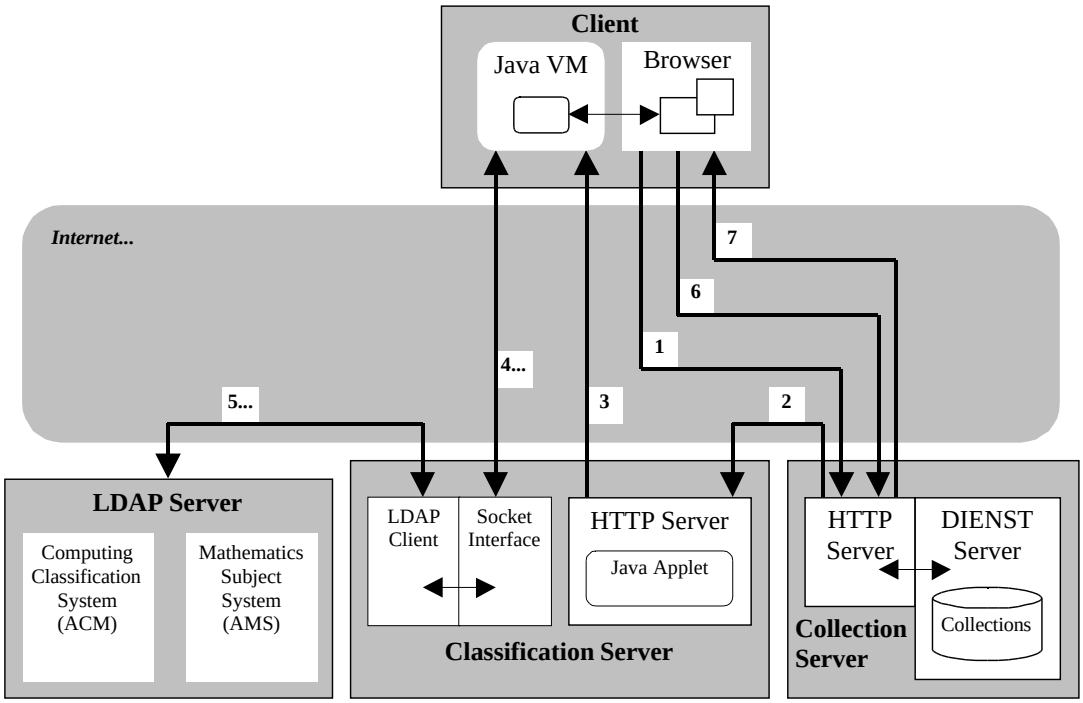


Figure 2 : Integration of Multilingual Classification Systems with Dienst

An interaction starts with a contact to the Collection Server, by HTTP, where the desired operation is selected (1).

If the Classification Server is requested, that request is transferred to it (2), which replies to the user sending him/her an applet (3).

With this applet the user can browse the classification systems available from the Classification Server (4). Those classification systems are stored in a X.500 directory, accessible by LDAP (5). The LDAP server used in this system is provided by the University of Michigan.

Actually we have available from the Classification Server the Computing Classification System from ACM, and the Mathematics Subject System from AMS, both in English and partially translated to Portuguese.

The directory was conceived to host also other structured classification systems, as also their translations in multiple languages. On the other side, the Java applet is completely independent of the contents of the directory, being configured according the information received from it.

In the interaction with the Classification Server the user can select the desired terms, in one or more the languages, and return to the Collection Server with those selections (6) to be used in the task in course (7).

Depending of the actual task, the selected terms can be used in the Collection Server in a pre-coordination purpose, to classify a new document, or in a post-coordination purpose to search in the indexes. The actual Classification Server can maintain several indexes, depending of the metadata structure of the collection. In the actual system, we support one index for each classification schema, but the terms included in those indexes are also used as generic keywords and indexed in the keywords' index (in this sense, a user can perform a free searching in that index using terms from the classification systems and find the right documents in the same way).

CONCLUSION

Using the classification systems to improve the document's metadata allows better results in information retrieval. But the classification should be made by the author, to assure a correct document classification. In this paper we have described a user interface for document classification and retrieval using the available classification systems. The classification systems are stored in LDAP directories. These directories are accessed by HTTP, through a java applet, where the user can select the terms from the classification systems, either for browsing/searching, or document classification.

The cross-language information retrieval service is achieved by including the terms from the classification systems and their respective identifiers in the document's metadata. Using the identifiers, it is possible to find documents classified in any language because the identifiers are common to all languages.

REFERENCES

[1] Cooper,J.;Ratcliffe,N. The role of LDAP and X.500. Data Connection, August 1996. *Available on-line in* 3 July 1998 *at* *http://www.datcon.co.uk/docs/press/mdwhite1.htm*

[2] Howes,T.; Smith, M. **LDAP Programming Directory-enabled Applications with Lightweight Directory Access Protocol.** Macmillan Technology Series (1997)

[3] Howes, T;Smith,M. (1995). **RFC1823: The LDAP Application Program Interface**. IETF Network Working Group, August 1995. *Available on-line in* 3 July 1998 *at* http://ds.internic.net/rfc/rfc1823.txt

Automatic Translation in Cross-Lingual Access to Legislative Databases

Catherine Bounsaythip, Aarno Lehtola, Jarno Tenni
VTT Information Technology P. Box 1201, FIN-02044 VTT, Finland
Phone: +358 9 456 5957. Fax: +358 9 456 6027.
Email: {catherine.bounsaythip, aarno.lehtola, jarno.tenni}@vtt.fi
http://www.vtt.fi/tte

Abstract:
This paper considers the use of controlled languages for query translation in a legislative document retrieval system. Problem statement and analysis of the approach are described. The use of controlled languages is motivated by the fact that precision is very important in our case. In many information retrieval systems, the use of unrestricted language resources such as general purpose machine translation or bilingual lexica, provides better recall at the expense of precision. Ambiguities and polysemy make the search engine retrieve irrelevant documents as semantic knowledge is missing. Controlled languages help to better specify the word sense according to the domain of interest. Thus ambiguities are avoided and polysemy is specified according to the domain. We will implement our idea in the area of VAT regulation in Europe.

Introduction

Nowadays, information on European legislation and intergovernmental agreements are scattered in distributed repositories in heterogeneous formats and in many languages. This information is technically accessible through information networks, but it is extremely difficult even for professionals to use it because of differences in document structures and languages. This is a common problem in cross-lingual information retrieval (IR) systems where queries are made in one language to a document collection in several different languages and the goal is to retrieve only those documents relevant to the query. Before retrievals can be performed, deep linguistic analysis and translation of the query appears necessary.

Natural language processing in IR systems is special in the sense that a pattern of term occurrences in a document generally suffices to determine the subject matter; as word order is largely irrelevant. Because of ambiguities and polysemy, query translation is not a trivial task. One way to ensure the performance of the system is to control the query construction. This approach is discussed in this paper where we present our machine translation software called Webtran.

Webtran is a machine translation system for controlled languages (CL) to be embedded in WWW-based information service systems (Lehtola et al. 1998a). It is designed to support fully automatic translation in online WWW services, such as online mail order catalogue or information retrieval from cross-lingual databases. The framework in which Webtran is involved, consists of an interface through which the user can make queries in one language to search for legislative texts from different EU databases of EU regulatory information. In **Figure 1**, we show the basic architecture of the system. As existing repositories are located in different countries and stored in different formats, it is necessary to convert the request into the formats of the targeted databases.

The user can make queries in his own language and his request is translated by Webtran Translator into the language of the target documents before being directed to the multilingual databases. Retrieved documents will be displayed in their original language. In the domain of

legislation, usually users prefer to have the texts in their original language so that the interpretation is more reliable. Moreover, it is usually out of question to translate the whole text in an automatic way, as some legal terms have different meanings according to country and according to the area of laws. In any case, translations of legal texts need to be authorised to avoid misinterpretations. Therefore, only the query terms will be handled by the Webtran translator. Possibly, Webtran can provide approximate translation of some meta-information related to the document (e.g., headers, titles, summary or keywords).

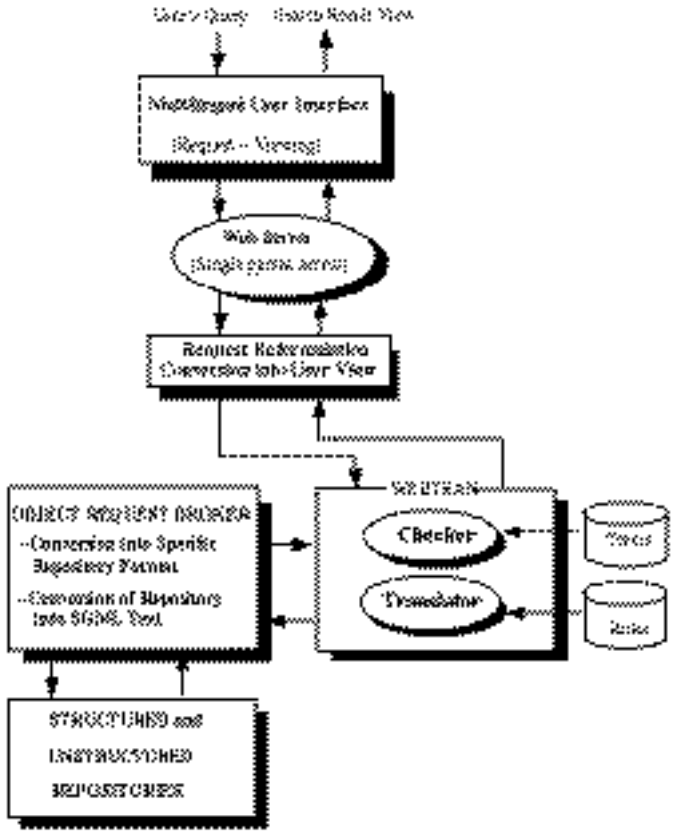


Figure 1 *: A cross-lingual IR architecture for accessing and viewing EU legislative databases.*

Cross-Lingual Information Retrieval

Language technology is important in cross-lingual document retrieval systems. In TITAN system (Kikui et al. 1996), the language processor contains language identifier (English / Japanese) and bilingual dictionaries. The user can make requests in Japanese or in English and the URLs found are displayed with their headers translated into the query language. In EMIR (Fluhr et al. 1996), SYSTRAN is used in the language processing part of the retrieving system.

Translation of queries and keywords does not need just multilingual machine-readable dictionaries as many ambiguous terms and polysemy may appear. Many approaches have been used, such as interlingua (Landauer 1990), alignment of large parallel text corpora in different languages (Davis and Dunning 1995), concept-based (Chen 1993) and controlled vocabulary (Soergel 1997).

In (Landauer 1990), an approach for fully automatic cross-language document retrieval was presented. Their system is based on a language-independent representation where no humanly constructed dictionary, thesaurus, or term bank are needed. The construction of the interlingua

is based on a statistical method using paragraph alignment of a sample collection of parallel texts. This is done once for a subject area. Each word in the sample is then assigned a vector value determined by the total pattern of usage of all the words in all the sample paragraphs. In the second step, a new document or query in any of the original languages is assigned a vector value that is an average of the values of the words it contains. Tests on a French-English corpus showed that the method works well, because the two languages are quite close to each other. This wouldn't work for example between Finnish and Swedish.

The approach to query translation in multilingual IR systems in (Davis and Dunning 1995) used evolutionary programming to optimise the construction of a query from bilingual dictionaries. The assignment of term weights is done by means of a population of potential weighting schemes to generate translated queries. Sentence-level alignments from a large parallel text collections were used to evaluate the correctness of a query translation. The approach is based on the consideration that translated queries are primarily derived by a mapping from a word set in the query language to a word set in the language of the derived query. They reported good results for the case where the original query is closely related to the document collection. Results are unclear for queries that are not closely related to the documents. Moreover evolutionary optimisation for discovering optimal queries using a parallel training corpus takes too much time for ``on-line" IR systems.

In (Chen 1993), the system was based on concept exploration. Concepts are extracted from the keywords used in the set of user-selected documents and Genetic Algorithm (GA) was used to perform concept optimisation. The optimisation is based on the relevance of each document to other documents in the user-selected set. A document which included more concepts shared by other documents had a higher score. The optimised chromosome contained relevant keywords which best described the initial set of documents. Then, the optimised concepts are put into a Hopfield Network to activate other relevant concepts, e.g., when the user selected a new document. The new keyword was then used to identify more relevant documents and the GA/HP process continued.

For performance and simplicity, many systems avoid sophisticated linguistic analysis of the documents by imposing a specialised ``controlled language" (Oard 1997). In (Soergel 1997), a multilingual thesaurus is built to relate the selected terms from each language to a common set of language_independent concept identifiers, and document selection is based on concept identifier matching. The user is assisted for specifying from a semantic field the term that best describes his intended meaning.

Webtran IR Approach

For low-cost services of the access to the legislative databases through WWW search engines, it is necessary that fully automatic translation achieves a reasonable performance. To do so, the approach adopted by Webtran is based on controlled vocabulary. This would help to relate terms from each language to a common set of language dependent concept identifiers. By the word concept we mean in this paper interrelated items in a conceptual model, that have been defined by humans for the target domain. At the language level a concept can be expressed by a term and its synonyms which can be single words or longer surface expressions. Term is the most obvious or most widely agreed expression of the synonyms. Then there can be semantically close expressions that are not accurate but approximately reflect the meaning of the term.

For example, the official term for the concept of "*avoiding payroll tax*" in Finnish is "ennakonpidätysvelvollisuudesta vapauttaminen", and one way of expressing it can be "ennakonpidätyksen välttäminen". Expressions of a term in different languages can also be

viewed as synonyms. **Table 1** shows an example of Finnish and Swedish surface expressions of a concept.

Finnish: "ennakonpidätysvelvollisuudesta vapauttaminen"
Swedish: "befrielse från skyldighet att verkställa förskottsinnehållning"
English (approx.): *acquitting from the responsibility of paying payroll tax.*

Table 1 : *Example of surface expressions of a concept as used in legislation.*

The use of concepts for query translation can enhance the retrieval performance. For example, an inexperienced user would likely make a request in Finnish about *avoiding payroll tax* to a Swedish database with: "ennakonpidätyksen välttäminen". The actual official term used in legal texts is: "ennakonpidätysvelvollisuudesta vapauttaminen". With a plain Boolean operator, the search may fail if the translation of "välttäminen" does not match in the target database. If semantically related words of the domain are not considered, precision of the retrieval is decreased. Kekäläinen and Järvelin (1998) have shown that expansion of queries into concepts and synonyms yielded better retrieval scores.

To achieve that, the system requires experts in legislation to define the conceptual models and relationships to surface expressions in the covered languages. This will be used for helping the construction of queries in a controlled way. The creation conceptual models can be done by analysing the existing repositories to create dictionaries of common elements or by aligning parallel texts in different EU languages.

Besides, for the end-users, it is not easy to find the proper term for making a request in the legislation domain and especially from foreign text databases. For instance, if the document is in Swedish, the system should help the user in finding correct search terminology by providing an automatic translation of search sentences from the user's native language to Swedish. If the documentation were not available in Swedish, the system should assist in translating the search terms provided by the users to proper search terms in the local language of the document database.

A help system will be developed to assist the user in defining the proper search term. In such a situation, a multilingual thesaurus can be used. One word chosen by the user can trigger inference of new words by the conceptual model. The help interface should be easy enough so that the user is not required to be trained in order to effectively select proper search terms and to exploit thesaurus relationships. These observations indicate that the user interface must be designed to adapt to the needs of each category of users (see, e.g., Lehtola et al. 1998b).

A user interface for developing controlled languages can be found in (Lehtola et al. 1998a). An IR user interface will be based on a WWW browser as the example shown in **Figure 2**. In this sketchy example, a click on button "HAE" would send commands to the translation component of Webtran. It is transparent to the user. The click of "APU" would trigger the opening of the help interface. This interface would share some functionality of the user interface built for controlled language designer described in (Lehtola et al. 1998a).

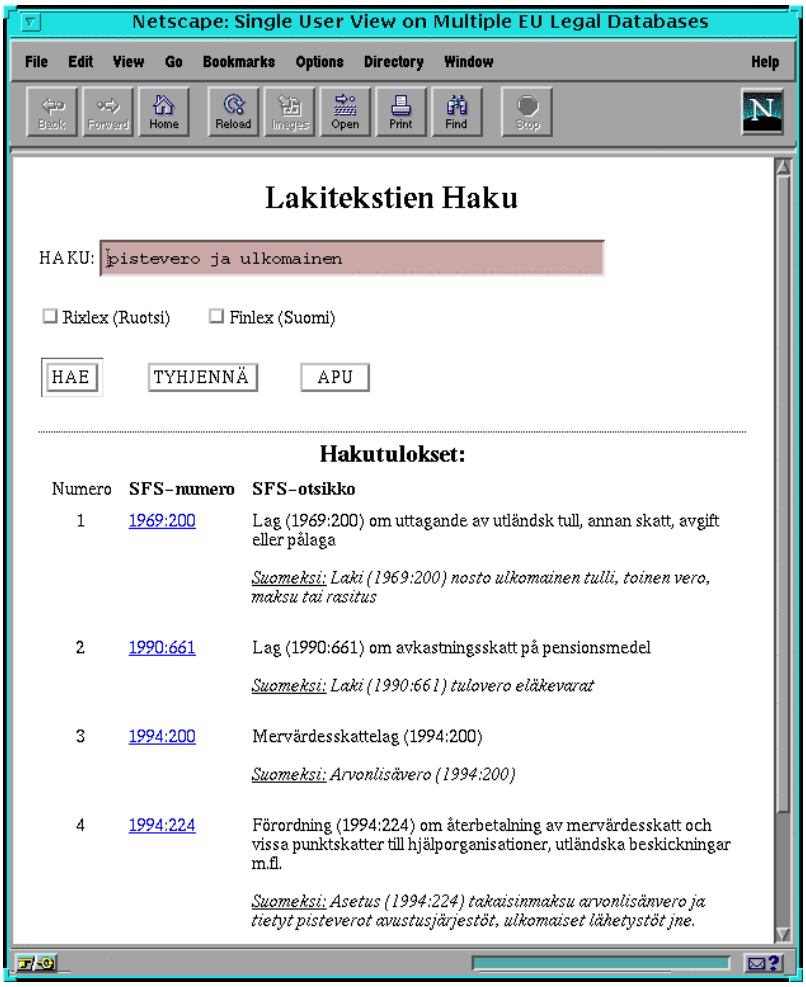


Figure 2 *: Sketch of a WWW-based cross-lingual information access interface. In this illustrative example, the query is made in Finnish to Swedish legislation databases.*

Conclusion

In this paper, we have described our ongoing work about using controlled languages for cross-lingual access to legislative databases. As a term may have different meanings in different areas of laws, controlled language should be designed for a specific domain of law. Prototype system will focus on VAT regulation texts from different European countries. We are now on the phase of gathering corpora in this domain in order to build controlled languages in Finnish and Swedish.

Acknowledgements

The authors would like to thank the Technical Development Centre of Finland (TEKES), Tieto Corporation Ltd., and Ellos Ltd., all from Finland, for supporting our work in many ways. Also, many thanks to Prof. Seppo Linnainmaa, Prof. Timo Honkela and, Kuldar Taveter for their useful comments on this paper.

References

Chen, H. (1994). GANNET: Information Retrieval Using Genetic Algorithms and Neural Nets. Working Paper, Center for Management of Information, College of Business and Public

Administration, University of Arizona, CMI-WPS.
http://ai.bpa.arizona.edu/papers/gannet93/gannet93.html

Davis, M. W., Dunning ,T. E. (1995), Query Translation Using Evolutionary Programming for Multi-lingual Information Retrieval, *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, San Diego, CA, March.

Fluhr, C. Schmit, D. Ortet, P. Elkateb, F. Gurtner, K. (1996). Distributed Multilingual Information Retrieval, MULSAIC'96, Multilingual in Software Engineering: AI Contribution.
http://www.iit.nrcps.ariadne-t.gr/ costass/mulsaic.html

Kekäläinen, J., Järvelin, K. (1998). The Impact of Query Structure and Query Expansion on Retrieval Performance. In: Croft, W. B. & Moffat, A. & van Rijsbergen, C.J. & Wilkinson, R. & Zobel, J. (Eds.), *Proc. the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (ACM SIGIR '98), Melbourne, Australia, August 23-28, 1998. New York, NY: ACM Press.

Kikui, G., Hayashi, Y. and Suzaki, S. (1996). Cross-Lingual Information Retrieval on the WWW, MULSAIC'96, Multilingual in Software Engineering: AI.
Paper: http://www.iit.nrcps.ariadne-t.gr/ costass/mulsaic.html.
TITAN URL: http://sting.navi.ntt.co.jp/titan/titan-e.html

Lehtola, A., Tenni, J., Bounsaythip, C. (1998a). Definition of a Controlled Language Based on Augmented Lexical Entries. *Proceedings of the Controlled Language Applications Workshop 98*, Carnegie Mellon, Pittsburg, USA, 21-22 May, 1998, pp. 16-29.

Lehtola, A., Tenni, J., Bounsaythip, C. (1998b). Controlled Language Technology in Multilingual User Interfaces. To appear in *Proceedings of the 4th ERCIM Workshop User Interface for All (UI4All)*, Special Theme: "Towards an Accessible Web", Stockholm, Sweden, 19-21 October, 1998.

Landauer, T. K. and Littman, M.L. (1990). Fully Automatic Cross-Language Document Retrieval Using Latent Semantic Indexing. *Proceedings of the Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research*, pages 31-38.
UW Centre for the New OED and Text Research, Waterloo Ontario, October.
http://www.cs.duke.edu/~mlittman/docs/refer.html

Oard, D. (1997). Alternative Approaches for Cross-Language Text Retrieval. *AAAI Symposium on Crosslanguage and speech retrieval*, March 24-26.
http://www.ee.umd.edu/medlab/mlir/

Soergel, D. (1997). Multilingual Thesauri in Cross-Language Text and Speech Retrieval", *AAAI Symposium on Crosslanguage and speech retrieval*, March 24-26.
http://www.ee.umd.edu/medlab/mlir/

Intelligent Information Retrieval Based on Interconnected Concepts and Classes of Retrieval Domains

Kuldar Taveter, VTT Information Technology, kuldar.taveter@vtt.fi

Keywords: information seeking behaviour, visualisation, metaknowledge

1. Introduction

The structure of information in any information source represents concepts and inter-concept relationships of the domain in question that the creator of the information source had in his mind. On the other hand, people often express their information needs in terms of concepts that information is needed about. The *conceptualization* of the world embodied in some information source may be in the form of a database schema, or it may also be some classification that the information in the source is based on. One of the most important problems that has to be solved in semantical information retrieval from heterogeneous sources is to reconcile different conceptualizations of the world represented by different information sources [1]. A part of this problem is that different information sources make use of different classifications of the same objects of the world. One way to tackle this is to distinguish the ontological aspects of objects (i. e. the conditions of their being, structure, integrity, identity) from the taxonomic aspects (i. e. the conditions for seeing them as members of one or another particular class) [3]. This is the approach we have chosen in SARI, which is an agent-based system of semantical information retrieval that is being jointly worked out by VTT Information Technology, Tampere University of Technology, and Tampere University.

2. Ontology vs. taxonomy

According to Guarino et al [2] *ontology* can be understood as an intensional semantic structure which encodes the implicit rules constraining the structure of a piece of reality. Ontologies are thus aimed at answering the question "What kinds of objects exist in one or another domain of the real world and how are they interrelated?". Ontologies can be made explicit by forming a logical theory which gives an explicit and partial account of the above-mentioned intensional semantic structure. Such logical theory contains concepts, their definitions, and relationships between them like e. g. subsumption (inheritance) and aggregation. Ontologies contain concepts of two kinds: *types* and *roles*. The basic difference between them is that the former are semantically rigid, i. e. their instances are necessarily such that they always belong to them, while this is not the case for the latter. For example, a plant will be a plant during its whole lifetime, while a student can cease to be a student and still remain the same individual [9].

In [3] it is claimed that ontologies should be separated from taxonomies because since there exist several different ways to classify the same objects in concurrent taxonomies, the objects must be independent from these taxonomies. In other words: objects that belong to a certain concept can be classified in very different ways depending on the *viewpoint*. For example, genes in biology can be classified differently from functional, chemical, and evolutionary viewpoints. We are now trying to bring this claim to a firmer ontological basis by further claiming that each of the concurrent taxonomies classifies objects of some concept according to a particular *role* that is subsumed by this concept, and represented by the taxonomy's root class. For example, the taxonomies under the classes "Commodity" and "Product" in Figure 1 classify certain man-made objects (artifacts) according to the roles that they respectively play in the domains of foreign trade and manufacturing. Both of the mentioned classes are subsumed by the concept "Artifact" of the ontology by the Role-Of relationship (v. Figure 1).

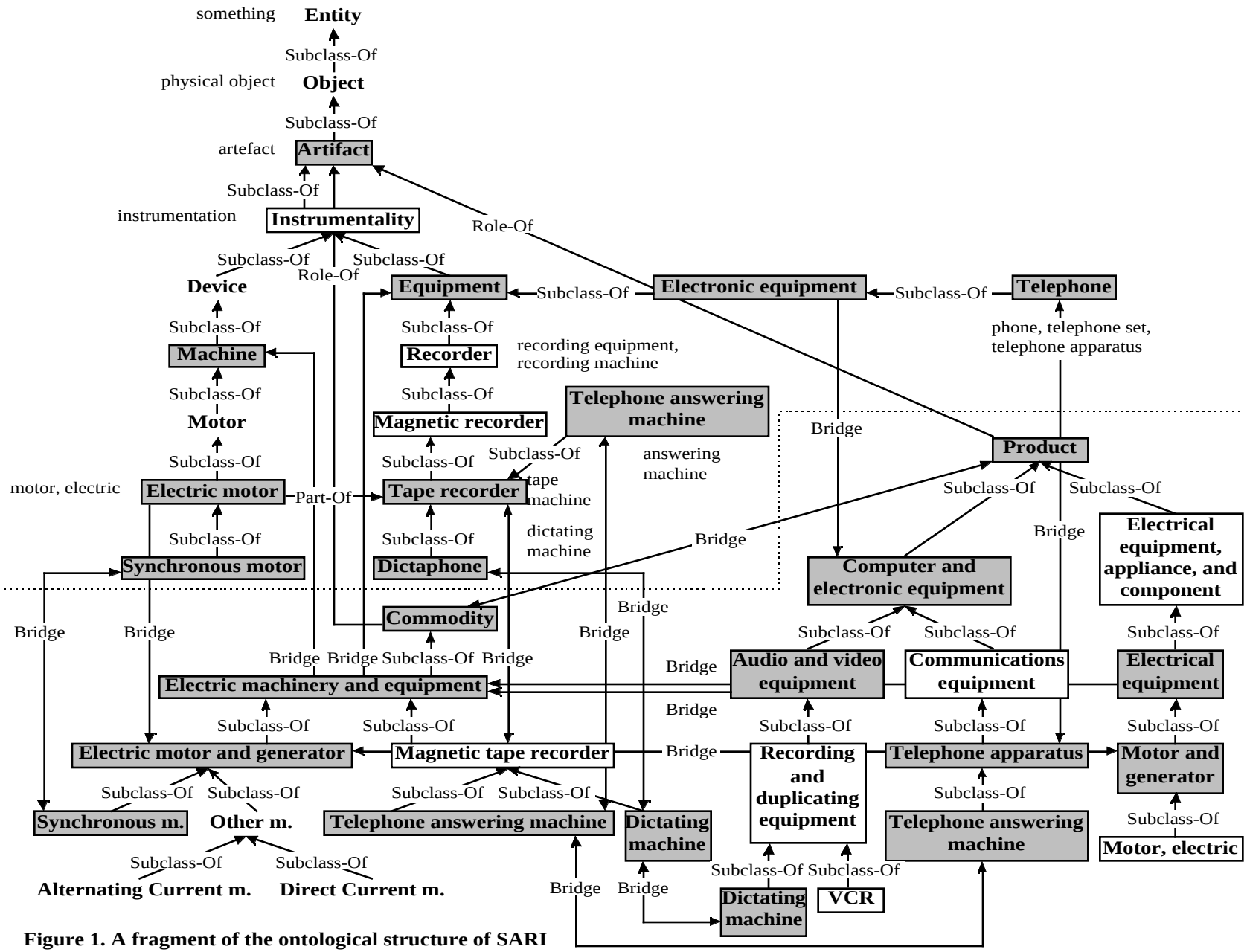


Figure 1. A fragment of the ontological structure of SARI

3. Viewpoints and bridges in information retrieval

Since a hierarchy of classes under which the instances of a concept (objects) can be classified is determined by a particular viewpoint, each role corresponds to a viewpoint. Consequently, we can say that the part of Figure 1 below the dotted line depicts the classifications (taxonomies) of certain artifacts under the *foreign trade viewpoint* (left) and *manufacturing viewpoint* (right).

In order to represent links between the class structures of different taxonomies, we make use of the notion of bridge. A *bridge* between two classes under different viewpoints means that an object which is a member of the source class under one viewpoint is also a member of the destination class under the other viewpoint [4]. In Figure 1 there is a bridge between the classes "Commodity" and "Product" which are respectively the root classes under the foreign trade and manufacturing viewpoints. While in [3] and [4] bridges and viewpoints are used for creating classifications of real world objects, we use them to describe already existing classifications that are conceptualized by information sources. For example, the classification under the class "Commodity" in Figure 1 is an excerpt from the standardized CN hierarchy of commodity types [5] which is used by the statistical database Ultika of Finnish foreign trade, and the classification under the class "Product" in the same figure is a subset of the NAICS classification of industry [6] used in North America. Bridges can be divided into one-way bridges and two-way bridges. We say that there is a *one-way bridge* between two classes under different viewpoints when all possible member objects of the source class also belong to the destination class. There is a *two-way bridge* between two classes when this is true in both ways, i. e. when the sets of possible extensions[1] of the two classes are equal. For example, there is a one-way bridge between the concepts "Electrical equipment" and "Electric machinery and equipment" under the viewpoints of manufacturing and foreign trade, respectively, because all possible instances of the first class also belong to the second class. On the other hand, since both the class "Electric motor and generator" under the foreign trade viewpoint and the class "Motor and generator" under the manufacturing viewpoint represent the set of *electric* motors and generators, there is a two-way bridge between them. Please note also that there is *no* bridge between the classes "Magnetic tape recorder" and "Recording and duplicating equipment" under the respective viewpoints of foreign trade and manufacturing because:

- while the first class involves telephone answering machines, the second class doesn't;
- while the second class involves video cassette recorders (VCRs), the first class doesn't.

The concepts that are involved in any bridge relationship are marked grey in the figure. The only restriction on forming bridges is that a bridge should not bring about inheritance contradictions between the class hierarchies of different viewpoints, i. e. bridges should not cross each other.

Let us now consider a task where the user wants to query Ultika and some database using NAICS in parallel which is quite a realistic situation. The possibility to browse in parallel the conceptual structures of both databases and have the needed queries generated makes the process of information retrieval cognitively natural and easy for the user. While browsing the user is able to switch between viewpoints at different locations of the databases' conceptual structures using bridges.

We also make use of the bridge relationships to link the concepts of the ontology to the corresponding classes of the taxonomies. Incorporating parallel conceptual structures into the *ontological structure* consisting of the ontology and its taxonomies considerably speeds up parallel browsing. For example, by using bridges, the user can immediately proceed from the browsing of the ontology's concept "Electric motor" to the browsing of the corresponding classes "Electric motor and generator" and "Motor and generator" of

[1] The extension of a class is any set of its individuals (objects, occurrences, instances).

the taxonomies of commodities and products, respectively. In our example an excerpt from the ontology that both of the taxonomies refine is depicted in the part of Figure 1 above the dotted line. In order to retain legibility of Figure 1, only some bridges between the concepts of the ontology and the classes of the taxonomy of products are represented there.

4. The use of terms and synonyms

The ontology presently used by us is based on WordNet [10]. The ontology represents concepts and relationships between them. In the figure subsumption (Subclass-Of) and aggregation (Part-Of) relationships are depicted. Each concept of the ontology is represented by a natural language expression called *term* which is the text in the corresponding rectangle, and possibly by one or more *synonyms* which are given at the side of the rectangle. The term is the most typical or obvious of the synonyms. The classes of the taxonomies in Figure 1 are also represented by their terms. Synonyms play an important role in concept-based query formulation and expansion [7], and in replying to free-form queries [8]. For example, a user of SARI who is looking for information about dictaphones can start off by entering or having generated with the help of the ontological structure the query "dictaphone" which is thereafter expanded by the synonym of the term "Dictaphone" to the query "'dictaphone' OR 'dictating machine'". The synonym "dictating machine" matches with the terms of the corresponding classes of both taxonomies. In the same manner, a user who enters the query "phone" will eventually be taken to the class "Telephone apparatus" under the manufacturing viewpoint because both "phone" and "telephone apparatus" are synonyms for the term "Telephone" of the ontology. In SARI all queries are implicitly processed as case-insensitive.

5. Conclusions and future work

The main contributions of our work are the following:

- further elaboration and clarification of the distinction and connections between ontology and taxonomy by utilizing the notion of role;
- the use of the notions of viewpoint and bridge in information retrieval;
- the coinage of the notions of one-way bridge and two-way bridge to be used for connecting the conceptual structures of different information sources;
- the use of bridges for connecting the concepts of an ontology to the classes of its taxonomies, resulting in the ontological structure.

The ontological structure consisting of the ontology and the taxonomies refining its concepts can be applied to:

- query expansion and generation for structured (e. g. relational, hierarchical, OO) databases, and for browsing their conceptual structures;
- query expansion for textual databases;
- query expansion for WWW.

The most important problem that remains to be solved in our future work is how to make the formation of bridges semiautomatic. In principle this can be done by successive comparing each of the ontology's terms and its expanding synonyms with the terms of its taxonomies. The main obstacles lie in the computational complexity of such a task on one hand, and in the abundance of different grammatical forms in which the terms and synonyms can be expressed. This is especially true for agglutinative languages like e. g. Finnish and Estonian. Our future work will also include the formalization of the distinction between ontology and its taxonomies using roles.

The results of our research work will be implemented in the third pilot of the SARI system.

6. References

1. F. Saltor, E. Rodríguez. On Intelligent Access to Heterogeneous Information. Proceedings of the 4th KBRD Workshop. Athens, Greece, August 1997.
2. N. Guarino, P. Giaretta. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In: N. J. I. Mars (ed.), Towards Very Large Knowledge Bases, IOS Press 1995, pp. 25-32.
3. J. Euzenat. On a purely taxonomic and descriptive meaning for classes. Proceedings of the IJCAI workshop on "Object-based representation systems". Chambéry, France, 1993.
4. J. Euzenat. Brief overview of T-tree: the Tropes Taxonomy building Tool. In: Philip Smith, Clare Beghtol, Raya Fidel, Barbara Kwasnik (eds.), Advances in Classification Research 4, Learning Information, Medford (NJ US), 1994.
5. Combined Nomenclature (CN). Commission Regulation (EC) No 3115/94 of 20 December 1994. Official Journal of the European Communities L 345, 31 December 1994.
6. See http://www.census.gov/epcd/www/naics.html
7. J. Kekäläinen, K. Järvelin. The impact of query structure and query expansion on retrieval performance. Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, August 24 - 28, 1998.
8. J. Kristensen. Expanding end-users query statements for free text searching with a search-aid thesaurus. Information Processing & Management, 29(6): 733_744.
9. N. Guarino, C. Masolo, G. Vetere. OntoSeek: Using Large Linguistic Ontologies for Gathering Information Resources from the Web. LADSEB-CNR Technical Report 01/98, March 1998. Submitted for publication.
10. See http://www.cogsci.princeton.edu/~wn/.

WWW Interface Design, Driven by Heuristic Evaluation: The EINS-Web Project

S. Mangiaracina

Consiglio Nazionale delle Ricerche
Area della Ricerca di Bologna

mangiaracina@area.bo.cnr.it

P.G. Marchetti

European Space Agency – ESRIN
Informatics Department

pmarchet@esrin.esa.it

Abstract

This paper describes the experience of the evaluation and design of the EINS-Web user interface. EINS-Web allows the access to distributed collections of bibliographic and textual databases, together with a seamless interaction with the whole World Wide Web on Internet. The heuristic evaluation of the Graphical User Interface (GUI) was run in two steps. The results of the evaluation of the first version of the GUI were used to drive the design of the Web version. This paper describes the methodology used and the lessons learned. The interaction among evaluators and designers proved to improve the success of a spiral design methodology, which is needed to cope with the requirements of designing interfaces targeted at the rapidly evolving Internet world.

1. Introduction

Four different approaches can be used to perform GUI evaluation: formal (by means of technical analysis), automatic (by means of ad hoc software tools), empirical (by means of experiments together with user) and heuristic (judgments and opinions stated after the interaction with the GUI) [1]. We did select the latter methodology, because the heuristic evaluation is largely independent of the software and it is proactive, allowing picking up easily the suggestions and adopting them during the design and its review. During the EINS Web project we decided to drive the interface design by means of the heuristic evaluation, using a spiral design approach.

Heuristic evaluation is performed by looking at an interface and trying to come up with an opinion about what is good and bad about the interface. Ideally, people would conduct such evaluations according to certain rules, such as those listed in typical guideline documents. Most people perform their own «heuristic evaluation» on the basis of their intuition and common sense instead.

In order to evaluate a GUI that allows access to collections of bibliographical and textual databases, an information problem has to be formulated and opinions on how the interface supports the user throughout the satisfaction of his/her information needs have to be assessed.

2. The evaluation process

EINS-Web allows accessing distributed collections of bibliographic and textual databases, together with a seamless interaction with the whole World Wide Web on Internet (see Figure 1 for an overview of the architecture).

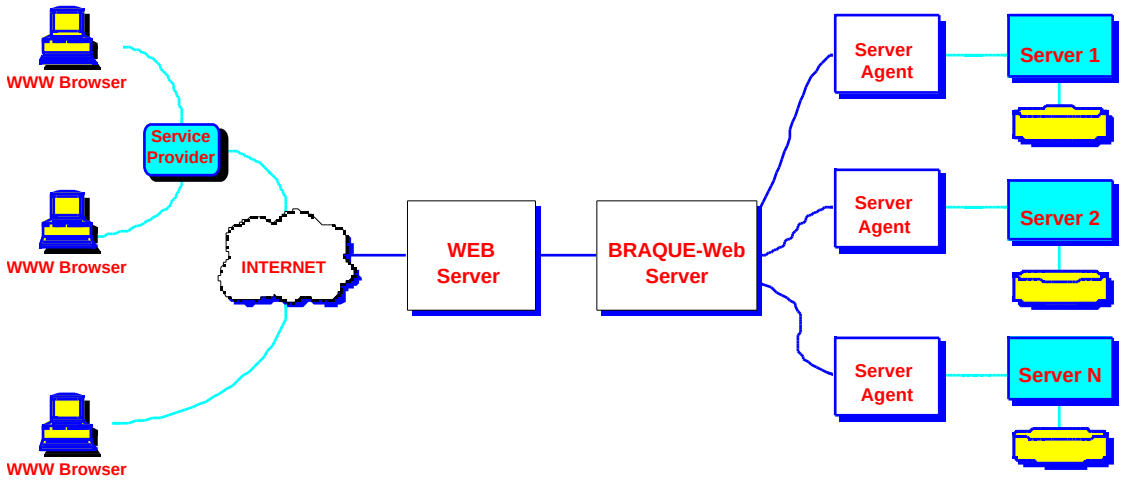


Figure 1 EINS Web Architecture

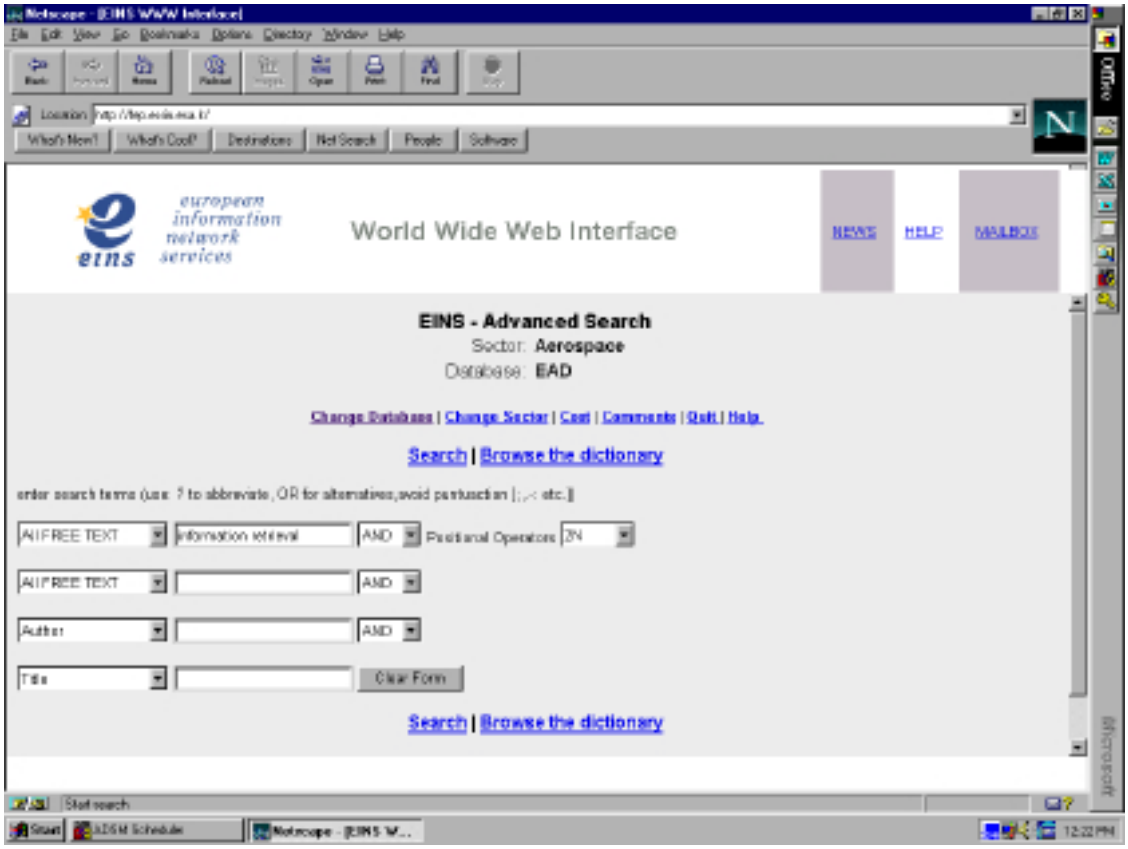


Figure 2 EINS-Web Advanced Search Form

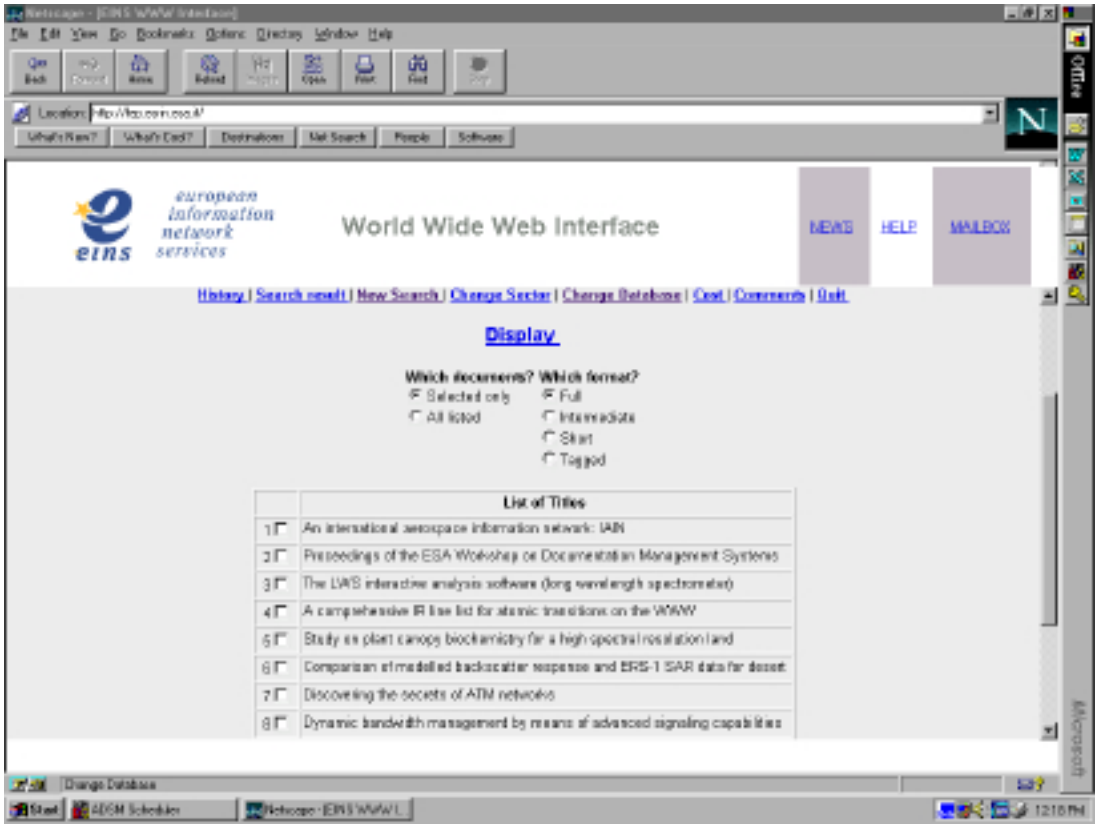


Figure 3 EINS-Web Title List Form

The evaluation of EINS-Web has been carried on in the Library of the National Research Council (CNR) in Bologna (Italy), which was selected as the test site, involving several researchers of the CNR campus with their real information problems (in various fields, such as chemistry, material science, electronics, physics, geology, environment, etc...) and a mixed group of evaluators, composed by information specialists and experts in user interfaces.

Figure 2 and 3 give an idea of the look and feel of EINS-Web GUI.

In the first step towards the construction of the EINS Web interface, we re used the design efforts for the development of the previous version of the interface (BRAQUE PC) (BRAQUE = BRowse And QUEry), developed for the Windows environment.

The design of the BRAQUE PC interface was initially based on an analysis of users' information seeking behavior and Cognitive Task Analysis [2]. As a matter of fact, the information seeking strategies can identify a multiple dimensional space. This space is characterized by the information problem, by the nature of the information itself (e.g. information, meta-information), by the user's goal (e.g. learn, select), by the information access method (e.g. browse, search) or by the information access mode (for example: recognize, specify). During the design process we understood that different information seeking behaviors were relying on common functional elements. We used the two basic functional elements: the browser and the searcher to cast the multidimensional space defined by the identified information seeking strategy [3]. The heuristic evaluation method was then evaluated and used to assess user satisfaction, ease of learning, ease of use, error prevention and efficiency of the interface and as a feedback tool to drive a spiral design process. Evaluator's judgment was based on the nine heuristics from Nielsen (see Table 1) [4].

Any usability problem was fitted and analyzed against one of the nine criteria (see Table 2). A number of issues were detected, mainly of aesthetic nature. Some were serious, influencing heavily the interaction. Where possible, suitable solutions were identified. Sometimes solutions could be drawn from the nature of the problem itself.

The designers then revised the report produced by the evaluation team [5]. It came up that in some cases they were already aware of the presence of an interface problem, but did show that the implementation approach was lead by precise constraints from the underlying architecture. In most of the cases the developers agreed with the evaluators recognizing the usability problems. In one case only the disagreement could not be resolved. For BRAQUE PC the designers used the classic desktop metaphor. A good management of buttons, windows and pull down menus together with a simple use of colors, makes simple and effective the objects' manipulation. The designers selected a task-oriented approach, associating different tasks to different environments (windows). In BRAQUE PC it is possible to interact with the databases either via the query language or via graphical objects. This possibility was considered very important, in fact it allows either to enter the commands in the query language or to interact with graphical tools (and eventually verifying the correspondence with the query language commands). Good also was considered the possibility to record on a file any operation executed (log file). This is especially useful for search results and queries.

A new report was produced [6], putting together evaluators' and designers' replies, that served as «seed» to drive the design process for the World Wide Web version of BRAQUE. This new version was called EINS Web following the decision of the international EINS consortium that was going to exploit the interface.

The report provided a form containing the evaluation criteria and rules to be followed in the development of the new EINS-Web GUI. The EINS-Web heuristic evaluation form contains a suitable subset of the 101 usability heuristics as retrieved and analyzed from a database of 249 usability problems by Nielsen in [7]. The heuristics have been selected according to our judgment that they were likely to fully describe problems in the WEB interface and that they could be easily understood by different evaluators (see Table 3).

A second evaluation session was then run. Students of the Department of Computer Science of the University of Bologna, who had taken the one-year course in User Interface Design and Evaluation, carried on the evaluation in the Library of the CNR. Before arranging for the EINS-Web evaluation session, the students were given a training seminar on architecture and languages of information retrieval systems, which most of them had never previously used. The heuristic evaluation form (see Table 3) was then distributed to the evaluators and discussed with them. For each usability problem a separate column was given to be filled with a rating value: we agreed to assign values from 1 (the interface does not keep into consideration the usability problem at all) to 5 (the usability problem has been completely worked out). The evaluators were requested to identify potential usability problems and to tie each problem found to the specific heuristic it violated. As in the previous evaluation, multiple heuristics could be linked to any given violation. Finally, the evaluation session was carried on. The same evaluation, dealing with same information problems, was run by four different groups of students in parallel sessions. One group tested the interface using the Internet Explorer browser, while the others performed the evaluation using the Netscape browser.

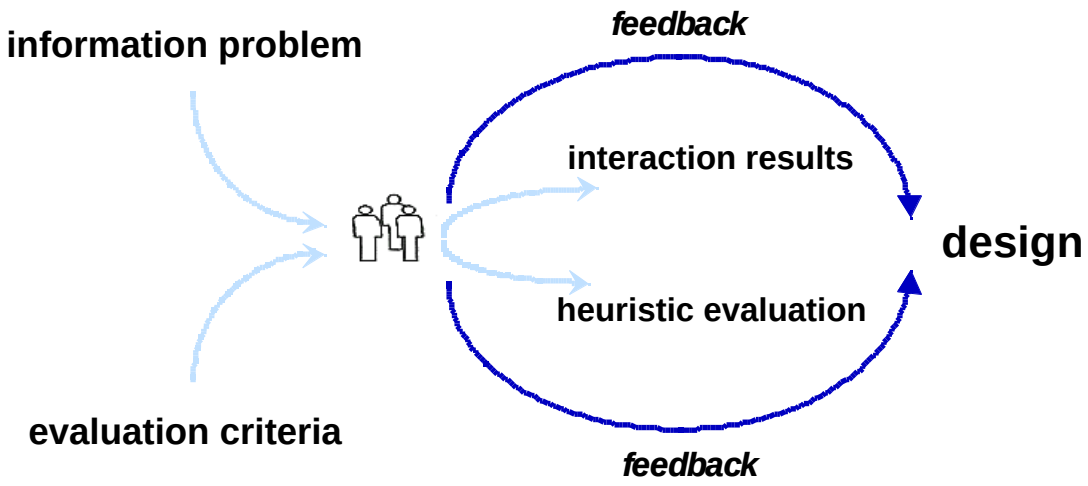


Figure 4

The feedback to the designers (see figure 4) allowed to modify the design during the implementation phase.

3. Lessons learned

Here is our assessment of the evaluation methodology:

· When the results of the four different groups were put together, it came up that the evaluators had used all the 45 heuristic present in the evaluation form, either in positive matches (that is with a score >= 3), or in negative matches (that is, one or more problems recognized that was tied to the heuristic).

· Aesthetic heuristic had to be taken in account and should be present in the evaluation form.

· Different nature of evaluation results, depending on the presence of real users or not. It appeared in fact that in a «pure» heuristic evaluation session (only interface experts, no real users) it was possible to detect problems relating more to the interactive behavior of the interface, such as users' behavior problems, conceptual user model, aesthetic design. The evaluation done in presence of real users allows getting deeply through the information seeking interaction problems.

Interface evaluation:

· The evaluation was useful to detect some design issues. In particular two problem areas were identified. The first related to feedback and visibility of system status. In many cases, for example when «search» or «refine» action is selected, it is not clear to the user if input has been received. There are no messages or status bar indicating progress in task performance. However, the choice of not providing this information to the user is motivated by the fact that between user and responding host there is a network, gateways, multiple hosts, etc.. and implementing a status control would have overloaded the system.

The second issue related to user background knowledge and user conceptual model. It was in fact recognized that EINS-Web should provide the same or improved functionality already present in EINS PC. The presence of a time-out resulted also very «annoying». Apparently it was not related to any particular user action or network error. Users get disconnected without notice or error message (or, after having waited for the result of a query, and having stopped the execution with the Browser's STOP button, the system replied with the time-out disconnection message error). This actually happens because there are three different time-outs, not tied or synchronized to each other, that convey to the user the erroneous model of a «single wild time-out».

· The matching of the interface design and user expectation is difficult when information space is dispersed over very large collections: the expert users require to increase the interface functionality to achieve their goals, whilst non-expert users require to reduce the interface functionality in favor of intuitive and simple features.

As preliminary conclusion we feel that the interaction among evaluators and designers proved to improve the success of the spiral design methodology depicted in figure 4, which is needed to cope with the requirements of designing interfaces targeted at the rapidly evolving Internet world.

Acknowledgments

We are indebted with Prof. Cesare Maioli of the Department of Computer Science of the University of Bologna, and his students of the User Interface Design and Evaluation course, for the time and effort spent in the evaluation of EINS Web interface. CINECA designed the look and feel of the EINS Web interface as available to the EINS users at http://www.eins.org. The technical implementation was done by Vitrociset for the European Space Agency and the EINS consortium in the framework of the BRIDGE and CIME projects co-financed by the European Union.

References

[1] Nielsen, J., Mack, R. L. Editors; ***Usability Inspection Methods,*** Wiley and Sons Inc. , 1994

[2] Belkin N., Marchetti P.G., Cool C. ***BRAQUE: Design of an Interface to Support User Interaction in Information Retrieval,*** Information Processing & Management, vol.29 n.3, 1993, pp.325-344

[3] Marchetti, P.G.; ***BRAQUE: A Hypertext-based Interface for Accessing Large Text Databases***, Informatica e Diritto, Special Issue: on Hypertext and Hypermedia, vol. III (1994) - n.2 pp.10-112

[4] Nielsen J. and Molich R. ***Improving a Human-Computer Dialogue;*** CACM, 33 (3), March 1990, pp. 338-348

[5] Mangiaracina S. , Merivot. F., Statti F.; ***BRAQUE evaluation report,*** Technical Report: BRIDGE - IMPACT Project BIS-4017 10284/0 BRIDGE/WP3/EV/PGM/01/00

[6] Zappaterreno P., Marchetti P.G. ***BRAQUE-WEB Project. An heuristic evaluation guideline*** Technical Report: BRIDGE - IMPACT Project BIS-4017 10284/0 - BRIDGE/WP3//HEG/PZ/01

[7] Nielsen, J. ***Enhancing the explanatory power of usability Heuristics***; Proceedings ACM CHI'94 Conf., April 1994, pp.152-158

Table 1 Nielsen's heuristic guidelines

Visibility of system status	The system should always keep the user informed about what is going on by providing him or her with appropriate feedback within reasonable time.
Match between system and the real world	The dialogue should be expressed clearly in words, phrases, and concepts familiar to the user rather than in system-oriented terms.
User control and freedom	A system should never capture users in situations that have no visible escape. Users often choose system functions by mistake and will need a clearly marked «emergency exit» to leave the unwanted state without having to go through an extended dialogue.
Consistency and standards	Users should not have to wonder whether different words, situations, or actions mean the same thing. A particular system action - when appropriate - should always be achievable by one particular user action. Consistency also means coordination between subsystems and between major independent systems with common user population
Helping users recognize, diagnose, and recover from errors	Good error messages are defensive, precise, and constructive. Defensive error messages blame the problem on system deficiencies and never criticize the user. Precise error messages provide the user with exact information about the cause of the problem. Constructive error messages provide meaningful suggestions to the user about what to do next.
Error prevention	Even better than good messages is a careful design that prevents a problem from occurring in the first place
Recognition rather than recall	The user's short-term memory is limited. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. Complicated instructions should be simplified.
Flexibility and efficiency of use	The features that make a system easy to learn -such as verbose dialogues and few entry fields on each display - are often cumbersome to the experienced user. Clever shortcuts - unseen by the novice user - may often be included in a system such that the system caters to both inexperienced and experienced users.
Aesthetic and minimalist design	Simple things should be simple complex things should be possible. Things should look good, keep graphic design simple, and follow the graphic language of the interface without introducing arbitrary images to represent concepts.

Table 2 BRAQUE PC evaluation results

Heuristic	**BRAQUE PC Evaluation**	**Problems**
System status visibility	Normally information on system status is available. Feedback is good, as well as overall direct object manipulation	in some instances, when a task is activated, it is not possible to execute other operations. All that is not signaled. It is preferable, in such cases, to change mouse pointer into the hourglass.
Match of the system with real world	The interaction is driven by the desktop metaphor. It is not necessary to know specific terms to use the system. Users must anyway know English language. In the Italian version most of interaction dialogues is in English	In the Italian version most of the messages are in English. In some message box some of the text is in Italian some in English. We suggest to perform a coherent complete translation or to keep the original version.
Control and user	User control of the GUI is good. It is possible to enable via the «edit» menu	In the «Search Results» environment, it is not possible to cancel sets. If a large number of sets are

freedom	the «undo/redo» option for one level	available the user may be confused. If the cancel utility would be available it would be easier to organize the work and keep the environment under control.
Coherence and standard	In general, the standard coherence rules of GUI have been followed.	· It is possible to perform the same operation (search) in the «Document Searcher» and in the «Idea Finder». The same search executed in the two environments, leads often to two different results. This is considered the most serious flaw of the interface · According to Windows95 specifications «x» and «-» buttons placed in the rightmost angle of the window should close the environment and minimize the window respectively. In EINS windows both buttons minimize the window without closing the application. · «Document Searcher» and «Document Pool» icons are ambiguous since their label starts with the same word: Document. · Tested version was 1.3, but on-line help referred to 1.2. · Password is not masked.
Error prevention	Error prevention is good. It is based on a clear organization of interaction tools.	If a file of type «Term Pool» is opened active window is canceled and replaced with the new archive. In such a way previous archive information is lost. The same is valid for the «Document Pool». In the latter case files are on disk and are not lost. However, is not possible to compare two Document Pools.
Recognition instead than recall	Graphical tools offer options clear and self explanatory, it is not necessary to record the steps needed to perform a search	It is not clear the method to be used to insert search keys in the «Document Searcher». The onscreen explanation is not exhaustive. The user has to remember the default operator inserted among the terms on the same line. The fact that a fixed logical AND links the four entry boxes for the search terms is a constraint for the resulting query
Flexibility and use efficiency	The separation of tasks allows a flexible and efficient use of the system	· Whilst executing commands in remote mode, it is not possible to insert anything locally. · In order to save terms in the «Document Searcher» it is necessary to close at least once the window during the working session and then to reopen it.
Aesthetic characteristics and minimalist design	The structure of the interaction tools, the colors' management, the small number of graphical objects are characteristics of the EINS interface, determining a nice looking interface	
User aids in recognize detect and correct mistakes	Error messages are always clear	When leaving EINS, the interface suggests to disconnect using the DISCONNECT command even if the command was already executed, and therefore not active

Table 3 Heuristic evaluation form

Visibility of system status	**Score (1-5)**
Feedback: keep user informed about what goes on	
Provide status information	
Feedback: show that input has been received	
Features change as user carries out task	
Feedback provided for all actions	
Feedback timely and accurate	
Indicate progress in task performance	
Direct manipulation: visible objects, visible results	
Identity cues system response vs. user's goal	
Show icons and other visual indicators	
WYSIWYG; do not hide features	
Match between system and the real world	
Speak the user's language	
Contains familiar terms and natural language	
Metaphors from the real world	
Familiar user's conceptual model	
Use of user's background knowledge	
User control and freedom	
Undo e redo should be supported	
Obvious way to undo actions	
Forgiveness: make actions reversible	
Ability to undo prior commands	
Clearly marked exits	
Ability to re-order or cancel tasks	
Modeless interaction	
User control: allow user to initiate/control actions	
Consistency and standards	
Consistency: express same thing same way	
Consistency: same things look the same	
Uniform command syntax	
Conform to platform interface conventions	
Show similar inf. at same place on each screen	
Error prevention	
Prevent errors from occurring in the first place	
System designed to prevent errors	
What planning mistakes are most likely ?	
Recognition rather than recall	
See-and-point instead of remember-and-type	
Make the repertoire of available actions salient	
Seeing and pointing: objects and actions visible	
What features often missed and at what cost ?	
Provide list of choices and picking from list	
Minimise the user's memory load	
Easy or difficult to perform (execute) tasks ?	
Allow access to operations from other applications	
Show icons and other visual indicators	
Flexibility and efficiency of use	
Shortcuts: Accelerators should be provided	
User tailorability to speed up frequent actions	
User interface should be customisable	

Multiple Metaphor Environments: Issues for effective interaction design

C. Stephanidis and D. Akoumianakis

Institute of Computer Science, Foundation for Research and Technology-Hellas (FORTH)
Science and Technology Park of Crete
GR-71110, Heraklion, Crete, Greece
Tel.: +30-81-391741, Fax: +30 - 81 - 391740
emails: cs@ics.forth.gr, demosthe@ics.forth.gr

Abstract. This position paper presents the notion of multiple metaphor environment and discusses principles and techniques for constructing user interfaces as multiple metaphor environments. Though, multiple metaphor environments represent a generic concept, they are particularly relevant to novel application domains and technologies, such as Digital Libraries.

1. Introduction: Metaphors and the user interface

The notion of a metaphor in interface design is increasingly becoming a critical aspect in the attempt to provide more effective and higher quality interaction between humans and artefacts. Though several efforts have been devoted to the study of metaphors (e.g., Carroll et al., 1988; Henderson et al., 1986; Moll-Carrillo et al., 1995), very little is known as to how they can be systematically embedded into computer-based interactive software. At the same time, the number and diversity of application domains, in which the use of metaphors is critical, continuously increases (examples include, educational software, digital libraries, home-based interaction environments, virtual and augmented realities, health care records, electronic commerce, etc). A closer look into these application areas and the study of success of respective systems, reveals that, whereas in the past the use of metaphors was at the discretion of the designer, or in the best of the cases, bound to what the underlying development toolkit offers (i.e., trashbins, form filling), today and for certain non-traditional / non-business applications, embedding metaphors to interface design is compelling for the wide adoption and user acceptance of the application.

The use of metaphor may be studied at various levels, ranging from the overall interactive environment offered by an application, to the task level (i.e., how users engage and perform specific goal-oriented activities), as well as the physical level of interactions (i.e., icons used to convey intended meaning). Moreover, it is important that each of those levels may not involve the articulation of the same real world metaphor, but variants of different ones. Thus, at the level of the overall interactive environment, users may be exposed to a books-like (Moll-Carrillo, et al., 1995) or rooms-like (Henderson et al., 1986) metaphor, while in order to accomplish specific

tasks (such as for example, deletion of a file) alternative metaphors (i.e., deleting a file from a folder) may be recruited.

This leads to the conclusion that progressively interactive computer-based applications move towards a state which can be characterised as *multiple metaphor environments.* The notion of a multiple metaphor environment was firstly introduced and elaborated in the context of FRIEND21 (a major Japanese collaborative research and development project). There, it was claimed that interactive systems capable of mapping concepts from a source domain (i.e., database search operation) to different / multiple target domain functions (e.g., newspaper or HTML-based search), and vice versa, provide multiple metaphor environments. However, the main thrust of work in FRIEND21 was mainly conceptual and, as a result, did not deliver any detailed insight into how such systems may be specified, designed or implemented, other than guidelines for the human interface of the next century (Institute for Personalised Information Environment, 1995).

In this position paper, we revisit the notion of multiple metaphor environments from a slightly different view angle. In particular, we are interested to investigate the contributions of multiple metaphor environments to the design and development of user interfaces for different user groups, including people with disabilities (Stephanidis, 1997; Akoumianakis, et al., in press). Our objective is to draw upon recent experience, in the context of collaborative research and development projects, and shed light into the way in which (i) metaphors may become embedded into user interfaces, (ii) multiple metaphor environments may be specified, realised and implemented, and (iii) the above impact on the architectural abstractions of user interface software. The presentation of the relevant issues will be complemented by reference to example case studies in which multiple metaphor environments have been designed and implemented. In particular, examples, involving the fusion of conventional visual desktop interaction, augmented interaction elements (Savidis et al., 1997), such as scanning, and alternatives interaction environments, such as non-visual Rooms (Savidis et al., 1995), will be discussed with the view to reveal characteristic properties of such interactive software, development tool requirements, and prospective challenges (Stephanidis, 1997).

2. Multiple metaphor environments

In the context of the present work, metaphors are considered to have a two-fold purpose in interface design. They can either be embedded in the user interface, or characterise the overall interactive environment of an application. For example, the menu interaction object class, as commonly encountered in popular user interface development toolkits, follows the “restaurant” metaphor, and provides an example of embedding metaphor into a user interface. This is because it is commonly found as embedded element in systems conveying radically different interactive embodiments of the computer; examples are the visual desktop as in Windows95TM, rooms as in (Henderson et al., 1986), or book as in (Moll-Carrillo et al., 1995).

Alternatively, a metaphor may characterise the properties and the attitude of the overall interaction environment. For instance, the visual embodiment of the desktop metaphor in Windows95TM presents the user with an interaction environment based on high level containers, such as sheets of paper called windows, folders, etc., which characterise the overall interactive embodiment of the computer. Systems, such as those in (Henderson et al., 1986) or (Moll-Carrillo et al., 1995), are examples of alternative embodiments of real world metaphors into a user interface. It should be noted that a particular real world metaphor may have different interactive instantiations. Thus, for example, OSF/MotifTM and Windows95^{TM} support variations (mainly in the look and feel) of the visual embodiment of the desktop metaphor. From the above, it follows that the interactive environment of a metaphor is realised by specific user interface development toolkits.

Different interaction metaphors may be facilitated either through the enhancement, or augmentation of existing development toolkits, or by developing new toolkits. For instance, an enhancement of the interactive environment of a metaphor may be facilitated by introducing new composite object classes, such as the note cards in prevailing Windows-like systems, or by embedding in the toolkit additional interaction techniques, such as automatic scanning facilities for interaction object classes (Savidis et al., 1997). What is important to note about enhancement, or augmentation is that it rarely alters the overall interactive environment of the metaphor. This is because, the scope of the enhancement, or augmentation does not account for top-level container object classes (such as a window in Windows95TM, the room in (Henderson et al., 1986) or the book in (Moll-Carrillo et al., 1995). Instead, through sub-classing, augmentation extends the range of simple or composite interaction elements that may be supported in a toolkit's object hierarchy.

In case that an alternative interaction metaphor needs to be supported, then it may be realised through new toolkits. An example of the latter case is reported in (Savidis et al., 1995; Savidis et al., in press) where Commonkit is used to support user interaction based on a non-visual embodiment of the Rooms metaphor through speech and / or Braille output and keyboard input. COMMONKIT offers the full range of programming features encountered in currently available GUI toolkits, such as hierarchical object composition, dynamic instantiation, call-back registration and event-handling. In its current version, COMMONKIT implements only one container, namely Room, and several object classes (e.g., floor, ceiling, front / back / left / right wall), in addition to conventional objects, such as menu, toggle (represented as on / off switch), button, text reviewer, etc. A more elaborate account of the object library of COMMONKIT, as well as applications built with it, can be found in (Savidis et al., in press).

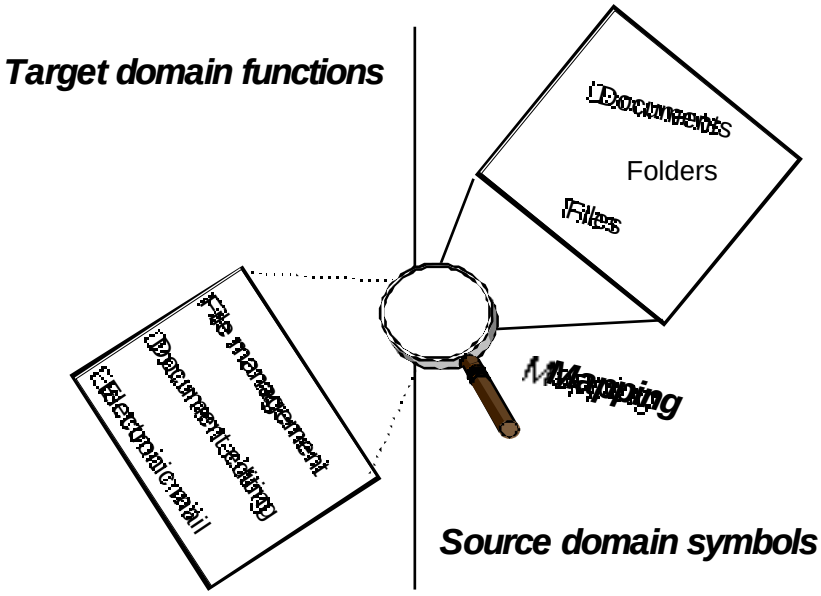


Figure 1: Concept of source and target domains

Following the above, the notion of a multiple metaphor environment implies a particular computer-based embodiment of an integrated system, capable of performing context-sensitive mapping between functions in a target domain (e.g., functions of a computer environment) to symbols in a source, or presentation domain (e.g., the desktop interactive embodiment), and vice-versa (Figure 1). Alternatively, it may be conceived as an integrated multiple toolkit platform, capable of context-sensitive mapping. For example, consider typical functions such as file management, electronic mail and editing, as performed in a computer environment (target domain). Such functions in the target domain are mapped onto user operations on objects (i.e., folders, documents, drawers) of the source domain, namely the desktop.

The visual desktop embodiment in current computer systems performs precisely such mappings between symbols from a target domain to symbols in the designated source domain. However, the visual desktop, as embedded in currently available user interface development environments, does not satisfy the conditions of multiple metaphor environment, since it does not perform any context-sensitive processing to map functions from the target domain to corresponding symbols in the source domain. This is because the source domain is fixed and unique (i.e., the desktop of an office). In other words, there is no possibility to map a file management function onto a book operation, and vice versa. Consequently, the construction of multiple metaphor environments reflects two important properties, namely the explicit embodiment of alternative metaphors (i.e., desktop, book, library) into the user interface, as well as their fusion into an integrated environment (i.e., context-sensitive mapping).

To demonstrate the principles underpinning the design and development of multiple metaphor environments, let us assume three users, namely a sighted user, a child and a blind user. All three are tasked to carry out a file management operation, namely delete a file. Since the capabilities of the users differ (e.g., with regards to the modalities that may be employed to facilitate the interactive task), the interface should ideally

undertake the required transformation so as to present an appropriate (i.e., accessible and usable) instantiation, suitable for each user.

Figure 2 depicts indicative examples of plausible alternatives which can be realised in a programming-intensive manner, by providing separate interface implementations for each user. Alternatively, the same interface could be built, in such a way, so that it is capable of context-sensitive processing leading automatically to the undertaking of suitable transformations to map the file management operation onto appropriate interactive environments, such as those depicted in Figure 2.

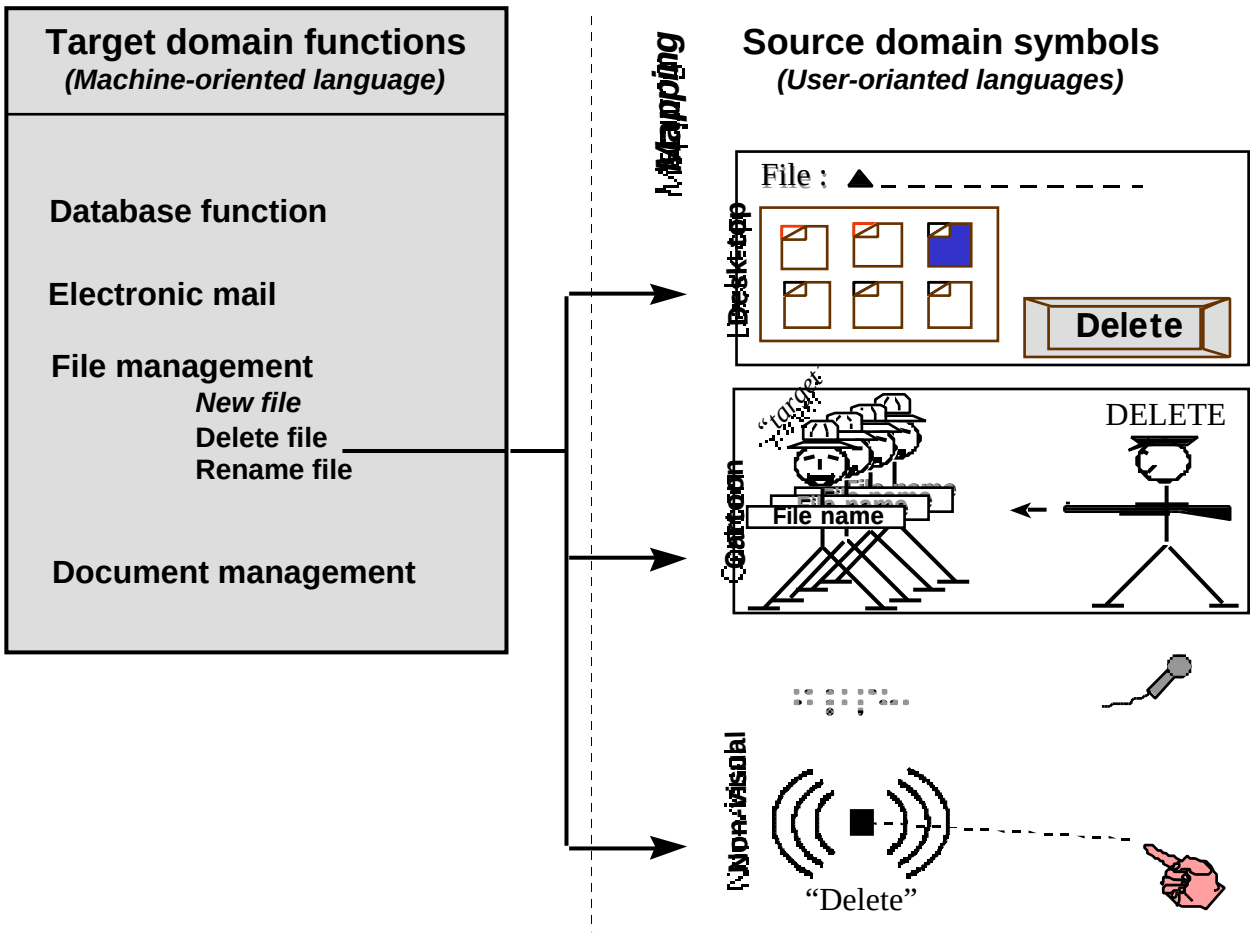


Figure 2: Mapping target domain functions to source (presentation) domain symbols

3. Conclusions

From the above, it follows that multiple metaphor environments are necessitated from the diversity of users (i.e., diverse requirements of different target user groups), the diversity of contexts of use (i.e., the variety of contexts in which artefacts may be encountered) and the diversity of interaction platforms (i.e., proliferation of different interaction toolkits), all of which may necessitate sometimes radical changes in the design. As a result, the important features characterising such environments are that: (a) there is a clear separation between knowledge and presentation; (b) the system integrates components (i.e., toolkits) implementing alternative interactive embodiments of a particular artefact; (c) the system is capable of performing context-sensitive processing and selection of suitable symbols to interact with the user, based on information provided by a dedicated tool usually referred to as user modelling component, or user information manager, offering information, both general and task specific, on the current user; (d) multi-modality is preserved through the fusion of metaphors into an integrated environment.

References

Akoumianakis, D., Savidis, A., Stephanidis, C., in press. *Encapsulating intelligent interactive behaviour in unified user interface artefacts*. To appear in the International Journal on Interacting with Computers, Special Issue on The Realities of Intelligent Interface Technology.

Carroll., J., Mack, R., Kellog, W., 1988. *Interface Metaphors and User Interface Design*. In Handbook of Human-Computer Interaction, M. Helander (Ed.), North-Holland, pp. 67-82.

Henderson Jr., A., Card, S., 1986. *Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface*. ACM Transactions on Graphics, vol. 5(3), pp. 211-243.

Institute for Personalised Information Environment, 1995. *FRIEND21 Human Interface Architecture Guidelines*. Tokyo: Institute for Personalised Information Environment.

Moll-Carrillo, Salomon G., March, M., Fulton Suri, J., Spreenber, P., 1995. *Articulating a Metaphor Through User-Centred Design*. In the Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'95), Denver, Colorado, New York: ACM Press, 7-11 May, pp. 566-572.

Savidis, A., Stephanidis, C., 1995. *Building Non-Visual Interaction through the development of the Rooms metaphor*. Companion Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '95), Denver, Colorado, New York: ACM Press, 7-11 May, pp. 244-245.

Savidis, A., Stephanidis, C., in press. *The HOMER UIMS for Dual User Interface Development: Fusing Visual and Non-visual Interactions*. To appear in the International Journal of Interacting with Computers, 47 pages.

Savidis, A., Vernardos, G., Stephanidis, C., 1997. *Embedding Scanning Techniques Accessible to Motor-Impaired Users in the Windows Object Library*. In the Proceedings of 7th International Conference on Human-Computer Interaction (HCI International '97), San Francisco, California, USA, 24-29 August, pp. 429-432.

Stephanidis, C., 1997. *Towards the Next Generation of UIST: Developing for all users*. In the Proceedings of 7th International Conference on Human-Computer Interaction (HCI International '97), San Francisco, California, USA, 24-29 August, pp. 473-476.

Implementing the Common User Interface for a Digital Library: The ETRDL experience

Maria Bruna Baldacci, Stefania Biagioni, Carlo Carlesi, Donatella Castelli, Carol Peters
IEI-CNR, Pisa, Italy

Abstract

The Common User Interfaces for the ERCIM Technical Reference Digital Library (ETRDL) are described and the underlying motivations for certain design decisions are discussed. The lessons that have been learnt from this experience are outlined and possible future developments are suggested.

1. Introduction

Towards the end of 1997 the decision was taken by ERCIM (European Consortium for Informatics and Mathematics) to create a digital collection of the technical documentation produced by its scientists and to provide on-line distributed public access to this collection. The intention was to offer a service similar to that provided in the United States by NCSTRL, the Networked Computer Science Technical Reference Library (1). The aim was to assist the ERCIM scientists to make their research results immediately available world-wide and provide them with appropriate on-line facilities to access the technical documentation of others working in the same field.

It was clearly desirable to ensure that these two parallel services were compatible. We thus decided to adopt the same system as that used by NCSTRL: the Dienst system[1] developed by a US consortium led by Cornell University (2, 3), and to include our collections as part of the NCSTRL collection. However, it was quickly apparent that the ERCIM scientific community has its own specific requirements, not all of which are covered by the basic Dienst system as adopted by NCSTRL. Our point of reference was a meeting of ERCIM librarians and information scientists, at the end of 1995, in which the main requirements for the next generation library systems were listed and discussed. With respect to this list, the NCSTRL service was deficient in three important aspects: the need for classification mechanisms; the need to cater for languages other than English; the need to provide on-line document submission facilities. Our task has thus been to implement a system which maintains interoperability with NCSTRL so that users can perform cross-Atlantic bibliographic searches while at the same time extending this system to provide the functionalities requested by the ERCIM users. This means that a user accessing the NCSTRL system can view and query any of the collections of the eight ERCIM institutions currently participating in this initiative using the standard NCSTRL search functions, whereas a user accessing the ETRDL system directly has an additional set of functions available.

The services to be offered by the ETRDL were defined in a meeting between the partners[2] in January 1997 and the specifics for the common user interfaces were agreed. IEI-CNR was given the task of implementing them. In the rest of this paper, we will provide an analysis of the ERCIM user needs, list the extensions that have been made to Dienst in order to meet these needs, and describe how we have developed the Common User Interfaces to the

[1] The Dienst system to which reference is made in this paper is version 4.9.1. This version of Dienst was made freely available to us by the developers.

[2] The following ERCIM institutions currently participate in the ETRDL initiative: IEI-CNR, INESC, GMD-IPSI, SICS, ICS-FORTH, SZTAKI, CWI, INRIA. ETRDL is funded in part by the DELOS WG (ESPRIT LTR No. 21057), in part by ERCIM, and in part by the partners themselves.

system with respect to these extensions. Finally, we discuss the lessons we have learnt from this experience and discuss possible steps for the future.

2. The ETRDL Users

We recognise three distinct classes of users of our technical reference digital library within the ERCIM community;

- *information users*, i.e. people who will access one or more of the document collections available to find pertinent material;
- *information providers*, i.e. authors, or their representatives, who will submit new documents to a specific collection with associated bibliographic records;
- *information administrators*, i.e. those responsible (usually, but not always, the librarians) for verifying the correctness of the bibliographic records and the associated document files before inserting them into the relevant collection.

The main system interfaces had therefore to cater for the needs of these very different types of users: submission/elimination of information; access/searching for information, management of information. Here below, we discuss in detail the needs of these three user classes.

2.1 Information Users

The basic requirements[3] of the users browsing and querying the collections are to:

- find and retrieve pertinent information through interfaces which are simple, intuitive and homogeneous;
- retrieve documents meeting specific criteria, such as a given date, language, or type;
- have the results of a browse or search presented in an easy-to-understand format;
- view and download parts or all of documents retrieved.

Scientific users also want to be able to:

- access information on a given domain, using a familiar classification scheme;
- have clear indication of the status of the information retrieved (i.e. date, version, source).

Ideally, users would also like to be able to:

- access information using their preferred language;
- access all information available on given topic, whatever the language.

2.2 Information Providers

These users want to:

- submit their document files and the relevant bibliographic records to the system in an easy, fast and efficient way;
- be able to classify their documents using familiar classification schemes plus, if necessary, their own keywords;
- have mechanisms which automatically check and signal formal errors during compilation of the bibliographic records, e.g. when an mandatory field has not been filled in, when the incorrect syntax is used, etc.;
- have direct communication with the system administrator or librarian if necessary;
- make their documents as widely available as possible;
- be able to update or eliminate information files when necessary.

2.3 Information Administrator

In most of the ERCIM institutions the librarians will be responsible for verifying the formal correctness of the documents and bibliographic records submitted, deciding in which collection they will be inserted, and assigning the identification number. This type of user wants to:

[3] We have not considered here more sophisticated requirements, such as ranking and merging of results over collections and over languages. The feasibility of developing functions to obtain similar capabilities will be studied if it is decided to develop a more advanced version of the ETRDL system.

- receive homogeneous bibliographic records for all documents, which have been compiled correctly by provider and that conform with the type of bibliographic record normally used by the institution library;
- have an easy-to-follow procedure in order to enter new documents into a selected collection;
- be able to communicate with the information provider, if necessary.

2.4 General Needs

Common to all types of users is the need to have:

- on-line helps available at every step, and for every field, explaining both syntax and semantics;
- contact with the system administrator, when necessary.
- on-line access to the classification schemes adopted and mechanisms which make it possible to adopt a selected term from the schemes without having to rewrite it.

3. User Interface Design

The user needs outlined above have clearly (i) necessitated a series of modifications to the basic Dienst system, (ii) affected the interface design decisions. It is important to stress that any change to the underlying system will impact, to a greater or lesser extent, the implementation of the interfaces. In this section, we discuss briefly the main issues that have been considered when developing the ETRDL Common User Interfaces. These include the adoption of a common metadata description standard, the introduction of common classification schemes and methods to manage them, the implementation of multilingual interfaces. The first step, however, was to decide on the best way to present the system to its multiple user classes; this influenced the design of the Home Pages.

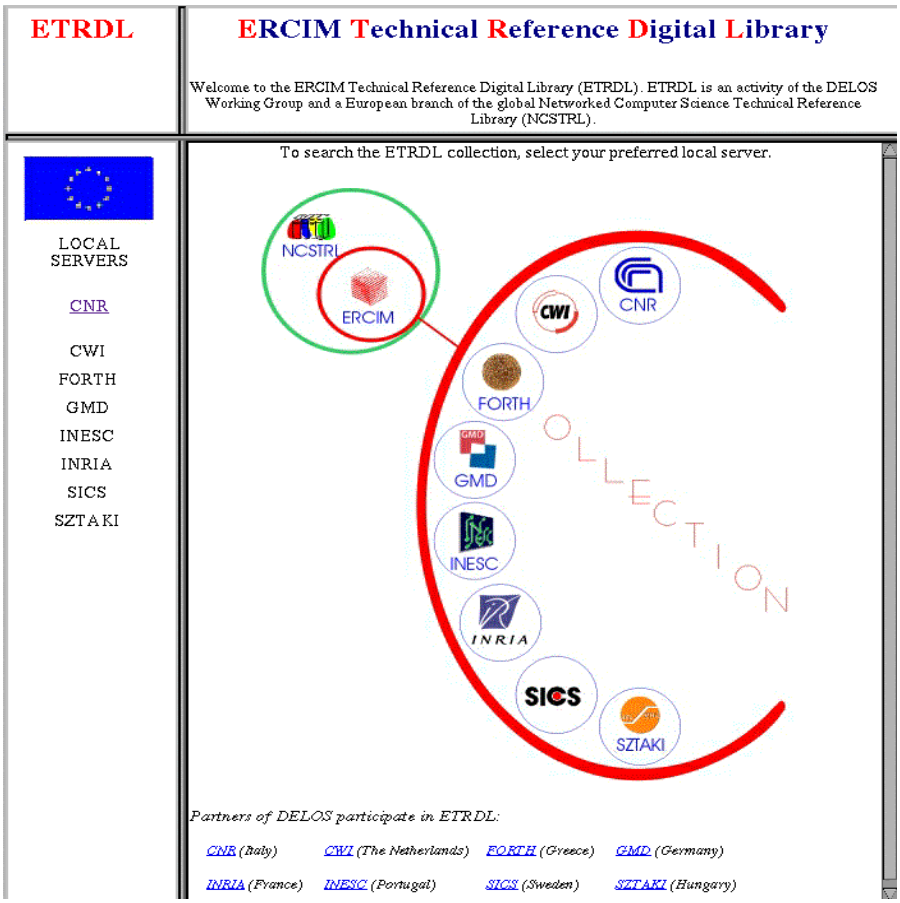


Figure 1 - The Centralised Home Page

3.1 ETRDL Home Pages

A first major decision regarded the system Home Page(s), i.e. the initial access points. In addition to the different types of users listed above, we have also had to consider two other dimensions: public vs. private; centralised vs. local. The ETRDL collection is intended to be publicly accessible, i.e. also by non-ERCIM users, but such users only need to access the search and browse functionalities; the information provider and administrator services are not relevant to them. At the same time, ETRDL is a distributed collection, consisting of the set of the local collections. The local collections are maintained on the local servers of each partner institution. This has comported the implementation of two levels of Home Pages. A centralised access point has been provided to the system through the DELOS Web site (http://iei.pi.cnr.it/DELOS/), whereas a local home page is installed on each local server. The "views" provided by these two different Home Pages respect the needs of the potential users at each site (centralised and local) and thus provide different points of entry.

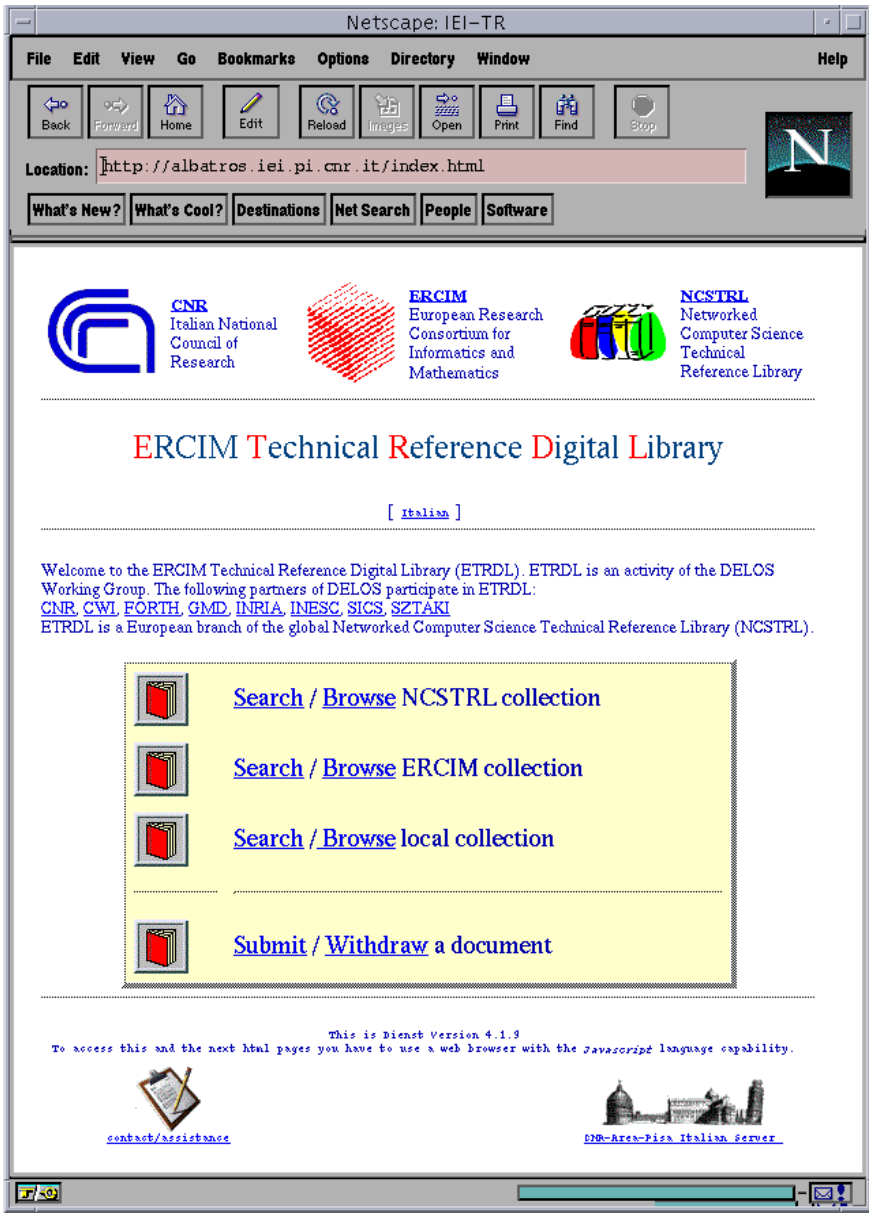


Figure 2 - The Local Home Page

The *Centralised Home Page* is in English only and has been designed for IT information users in general, not necessarily from ERCIM. For this reason, it provides links to pages that describe the objectives of the ETRDL, to on-line documentation, and to other relevant Web sites. It allows the user to access the ETRDL through one of the local servers (see Figure 1). Clicking on the logo of a given institution will open the relevant local home page interface. Our initial intention was to provide direct access to the ETRDL collection (with the extended set of functionalities) from the centralised Home Page. However, it was decided that this was not realistic; it implied maintaining a centralised server as well as the local ones. The user is thus informed that in order to search the ERCIM DL collection, he should select one of the local servers. At the same time, he is given a choice of language as each local server will maintain interfaces in English and in the local language.

The *Local Home Page* interface caters simultaneously for two user classes: *information users* and *information providers* by offering two main options: search/browse any collection; submit/withdraw a document to/from a local collection. This decision was taken to facilitate the local user, e.g. the ERCIM scientist, who can typically play either of these two roles and prefers to use just one main access point to the ETRDL.

A common local home page has been designed; its implementation is localised by each partner institution. The logo of the institution appears in the top left hand corner and, under the title, a button allows the user to switch between the interfaces in English or in the local language (see Figure 2).

From the local home pages, the search and browse functions can be activated over the entire NCSTRL collection, over the ERCIM collection, or over the collection(s) of the local institution. In each case, the user is not only accessing a different collection (or sub-collection), but is provided with a different perspective on the information, depending on the functions that have been implemented at that particular level.

The *Administrator Home Page* is transparent to the general public and accessible by authorised persons only. The main functions to be provided by the administrator interfaces were decided and defined in agreement with all the partner institutions; they depend to a large extent on the specifics for the Submission and Withdrawal forms. However, no common administrator interfaces have been designed; each local institution implements them according to local requirements.

3.2 The Metadata Set

In the ETRDL collection, each document has a common metadata description associated with it. This description is an extension of the basic metadata set used by Dienst and is compatible with the Dublin Core metadescription standard (4). This is important in order to guarantee future interoperability of the ETRDL with other DL systems.

This description contains the following elements: *Title*, *Author(s)* (the person(s) or organisation(s) primarily responsible for the intellectual content), *Subject* (a list of descriptors selected from the ACM and MSC categories, or free keyowrds), *Abstract* (an English abstract, and, optionally, a local language abstract), *Publisher* (the ERCIM institution), *Date* (the date of the intellectual content), *Type* (default value is Technical Report), *Format* (postscript, pdf, html, text, gif and tif), *Identifier* and *Language*.
Each field must be filled in when a new document is submitted to the collection.

The use of the Dublin Core elements in the user interfaces helps to impose uniformity over the collections and also ensures that the occasional user of the system is presented with a standard well-known set of metadata.

3.3 Search Interface

The common search interface to a digital library system is of great importance as the search function is the operation most frequently invoked by the users. The system thus tends to be judged on the merits of this interface: it must provide all necessary capabilities yet be *easy-to-understand* and *easy-to-use*. This has been our objective when designing the ETRDL search

interface. Although we have tried to implement an interface that does not appear too unfamiliar to an NCSTRL user (homogeneity between systems being a primary user requirement), as will be seen from the following description, ETRDL offers the user the possibility of performing a more finely structured search. This is obviously reflected in our interface.

The choice between three kinds of search, *direct search*, *simple search* or *fielded search*, has been maintained. The first two operate in the same way as in NCSTRL: the direct search can be used to retrieve a document via its Document-Id No.; terms entered in the simple search field are searched throughout the documents in all fields. However, in the fielded search the ETRDL interface also offers a number of additional functionalities: these include searching through a "subject" field and imposing conditions on the language or type of documents to be searched (see Figure 3).

Three different kinds of terms can be entered in the Subject field: ACM Computing Classification descriptors, AMS Mathematics Subject classification descriptors, and/or free keywords.

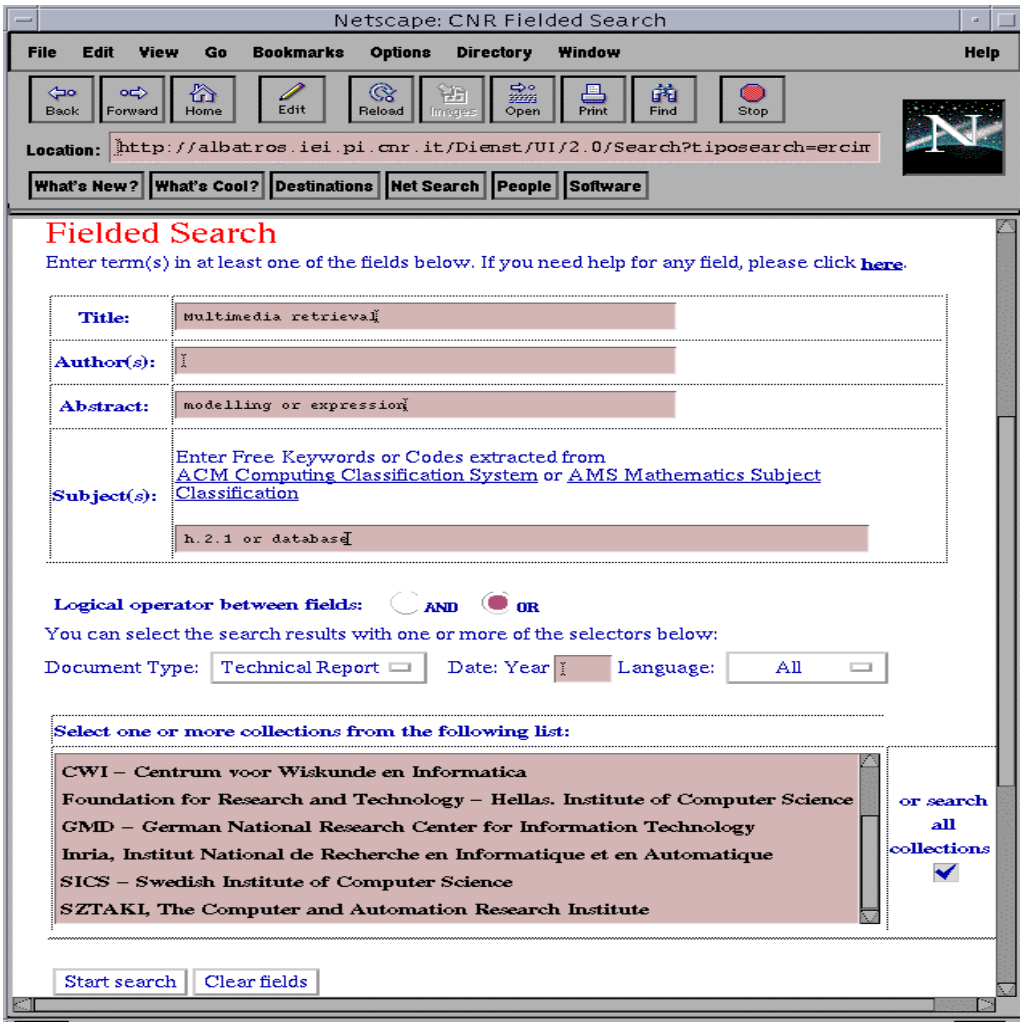


Figure 3 - The fielded search

There are two reasons why ETRDL offers the possibility of using standard classification schemes to describe and search the documents in its collections: the ERCIM librarians and scientists are accustomed to using such schemes and think of them as an efficient way to store and retrieve documentation; it is a way of imposing homogeneity over the distributed collections - if documents on the same subject are classified using the same descriptors they will be retrieved by the same query, whatever the collection they belong to, whatever the language they are written in. The classification schemes are accessible on-line so that the user can browse them in order to find the most appropriate search terms; he/she can then enter them in the subject field using "cut" and "paste" operations.

The capability of selecting documents by date or type that the NCSTRL interface provides implicitly by allowing the use of substrings when searching the Document-Id field, has been made explicit. Two additional fields, Date and Type, permit ETRDL users to set non-ambiguous conditions on the type, or date of documents to be searched (Dienst capabilities do not permit the selection of documents within a given range of dates). A selector for the choice of document language has also been added. Pop-up menus are installed to facilitate the user; he/she simply has to mark his/her selection by clicking on it with the mouse. On-line helps explaining the syntax and semantics adopted have been installed.

The main criterion adopted when designing and implementing this interface was to facilitate the user's task by guiding him/her as far as possible in formulating the query.

3.4 Submission Interface

In order to submit a new document to one of the collections which form part of the ETRDL, the document submission form or bibliographic record must be completed. For convenience, it was decided that the authors of documents should compile their own bibliographic records and submit these together with the text file(s) to the system. The design of this interface was thus extremely important. It was not sufficient to provide on-line helps and access to the classification schemes with «cut» and «paste» mechanisms to enter descriptors on the submission form without the risk of typos. A series of formal verifications are made by the system when the user submits the form, in order to check that all the mandatory fields have been filled in and, where possible, that the syntax has been respected. If the system does not accept the form, it returns it to the user requesting him to correct it. When a correct form is submitted, it is displayed to the user as a bibliographic record and the user is asked to confirm. On confirmation, the form is sent to the administrator of the collection indicated on the submission form; it is the administrator who is responsible for the actual insertion of the new document in the system – this is transparent to the user who may well believe that the document has been inserted directly into the system.

3.5 Rendering the Interfaces Multilingual

Multilinguality is an issue of strategic importance for the ERCIM scientific community, which currently consists of 14 member institutions, with 13 different major European languages. The first activities of the ETRDL in this area are aimed at (i) implementing user interfaces capable of handling multiple languages and (ii) providing very basic functionalities for cross-language querying.

Multilingual Access. Each national site is responsible for localisation, i.e. implementation of local site user interfaces in the national language as well as the CUI in English. The user will thus have the choice of using the system in English or in the local language. At the very simplest level, this means translating the common system interfaces (including the on-line helps) into the local language. For the system home pages, at each local site we maintain a version in English and in the local language; the user can switch from one to the other using the language button at the bottom of the local home page. However, all the other interfaces of the system are generated automatically during run-time. The system code thus includes a language variable, which determines whether the procedures should invoke interfaces and system messages in English or in the local language. Of course, localisation also implies providing the metadata field descriptors in the local language as well as in English. One of the tasks of the group is to investigate problems involved in rendering the Dublin Core element set multilingual (5).

More complex at both the interface and the system level is the question of being able to handle and visualise multiple character code sets. Each document submitted to the collection is tagged for language. Mechanisms are currently provided for the local display and printing of non-Latin-1 languages (this has been implemented at ICS-FORTH). In the future, we will probably move to Unicode. We are currently working on implementing mechanisms for the indexing of documents in languages other than English; this, however, is a question that remains transparent to the user.

Cross-language Querying. A simple form of cross-language querying is possible using the controlled vocabulary (ACM/AMS) terms. All documents in the ETRDL, in whatever language, classified using this scheme, can be searched. As authors are also requested to include an abstract in English, English free term searching over documents in any language is also possible. INESC has developed an LDAP service with a multilingual repository for the ACM and AMS classification systems (currently implemented in English and Portuguese), which will be integrated in the ETRDL system. This multilingual service will make cross-language querying in local languages possible in the future.

4. Lessons Learned

It is probably true to say that most of the lessons we have learnt from this experience are predictable from the literature and, at a first glance, may appear all too obvious. However, it is one thing to recognise the existence of a problem theoretically; it is quite another story to have to implement real-world solutions to this problem. A number of factors may well affect the decisions taken, e.g. time and cost issues, the need to counterbalance between different needs and different priorities. The solution adopted is often a compromise. An example of this is our decision concerning the Centralised Home Page; we would have preferred to provide direct access both to the NCSTRL server in the US and to the ETRDL service from this page. However, this raised two problems. In the first case, the NCSTRL service which is accessible via the US server is no longer the same as the one we have implemented on our local servers. It implements a different version of Dienst (see the following section for details). It was decided that it would only cause confusion to the ETRDL users if we offered them access to two different versions of the same system. In the second case, direct access to the ETRDL from the Centralised Home Page implied maintaining an extra server; this would have been costly in terms of implementation and we have thus been forced to provide ETRDL access only from the local servers.

In other cases, instead, a decision has been made in favour of the user requirements. For example, according to our user requirements, the ERCIM interface should offer i) subject access and ii) date/type/language selectors. However DIENST was found to be inadequate for these functions because:

i) DIENST search strategies permit only one boolean operator (either OR or AND) to be used between entered fields, but the content of the Subject field is matched against three different indexes (ACM, AMS, and free keywords) with an OR logic. This conflicts with DIENST search strategies if the user also wants to use the AND between the bibliographic fields;

ii) DIENST is a session-less system, i.e. it does not permit the search results to be further processed by the user. Consequently, the date/type/language fields cannot be used as true selectors (i.e., to select the search results); they must be used as search fields and always ANDed with any bibliographic fields entered. This conflicts with DIENST search strategies if the user wants to use OR between the bibliographic fields.

In this case, these conflicts have been resolved by heavy changes to the retrieval mechanisms of Dienst, of the help instructions, etc.

Therefore, the main two lessons we have learnt - or perhaps more correctly that we have had confirmed - while implementing the ETRDL interfaces are that:

- it is difficult to make clear distinctions between the interfaces and the underlying system; changes to one almost always affect the other – at times in an unexpected fashion;
- it is not easy to modify and extend an existing system; at times it is impossible when the extensions actually affect the philosophy of the system.

Dienst – as implemented by NCSTRL - provides a simple, monolingual free-text search service. We have extended this service by adding controlled vocabulary search facilities, multilingual interfaces, forms for the on-line compilation of bibliographic records and the submission and withdrawal of documents. All these extensions have led to the creation of a complex system, designed to meet the needs a number of different user types and profiles. This in its turn has lead to the need for a careful study of the interfaces in order to present the correct view of the system to each class of user.

It is true to say that during the implementation of the user interfaces, we have been forced to realise that many of our original assumptions were over-simplistic and did not reflect the true complexity of the system we were developing. This has necessitated a cyclic process: initial definition of the functionalities to be offered by ETRDL; consequent modification to the Dienst system; implementation of interfaces; distribution of the system to the other partners for testing; revision of the system and adaptation of the interfaces in response to the feedback received; redistribution and retesting[4].

To sum up, we may be a little sadder (and less optimistic) than when we started but we are certainly much wiser with respect to the underlying implications of the complex task of implementing user interfaces for a distributed, multilingual DL system.

5. Next Steps

Important decisions have now to be taken by the ERCIM DL group. In this paper we have described the first installation of ETRDL service, the design of the user interfaces, and the lessons we have learnt from this experience. Our aim was to implement a simple but effective service not only for the ERCIM but also for the general IT community in Europe, which would provide them with fast access to scientific documentation. At the same time, we provide the ERCIM scientists with an easy method to make their results immediately available to their peers, without having to wait maybe years for official publication. We are now evaluating the impact of the service on our users.

However, as so often happens in the computer science world, developments proceed at a break-neck pace. While we are in the process of completing the first stage of the common implementation, a new (and final) version of Dienst has been developed at Cornell. This version provides functionalities to order the results (including ranking). NCSTRL has adopted this new version of Dienst. If we want to maintain compatibility with the NCSTRL service, we must produce a new version of the ETRDL system which incorporates the new functionalities. The problem is that, as the ETRDL system now represents a heavily modified version of Dienst, it is difficult to estimate exactly how much work is involved in an upgrading to ensure compatibility with the new NCSTRL service. At the same time we recognise that the service we are offering is limited to text and images; a DL should also be capable to providing capabilities for the storage, management and access and retrieval of multimedia objects, e.g. also audio and video. The Cornell group has announced that it is not considering further developments to Dienst. It is now working on the design of a new object-oriented DL architecture (8,9), which will be able to handle such objects. Perhaps our next step should be to consider a system of this type. This would of course also imply a complete revision and redesign of the user interfaces.

6. References

1. Networked Computer Science Technical Report Library. http://ww.ncstrl.org
2. C. Lagoze, E. Shaw, J. R. Davis and D.B. Krafft, *Dienst: Implementation Reference Manual*, Cornell Computer Science Technical Report TR95-1514.
(http://cs-tr.cscornell.edu:80/Dienst/UI/2.0/Describe/ncstrl.cornell/tr96-1595)
3. C. Lagoze, J. R. Davis Dienst: an Architecture for Distributed Document
Libraries, *Communications of the ACM*, 38 (4) April 1995, page 45.

[4] The first implementation of the system, distributed for testing and demonstrated at ECDL'98, is described in (6 and 7).

4. Dublin Core Metadata Element Set: Resource Page. http://purl.org/metadata/dublincore.
5. Multilingual Dublin Core: http://www.cs.ait.ac.th/~tbaker/dc-multilingual.html
6. Biagioni, S., Borbinha, J., Ferber, R., Hansen, P., Kapidakis, S., Kovacs, L., Roos, F., Vercoustre, A.M. (1998). "The ERCIM Technical Reference Digital Library". in *ECDL'98 Proceedings*, Crete, Greece, September 1998, pp.905-906 - (http://www.iei.pi.cnr.it/DELOS/EDL/ETRDL98.html).
7. ETRDL Demo Decription: Handout distributed at ECDL'98, Crete, Greece, September 1998 (http://www.iei.pi.cnr.it/DELOS/EDL/JPEG/etrdl0998.html).
8. Payette Sandra and Lagoze Carl. (1998). Flexible and Extensible Digital Object and Repository Architecture (FEDORA). In: *Research and Advanced Technology for Digital Libraries : Second European Conference, Proceedings ECDL'98*, Christos Nikolaou and Constantine Stephanidis (Eds.).- Berlin : Springer, 1998. (Lecture Notes in Computer Science, Vol. 1513). ISBN 3-540-5101-2
9. FEDORA CORBA IDL. - (http://www2.cs.cornell.rdu/payette/papers/ECDL98/FEDORA-IDL.html)

Acknowledgements

The implementation of the ETRDL is the result of a collaborative activity; the development of the Common User Interfaces was the task of IEI-CNR. The authors would like to gratefully acknowledge the assistance of the other ERCIM participants in this activity, both in the initial formulation of the specifications, and in the feedback received as a result of testing the first prototype. They would also like to thank the developers of the Dienst system and, in particular, Carl Lagoze and David Fielding for their generous assistance and advice.

INFORMATION VISUALIZATION IN THE INTERACTION WITH IDL

Maria Francesca Costabile and Giovanni Semeraro
Dipartimento di Informatica, Università di Bari
Via Orabona 4, 70125 Bari, Italy
costabile, semeraro@di.uniba.it

Abstract

We briefly discuss the state of the art of the research in information visualization. Then, we describe a technique for visualizing meta-information about the content of a networked information system in the context of a digital library, which is being developed at the University of Bari.

1. Introduction and Motivation

The computer technology is providing everybody the possibility of directly exploring information resources. Huge amount of data are becoming available on networked information systems, ranging from unstructured and multimedia documents to structured data stored in databases. On the one side, this is extremely useful and exciting. On the other side, the ever growing amount of information at disposal generates cognitive overload and even anxiety, especially in novice or occasional users.

Nowadays, wide varieties of users access, extract, and display information that is distributed on various sources, which differ in type, form and content. The current user interfaces are usually too difficult for novice users and/or inadequate for experts, who need tools with many options, thus limiting the actual power of the computer. Users need to easily understand what kind of objects are stored in the available sources they have access to, how they can retrieve and organize them along ways that permit to make rapid decisions on what is relevant and which patterns exist among objects. Users also need to manipulate the retrieved information in order to incorporate it in their specific tasks. As a consequence, networked information systems must provide enhanced user interfaces that support this intensive interaction between users and information.

In this context, the conventional interfaces, based on the view of information retrieval as an isolated task in which the user formulates a query against a homogeneous collection to obtain matching documents, are completely out of date. Indeed, this view does not correspond for several reasons to the reality of users working with networked information systems. For example, users are often unable to formulate specific questions, and they realize what they are trying to ask and how to ask it by browsing the system. This process has been called *progressive querying* in [CCL94] and *iterative query refinement* in [RPH95]. Moreover, users often consult multiple sources with different contents, forms, and methods of access.

The same holds for searching in structured relational database system, for which the SQL language has become a widespread standard. With SQL, users perform queries that specify matches on attribute values, such as author, publisher, date of publication. Each document has values for the attributes, and database management methods enable rapid retrieval even with millions of documents. Although SQL is a standard, many form-fillin variants for expressing relational database queries have been proposed to aid novice searchers. However, the diversity is itself an impediment to ease of use. Designers must take into account that users of walk-up kiosks or WWW pages cannot invest hours or minutes to learn each interface. Finding a way to provide powerful search without overwhelming novice users is a current challenge.

We recognize three different needs of people exploring information: 1) to understand the content of the information system, 2) to extract the information of interest, and 3) to browse the retrieved information in order to verify that it matches what they wanted. To satisfy such needs, the user-interface designers are challenged to invent more powerful search techniques, simpler query facilities, and more effective presentation methods. When creating new techniques, we have to keep in mind the variability of the user population, ranging from first time or occasional versus frequent users, from task-domain novices versus experts, from naive (requesting very basic information) versus sophisticated users (interested in very detailed and specific information). Since there is not a technique

capable to satisfy the need of all such classes of users, the proposed techniques should be conceived as having a basic set of features, while additional features can be requested as users gain experience with the system.

1.1 Visualizing Meta-information

Several authors agree that users interacting with huge amount of unknown and various information find extremely useful some *meta-information* on the following different aspects of the stored data [RPH95]: 1) *content*, that is, what information is stored in the source; 2) *provenance*, which refers to how the information in the source is generated and maintained, whether it is a public source or a personal archive, how frequently it is maintained, etc.; 3) *form*, i.e. the schemes for the items in the source, including their attributes and the types of values for these attributes; 4) *functionality*, that concerns the capability of the access services, such as the kinds of search supported with their performance properties; 5) *usage statistics*, that is statistics about source usage, including previous use by the same user or other ones.

One of the goals of our work is to investigate effective ways for endowing the interaction environment of networked information system with appropriate representations of the above meta-information, particularly about content, in order to provide users with proper cues for locating the desired data. The various paradigms for representing content range from a textual description of what is stored in the information source to structured representations using some knowledge representation language.

Our choice is to exploit visual techniques, whose main advantage is the capability of shifting load from user's cognitive system to the perceptual system. Indeed, information needs to be visualized in an information space in order to be retrieved by users. This visualization can either be carried out by the users in their own mind, in which case it is essentially the users' conceptualization of that information, or it could be accomplished by the system, in which case the visualization is generated on the display screen. The latter is actually called *information visualization*, and is defined as "a process of transforming information into a visual form enabling the user to observe information" [CEG97]. Recent research has proved that a suitable visualization can reduce the time to get information, and to make sense out of it. In the information system context, visualizations have a wide range of applications, they can be used for visualizing various types of meta-information, as well as queries and retrieved results.

In Section 2 of this paper, we briefly report on the state of the art of the research in information visualization. In Section 3, we illustrate a technique for visualizing meta-information about the content of a prototypical digital library service, that is currently being developed at the University of Bari. Conclusions are given in Section 4.

2. Information Visualization

As McCormick et al. say in [CDB87], "Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization offers a method for seeing the unseen. It enriches the process of scientific discovery and fosters profound and unexpected insights".

We are now all familiar with direct manipulation interfaces; their success testify the power of using the computer in a more visual manner. Direct manipulation is based on some fundamental concepts, such as the visualization of actions and objects of interest, the use of fast, incremental and reversible actions, and the immediate visualization of the result. Visual displays give the possibility of showing relationships by proximity, containment, connected lines, color coding, etc. Highlighting techniques, like blinking, brightening, reverse video, can be used to focus the attention to specific items among thousands of items. Rapid selection can be performed by pointing to a visual display.

By visually presenting information, we exploit the potentiality of visual perception of human beings. Visual presentations are particularly useful since they allow users to activate perceptual procedures to quickly obtain the desired result. Such procedures substitute the logical inferences the user should perform without a visual presentation. Moreover, by allowing dynamic user control of the visual information through direct manipulation principles, it is possible to traverse large information spaces and facilitate comprehension with reduced anxiety. In a few tenths of a second, humans can recognize features in mega-pixel displays, identify patterns and exceptions, recall related images. The use of proximity coding, color coding, size coding, animated presentation, and user-controlled selections enable users to explore large information spaces rapidly and with fun.

2.1 Information Visualization Prototypes

In our work, we are particularly interested to designing visualization tools that provide users with a rapid overview of the content of an information system. Recently, many visual query systems have been developed [CCLB97]; such systems use visual representations to depict the domain of interest and express related requests. Indeed, exploring large multi-attribute databases is greatly facilitated by presenting information visually. Among different visualization techniques of databases proposed in the literature, Ahlberg and Shneiderman have proposed starfield displays [AS94], that plot items from a database as small selectable spots (either points or small 2D figures) using two of the ordinal attributes of the data as the variables along the display axes. The shown information can be filtered by changing the range of displayed values on either axes. If this is done incrementally and smoothly, the result is zooming in and out on the starfield display, and the user can track the motion of the spots without getting disoriented by sudden, large changes in context.

The values of other attributes of the database can also be varied by the user through appropriate widgets that allow to perform dynamic queries [SWA92]. This is a very interesting visual query formulation technique (see [CCLB97] for a classification of such techniques), based on range selection, i.e. it allows a search conditioned by a given range on multi-key data sets. The query is formulated through direct manipulation of graphical widgets, such as buttons, sliders, and scrollable lists, with one widget being used for every key. The user can either indicate a range of numerical values (with a range slider), or a sequence of names alphabetically ordered (with an alpha slider). Given a query, a new query is easily formulated by moving the position of a slider with a mouse; this is supposed to give a sense of power but also of fun to the user, who is challenged to try other queries and see how the result is modified.

Higher usability is ensured if the query results fit on a single screen and are displayed quickly, i.e. within a second [AWS92]. Moreover, input and output data are of the same type and may even coincide. As a consequence, dynamic query applications typically encode multi-attribute database items as dots or colored polygons on a starfield display.

An application of dynamic queries is shown in [SWA92] and refers to a real-estate database. There are sliders for location, number of bedrooms, and price of houses in the Washington, D. C. area. The user moves these sliders to find appropriate houses. Retrieved ones are indicated by bright points on a Washington, D. C. map shown on the screen. Another interesting application that combines dynamic queries and starfield displays is FilmFinder [AS94]; it allows information about movies to be retrieved by providing names of actors, actresses, or movie directors through alphasliders, or values of other attributes through appropriate range sliders and buttons. The user can select some values by using a slider, and this first choice determines the set of values that can be selected with the remaining widgets. For example, if the user has selected a specific movie director, only names of actors and actresses who worked with that director can be selected next. This strategy is called tight coupling and it is aimed at preventing users from specifying null sets. In other words, query widgets and their related query formulation mechanisms are designed to interact with each other to avoid empty query results; this is achieved by restricting users to specify query criteria that lead to non-empty results. A tightly coupled query is then a series of filters selecting a subset of a database. For each new filter that is set, users can only select values of the remaining filters that let through at least one database object still existing after the last filter.

Dynamic queries are also called direct-manipulation queries, since they are based on the same fundamental concepts of direct manipulation illustrated above. One of the big advantages of such interaction technique is that it allows focusing the attention on the task users have to perform. Objects of interest are all displayed so that actions occur in the high level semantic domain. Each command is a comprehensible action in the problem domain whose effect is immediately visible; this relieves the user from the burden of decomposing tasks into syntactically complex sequences, thus reducing user load in problem-solving. The sliders are a good metaphor for the operation of entering a value for a field in the query: changing the value is done by a physical action instead of entering the value by a keyboard. Such action is easily reversible by moving the drag box, if the obtained results are not what users expected. No action is illegal, hence error messages are not needed. More references to work on dynamic queries can be found in [Shn94].

At Xerox PARC in the last years a group of researchers has developed several information visualizations, with the aim of helping the users understand and process the information stored into the system [RCM93, RPH95, CRY96, Car96]. They have created the "information workspaces", i.e. computer environments in which the information is moved from the original source, such as networked databases, and where several tools are at disposal of users for browsing and manipulating the information. One of the main characteristic of such workspaces is that they offer graphical

representations of information that facilitate rapid perception of the overall patterns. Moreover, they use 3D and/or distortion techniques to show some portion of the information at a greater level of detail , but keeping it within a larger context. These are usually called *fisheye* techniques [Fur86], or alternatively *focus + context*, that better gives the idea of showing an area of interest (the focus) quite large and with detail, while the other areas are shown successively smaller and in less detail. Such an approach is very effective when applied to documents, and also to graphs [SB94]. It achieves a smooth integration of local detail and global context. It has more advantages of other approaches to filter information, such as 1) zooming or 2) the use of two or more views, one of the entire structure and the other of a zoomed portion; the former approach shows local details but looses the overall structure, the latter requires extra screen space and forces the viewer to mentally integrate the views. In the *focus + context* approach, it is effective to provide animated transitions when changing the focus, so that the user remain oriented across dynamic changes of the display avoiding unnecessary cognitive load. A good example is provided by the Perspective Wall [RCM93]. For other techniques developed at Xerox PARC see [RPH95].

Numerous prototypes have been proposed for information visualization. The ones mentioned above are among those providing the most novel ideas. Shneiderman provides in Shn98 a very good survey. Other useful references are [CEG97, CC96, Cru96, GB96].

2.2 Supported Tasks in Information Visualization

There are many visual design guidelines. A central principle for information visualization might be summarized in the Shneiderman's Visual Information Seeking *Mantra "Overview* first, *zoom* and *filter*, then *details on demand*" [Shn96].

The overview allows the user to grasp the content of the application and its distribution across the different attributes. Providing an overview is particularly useful in WWW interfaces for information systems, that give users direct access to the content and interconnections within an information domain. WWW navigation should be stimulating and attractive for the users; unfortunately, due to the large amount of accessible information, the search of some detailed information can often become a long and complex activity for the user. One of the main problem is the difficulty users have in generating their mental model of the system they are interacting with; it can be difficult for them to grasp the kind of information stored and the modality for managing it. Such a problem is particularly serious since WWW interfaces are mostly used by occasional users, who are not willing to perform an in-depth study, but need to easily grasp the kind of information they can have and want to get it quickly.

Zooming is another interesting task, since users typically have an interest in some portion of a collection, and they need tools to enable them to control the zoom focus and the zoom factor. A satisfying way to zoom in is to point to a location and to issue a zooming command. Smooth zooming helps users to preserve their sens of position and context. Another popular approach for keeping the context while zooming some areas of interest is the already mentioned fisheye strategy [Fur86]; the fisheye distortion magnifies one or more areas f the display.

Users may filter out uninteresting items, so that they can quickly focus on item of interest. Dynamic queries applied to the items in the collection constitute one of the key ideas in information visualization [AS94]. Sliders, buttons, or other control widgets coupled to rapid display update are used for the filter task.

We can select an item or a group of items to get details. Once we have obtained a few dozen of items, it should be easy to browse the details about the group or individual items. The usual approach is to simply click on an item to get a pop-up window with values of each attributes. In Spotfire [IVEE], the details-on-demand window can contain HTML text with links to further information.

Besides the four tasks explicitly mentioned in the Shneiderman's Mantra, three other tasks are very useful in information visualization, namely *relate*, *history*, *extract*. Referring to the first, users can view relationship among items. In the FilmFinder details-on-demand window [AS94] users could select an attribute, such as the film's director, and cause the director alphaslider to be reset to the director name, thereby displaying only films by that director. The Table Lens emphasizes finding correlations among pairs of numerical attributes [RPH95].

We can keep a history of actions to support undo, reply, and progressive refinement. Information exploration is inherently a process with many steps, thus keeping the history of actions and allowing users to retrace their steps is important. Currently, mostly prototypes fail to deal with this requirement.

It is also useful to allow extraction of sub-collections and of the query parameters. Once the users have obtained the item or the set of items they desire, it would be useful for them to be able to extract that set and to store into a file in a format that would facilitate other uses, such as sending by e-mail,

printing, inserting into a presentation package. As an alternative to saving the result set, they might want to save the settings for the control widgets. At the moment, few prototypes support this extract task.

3. An Approach to Visualizing Information Content

The work reported in this section refers to a WWW-based user interaction environment, focusing on a novel technique for visualizing the content of the information system of a digital library, called IDL, currently developed at the University of Bari [SEMF97, EMSF98]. More details on the overall interaction environment of IDL can be found in [CESF98].

A first prototype of IDL has been equipped with a WWW interface exploiting a typical form fill-in interaction style. That interface is powerful and flexible since it permits a search by a combination of fields, but it is more appropriate for users who are already acquainted with the library structure, and also have some information about the library content. By observing casual users interacting with the IDL prototype, we realized that often users performed queries whose result was null, just because they did not have any idea of the kind of documents stored in the library. Therefore, we decided to enrich the IDL interaction environment by developing some novel visual tools, that aim at allowing users to easily grasp the nature of the information stored in the available sources and the possible patterns among the objects, so that they can make rapid decisions about what they really need and how to get it.

3.1 The Topic Map

One of the new features of the IDL environment, that users appreciate the most, is the possibility of getting a rapid overview of the content of the stored data through the *topic map*. Such a visualization is actually an *interactive dynamic map* (interactive map for short), as it has been proposed in [ZB95]. An interactive map gives a global view of either the semantic content of a set of documents or the set of documents themselves. The semantic content reflects the topics contained within the set of documents and the way they are organized to relate to each other; it is represented by a thesaurus that is built automatically from a full-text analysis.

Interactive maps exploit the metaphor of exploring a geographic territory. A collection of topics, as well as a collection of documents, is considered to be a geographical territory that contains resources, which metaphorically represent either topics or documents; maps of these territories can be drawn, where regions, cities, and roads are used to convey the structure of the set of documents: a region represents a set of topics (documents), and the size of the region reflects the number of topics (documents) in that region. Similarly, the distance between two cities reflects the similarity relationship between them: if two cities are close to each other, then the topics (documents) are strongly related (for example, documents have related contents).

Topic maps are very effective since they provide an overview of the topics identified in a collection of documents, their importance, and similarities and correlations among them. The regions of the map are the classes of the thesaurus, each class contains a set of topics represented by cities on the map. Roads between cities represent relationships between topics. In this way, topic maps provide at a glance the semantic information about a large number of documents. Moreover, they allow users to perform some queries by direct manipulation of the visual representation.

Document maps represent collections of documents generated from a user query, that may be issued on the topic map by selecting regions, cities, and roads. The cities of these maps are documents, and they are laid out such that similar or highly correlated documents are placed close to each other.

In order to generate the topic map in IDL, we need to identify the set of topics or descriptors defining the semantic content of the stored documents; such topics constitute the IDL *thesaurus*. There are several thesauri used in the information retrieval literature; most of them are built manually and their descriptors are selected depending on specific goals. An example is the Roget's thesaurus, that contains general descriptors. When building the IDL thesaurus, we have used standard techniques, also taking into account the type of documents currently stored in IDL. They are scientific papers that have been published in the journal IEEE Transactions on Pattern Analysis and Machine Intelligence (*pami*), in the Proceedings of the International Symposium on Methodologies for Intelligent Systems (*ismis*), and in the Proceedings of the International Conference on Machine Learning (*icml*). Therefore, we have used the INSPEC thesaurus containing specific terms in the field of Artificial Intelligence. This thesaurus contains 629 keywords, that are either single words or expressions made up of more words (up to five).

We have represented documents and keywords (topics) by vectors, that is a common practice in information retrieval [SG83, Lar91]. The coordinates of the document vectors and those of the topic vectors are computed in the following way: the coordinate d_i of the vector representing document D is 1 if the topic T_i was found in D, and 0 otherwise; the coordinate t_i of the vector representing topic T is 1 if document D_i contains T, and 0 otherwise.

In the IDL maps, we have implemented some color based coding techniques and we have added several widgets to the original project described in [ZB95], in order to obtain a more effective visualization and to provide some mechanisms appropriate for a flexible interaction in a data-intensive context, such as a digital library. Details on the map visualizations and on the user interaction with them are in [CESF98].

3.2 Some Related Work

The IDL visual interaction environment has been influenced by recent research on information visualization, which turned out to be a way of improving the intensive interaction between users and information [CEG97, CC96, Cru96, GB96]. In [RPH95], a variety of studies, tools, and systems developed at Xerox PARC illustrates the style of rich interaction that users will have with digital libraries. The technique we have proposed is a contribution in that direction.

The topic map has been designed with the aim of assisting those users who are unable to formulate specific questions, and that can realize what they really want and how can retrieve it only by browsing the system. The process of formulating a query progressively, i.e. step by step, by first asking general questions, obtaining preliminary results, and then revisiting such outcomes to further direct the query in order to extract the result the user is interested in has been called *progressive querying* in [CCL94]. The idea is similar to the *iterative query refinement* in [RPH95]. It is an interesting feature a query system should have, and it has been exploited by the visual query tools implemented in IDL.

The interactive dynamic maps proposed in [ZB95] were the main source of inspiration for the topic map, as we have described above. However, in the IDL environment we have improved the visualization by exploiting some color-based coding techniques and added several widgets with the aim of providing interaction mechanisms suitable for a data intensive context, such as that one of online digital libraries. Furthermore, in our system the topic map allows users to perform queries by direct manipulation, a facility that was not very well developed in [ZB95].

The interaction through the topic map is in accordance with Shneiderman's Visual Information Seeking Mantra "Overview first, zoom and filter, then details on demand" [Shn96]. We have implemented mechanisms for zooming and filtering working on the overview provided by the topic map. Once some documents have been retrieved, a simple click on a document icon in the document map will provide detailed information on that document, up to a complete view of the whole document.

4. Conclusions

Networked information systems pose several demands due to the large number and variety of its users, and also to the nature of the stored data, that is distributed on autonomous information sources that differ in content, form, and type. One of the consequences is that they must be equipped with environments that permit a new style of rich interactions with such information-dense systems. The work presented in this paper is a contribution in this direction.

We are aware that usability is an extremely important requirement for applications having a large variety of users, such as digital libraries. The testing should be done with representatives of each of the primary user communities and of as many of the secondary communities as time and money allow. Some preliminary usability evaluation of the IDL interface has been already performed, and the topic map has been designed in order to overcome some problems detected while observing the interaction of novice users. We are currently planning more accurate usability testing to be performed in the near future.

References

[AS94] Ahlberg C., Shneiderman, B., «Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays», *Proceedings ACM CHI'94*: Human factors in computing systems, pp. 313-317.

[AWS92] Ahlberg C., Williamson C., Shneiderman B., «Dynamic Queries for Information Exploration: An Implementation and Evaluation», *Proceedings ACM CHI'92*: Human Factors in computing Systems, pp. 619-626.

[BRS92] Botafogo R., Rivlin E., Shneiderman B., «Structural analysis of hypertexts: Identifying hierarchies and useful metrics». *ACM Trans. Inf. Syst*. 10, 2, 1992, pp. 142-180.

[Car96] Card S.K., «Visualizing Retrieved Information: A Survey», *IEEE Computer Graphics and Applications*, March 1996, pp. 63-67.

[CC96] Catarci T., and Cruz I.F., «Special Issue on Information Visualization», *ACM Sigmod Record*, 4, 1996, p. 25.

[CCL94] Chang SK, Costabile M.F, Levialdi S., «Reality bites – progressive querying and result visualization in logical and VR spaces», *Proc. Of IEEE Symposium on Visual Languages*. St. Louis, 1994, pp 100-109.

[CCLB97] Catarci T., Costabile M. F., Levialdi S., Batini C., «Visual Query Systems for Databases: a Survey», *Journal of Visual Languages and Computing*, Vol. 8, 1997, pp. 215-260.

[CDB87] McCormick B.H., DeFanti T.A., Brown M.D., (Eds) «Visualization in Scientific Computing and Computer Graphics», *ACM SIGGRAPH*, 21, 6, 1987.

[CEG97] Card S., Eick S. G., and Gershon N., (1997) «Information visualization», *CHI97 Tutorial Notes*, Atlanta, GA, March 1997, pp. 22-27.

[CESF98] Costabile M.F., Esposito F., Semeraro G., Fanizzi N., Ferilli S., «Interacting with IDL: The Adaptive Visual Interface», in C. Nikolaou and C. Stephanidis (Eds.), *Research and Advanced Technology for Digital Libraries, Lecture Notes in Computer Science 1513*, Springer, Berlin, 1998, pp. 515-534.

[Cru96] Cruz I.F., «Taylorable information visualization», *ACM Computing Surveys*, 28A(4), 1996.

[CRY96] Card S. K., Robertson G.G., York W.,«The WebBook and the Web Forager: An Information Workspace for the World-Wide Web», *Proceedings CHI'96*, April 13-18, 1996, pp. 111-117.

[EMSF98] Esposito F., Malerba D., Semeraro G., Fanizzi N., Ferilli S., «Adding Machine Learning and Knowledge Intensive Techniques to a Digital Library Service», *International Journal on Digital Libraries*, Springer-Verlag, Berlin, 2, 1998, in print.

[Fur86] Furnas G.V., «Generalized Fisheye Views», *Proceedings CHI'86 Conference*: Human Factors in Computing Systems, ACM Press, New York, pp. 16-23.

[GB96] Gershon N., and Brown J. R., «Special Report on Computer Graphic and Visualization in the Global Information Infrastructure», *IEEE Computer Graphics and Applications*, 16(2), 1996, pp. 60-75.

[IVEE] IVEE Development, «Spotfire», Goteborg, Sweden, `http://www.ivee.com/pict/Spotfire-film1.gif`.

[Lar91] Larson, R.R., «Evaluation of retrieval techniques in an experimental on-line catalog», *JASIS*, 43, 1991, pp. 34-53.

[RCM93] Robertson G.G., Card S.K., Mackinlay J.D., «Information Visualization Using 3D Interactive Animation», *Communications of the ACM*, 36 (4), 1993, pp. 7-71.

[RPH95] Rao R., Pedersen J.O., Hearst M.A., Mackinlay J.D., Card S.K., Masinster L., Halvorsen P.-K., and Robertson G.G., «Rich interaction in the digital library», *Communications of the ACM*, 38 (4), 1995, pp. 29-39

[SB94] Sarkar M., Brown M.H., « Graphical Fisheye Views», *Communications of the ACM*, 37(12), 1994, pp. 73-84.

[SEMF97] Semeraro G., Esposito F., Malerba D., Fanizzi S., "Machine Learning + On-linr Libraries = IDL", in C. Peters and Thanos (Eds.), *Research and Advanced Technology for Digital Libraries, Lecture Notes in Computer Science 1324*, Springer, Berlin, 1997, pp. 195-214.

[SG83] Salton G., and McGill M.J., «Introduction to Modern Information Retrieval», New York, NY, McGraw-Hill, 1983.

[Shn94] Shneiderman B., "Dynamic Queries for Visual Information Seeking", *IEEE Software*, 11, 1994, pp. 70-77.

[Shn96] Shneiderman B., «The eyes have it: A task by data type taxonomy for information visualization», *Proceedings of 1996 IEEE Symposium on Visual Languages*, Boulder, Colorado, September 1996, pp. 336-343.

[Shn98] Shneiderman B.: «Designing the User Interface». Addison-Wesley, New York, 1998.

[SWA92] Shneiderman B., Williamson C., Ahlberg C., "Dynamic Queries: Database Searching by Direct Manipulation", *Proceedings ACM CHI'92*: Human Factors in computing Systems, pp. 669-670.

[ZB95] Zizi M., and Beaudouin-Lafon M. ,«Hypermedia exploration with interactive dynamic maps», *International Journal on Human-Computer Studies*, 43, 1995, pp. 441-464.

AQUA: An advanced user interface for the Dienst digital library system

László Kovács, András Micsik, Balázs Pataki

MTA SZTAKI
Computer and Automation Research Institute
of the Hungarian Academy of Sciences
Department of Distributed Systems
H-1111 Budapest XI. Lágymányosi u. 11. Hungary
{laszlo.kovacs, micsik, pataki}@sztaki.hu

Abstract

Dienst system is widely used for providing distributed digital library services. A new user interface for Dienst has been built as part of the AQUA project to enhance the capabilities of presently available Dienst user interfaces. This new interface is presented, and its features for interactive content exploration are described.

1 Introduction

Digital libraries are a promising and also highly required new type of network services. In the area of computer science both USA and Europe make a significant effort for creating a distributed digital library service. The American service is called Networked Computer Science Technical Reports Library (NCSTRL) [4], while the European one is the ERCIM Technical Reference Digital Library [5]. Currently these services offer more than 20,000 technical reports for the research community from more than 60 institutions, and it is constantly growing.

The underlying software is in both cases the Dienst distributed digital library system [3]. The system is accessible through any standard WWW browser, and offers searching, browsing and downloading functionality in a transparent way for the distributed document repositories. Repositories are managed by the publishing institutions themselves. SZTAKI has been serving as a regional center for the joint NCSTRL-ETRDL service since 1996 [2]. The Department of Distributed Systems (in charge of Dienst issues within SZTAKI) has built a prototype of a new user interface for the NCSTRL-ETRDL service which is introduced in this paper.

2 User interfaces for Dienst

Currently there exist several user interfaces for the Dienst system, including an interface for library users, and an interface for librarians. Both interfaces operate as a form-based WWW service. The interface for library users has several flavours: one for the ETRDL, and two different versions for NCSTRL. These interfaces offer *searching* by different fields such as author, title or abstract (ETRDL can also search for keyword and language fields), *browsing* by year of publication or by author name, and *document views* in several formats. However the basic user interaction method is the same in all these interfaces: the user formulates a query, gets a list of documents as result, and downloads documents in the list one at a time.

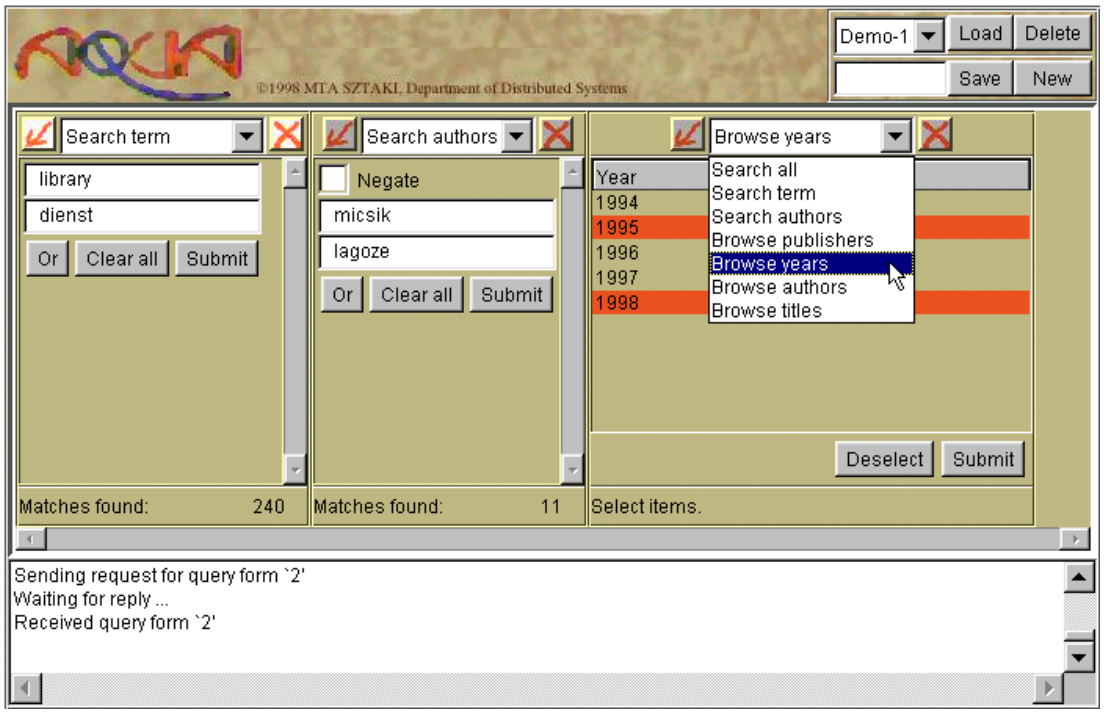


Figure 1: First part of example query

Searching and browsing functionality is totally separated, and the only way for refining a query is to return to the previous query formula, change and reapply it.

3 The new AQUA user interface for Dienst

AQUA stands for Advanced Query User Interface Architecture [1], and its main goal is to provide a generic, powerful and easily usable interface for digital libraries and Internet search engines. Currently AQUA has been applied to the functionality of the Dienst digital library system.

In this user interface a compound query expression is visualized as a series of panels in the middle part of the window (Fig. 1). Each panel represents one piece of the query expression, and performs that expression on the output result of the previous panel. In the example in Fig. 1 the first panel contains a search for terms "library" or "dienst", and this query has 240 documents as result. The second panel refines the first query by searching only for authors "Lagoze" or "Micsik" within that 240 documents, and the 11 documents found are browsed by publication year in the third panel. The example continues on Fig. 2 where the remaining document titles are listed after selecting years 1995 and 1998. Finally in the last panel a selected document is shown. The document view contains part of the bibliography record and a download menu, where the document can be downloaded in the selected format. At this point the user can choose to return to the document view of the original Dienst user interface as well.

Panels in the list form a so called query chain. The user can freely modify, grow or shrink this query chain. After every modification panels are updated according to the user's changes, and the actual results of the query chain are visualized for the user. Through this process the user can interactively explore the contents of the digital library. These query chains can be saved and later reused. Each user can have individual preferences, and an own set of saved query chains. Manipulation of query chains can be done in the upper right corner of the

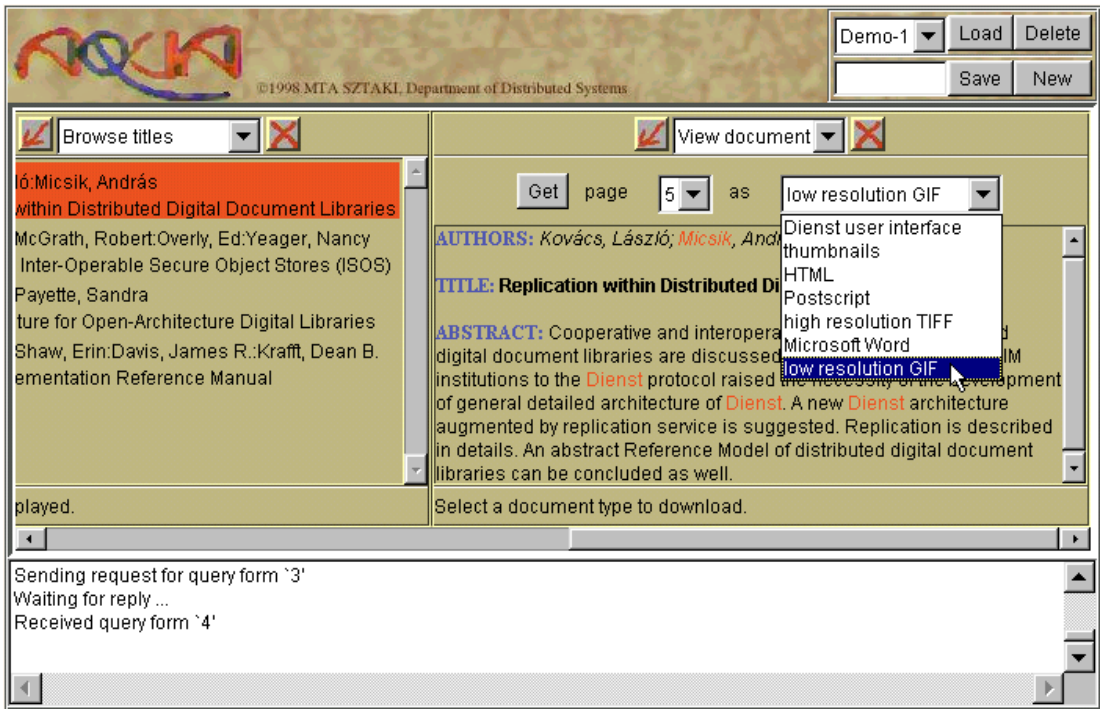


Figure 2: Second part of example query

window. The lower part of the window is a kind of console, where running processes can be tracked, and where errors are printed.

4 Implementation issues

This AQUA interface is implemented as a Java applet and a server (also written in Java). We followed the "thin client – fat server" principle, which gave us several advantages. The main advantage is that the client applet is more robust, because it needs less memory, and thus it can put stress on the quality of visualization instead of information retrieval and processing.

The server contains the following parts:

- *Wrappers* connect different digital library services to the AQUA server. This means that several digital library or database services can be accessed through AQUA at the same time, even using different query languages or protocols. Both session-oriented and sessionless services may be connected to the AQUA server.
- *Metadata* describing the properties of connected digital library services.
- *User database* containing the users' preferences and saved queries.
- *Query chain manager* translates user actions into actual queries for the digital library services, and generally cares for the consistency of the query chain visualization.

The current implementation relies on the functionality of the freewais-sf software as indexer service for the Dienst system. In order to shorten response times the AQUA server directly addresses queries to the freewais-sf search engine.

5 Summary

The new AQUA user interface for Dienst provides all the searching, browsing and document viewing functionality that is present in the traditional Dienst user interfaces, but integrates these functionalities into a new user interface paradigm. This paradigm is based on the use of query chains.

Further work will be done to enhance the comfort of the user by adding more possibilities for personalization and introducing an automatic notification service.

References

[1] AQUA project homepage, URL: http://www.sztaki.hu/sztaki/aszi/dsd/aqua/

[2] Dienst server at SZTAKI, URL: http://www.sztaki.hu/dienst/

[3] C. Lagoze, J. R. Davis: "Dienst: an Architecture for Distributed Document Libraries", Communications of the ACM, 38 (4) April 1995

[4] Networked Computer Science Technical Reports Library (NCSTRL), URL: http://www.ncstrl.org

[5] S. Biagioni, J. Borbinha, R. Ferber, P. Hansen, S. Kapidakis, L. Kovács, F. Roos, A. M. Vercoustre: "The ERCIM Technical Reference Digital Library", ECDL'98, Heraklion, Greece, September 1998

[6] L. Kovács, A. Micsik: "Replication within Distributed Digital Document Libraries", Proceedings of the 8th ERCIM Database Research Group Workshop on Database Issues and Infrastructure in Cooperative Information Systems, Trondheim, Norway, 1995

Iterative Information Retrieval Using Fast Clustering and Usage-Specific Genres

Jussi Karlgren, Ivan Bretan, Johan Dewe, Anders Hallberg, Niklas Wolkert
SICS, University of Helsinki, and Telia Research AB

Abstract

This paper describes how collection specific empirically defined stylistics based genre prediction can be brought together together with rapid topical clustering to build an interactive information retrieval interface with multi-dimensional presentation of search results. The prototype presented addresses two specific problems of information retrieval: how to enrich the information seeking dialog by encouraging and supporting iterative refinement of queries, and how to enrich the document representation past the shallow semantics allowed by term frequencies.

Searching For More Than Words

Today's tools for searching information in a document database are based on term occurrence in texts. The searcher enters a number of terms and a number of documents where those terms or closely related terms appear comparatively frequently are retrieved and presented by the system in list form. This method works well up to a point. It is intuitively understandable, and for competent users and well edited document bases it will give a predictably mediocre result. It also has obvious drawbacks.
Term frequencies provide a representation of document content which suffers from being both shallow and sparse: shallow, in that a text has more facets and features than a list of terms will be able to represent; and sparse, in that it usually is difficult for users to pinpoint which term to search for in a multi-laced hierachy of hyponymial relations and sets of near synonyms, related terms, or other variants.

We focus on two specific drawbacks of the traditional information retrieval search process. Firstly, as has been acknowledged by many recent research projects and some recent applications, searches are seldom one-shot affairs. Typically a search is improved and refined iteratively, until the retrieved set seems good enough, by some metric. Thus, the interface should support persistence and incremental refinement. Secondly, the objects of study are more complex than usually is assumed: document topic is more than term frequencies, and documents are more than the topic they are about.

Information Seeking Dialog

Dialog design is not one of the central fields of computational linguistics; most design principles have been made outside the field, in by researchers generally interested in human-computer interaction. This is a pity, for two reasons. First, linguists -- even computational -- have a lot to offer dialog design: language has a lot to do with dialog. Second, language technology applications need dialog design as well as parsing algorithms or terminology statistics.

We have put some thought in our dialog design. We want our system to transcend the typical one-shot dialogs of most fielded information systems today: ``enter your search terms'' -- ``browse the resulting list''. Our dialog builds on incremental specialization. The pipeline

architecture of the system allows users to specialize a query by specializing the resulting set to only some of the clusters retrieved initially.

The query and the resulting clusters are represented continuously: retrieved documents are inserted in the appropriate cluster irrespective of ranking. Specialization of a query can be done without figuring out specialized terms, since the enriched representation of documents can be used instead. The clusters themselves represent topic and genre: they can be selected for further inspection.

Nowhere is there a list format - it should be uninteresting to judge documents by internal processing order if topic and genre are displayed instead.

Seeking information is seldom done with a clear picture of the goal in one's mind. Typically users will need to familiarize themselves with the available materials to find how they will express their information need to the system -- by some preliminary searches, some back-tracking, and some reformulation of previous tries. An information retrieval interface should support persistent and manipulable dialog objects which represent the system's understanding of the search.

Topic and Document Representation

What effect a search term has in terms of retrieval is usually surprising to a user. The documents it retrieves may vary: if the term or term set is broad, by topic; if the topic or set of topics is broad, by style. An information retrieval interface should not try to shield the user from this variation but display it and allow the user to act on the results, iteratively.

Interaction - A Prototype Implementation

The assumption of interface designers is often that users cannot cope with complex interfaces and that the goal is to reduce the amount of information users are subjected to. We believe this not to be true. If the information is presented in a manner well founded in the task and usage context users will accept it.

We have built a fully functional prototype system, DropJaw, to experiment with iterative search. Our experiment database is the World Wide Web, and DropJaw bases its searches for web documents by the user entering terms, as in a traditional system. Rather than producing ranked lists of output based on term occurrence, DropJaw displays the distribution of the resulting set over two dimensions: dynamically generated topical clusters and user-defined, document-base oriented genre. The two-dimensional document space is displayed on a work board or matrix for further user processing.

DropJaw consists of two main modules: Easify, a presentation module, and Chunkify, a classification and clustering engine. DropJaw is implemented in C++ and runs as a stand-alone application under Microsoft Windows. It does not include a indexing or search engine -- currently we make use of any of several commercial search engines as indexers. DropJaw retrieves the documents the indexing service returns and reanalyzes them locally. The architecture is a pipeline model, where different processing components in Chunkify can be connected or disconnected -- by user requests to Easify -- to the pipeline
which delivers a stream of documents from the World Wide Web to the user.

Appearance

The main goal for the graphic design and the interaction design has been to build an interface which appeals to the untrained Internet user and is interesting and fun to use.

In most other cases information visualization tools have a somewhat scientific look and support an interaction designed for specialist users. Most are graphically designed with mainly the function in mind, rather than aesthetic quality. This most likely is a result of using standard elements from standard programming tools and little or no involvement of graphic or industrial designers in the design process.

Information visualization before the computer era (Tufte, 1983), in many cases is aesthetically very beautiful and pleasant to look at -- without compromising clarity or usability. There is no reason why the introduction of computers into the information visualization process should compromise aesthetics. Lack of aesthetic qualities may distract and bore the user and indirectly cause the loss of potentially useful information; usability is a function of both underlying functionality and aesthetic quality.

The graphic interface, Easify, is designed to feel playful and fun but without sacrificing a sense of strict reliability. Because of the high information density in the interface, colors have been used to help user choice, without dominating the interface look.
The use of strong colors and extreme shapes many times bores users after extensive use; this interface has a combination of soft, sober colors and simple but interesting shapes built from carefully chosen and composed non-standard graphic elements.

Functionality

The prototype was designed with a high degree of interactivity in mind. Users initiate the interaction by entering a query and clicking the search button. Easify sends the current parameter set -- number of initial clusters, i.a. -- to Chunkify, and starts the background pipeline. Chunkify consults the indexing engine for a list of likely documents, and starts retrieving candidate documents from the Internet. After initial clustering, Chunkify starts delivering documents to Easify for presentation.

The pipeline design leaves the user in control of the interface at all times instead of locking the interaction: there are indicators to show that the classification engine is running in the background. A stop button lets the user halt the background processes;
a clear button clears the current document set from the display.

There are many choices that can be made at each step of processing: the user can drag and drop subsets of the presented document set to a regrouping panel to request Chunkify to regroup the clusters. This does not stop the first filter in the pipeline from continuing its work: DropJaw simply adds a finer-grained Chunkify filter to the end of the pipeline. The first set keeps accruing, and can be returned to if the finer analysis turns out less useful. More information about the documents can be found in pop up menus, and the documents themselves are available for perusal at all times.

Document Representation

Rather than defined solely by their list ranking, Easify represents documents as members of topically and stylistically homogenous clusters in the interface.

Stylistic variation among texts shows through stylistic items: observable choices of linguistic items. Stylistic items can be observed on any level of linguistic abstraction: lexical, for the choice between words of similar meaning but different connotations; syntactic, for the choice between equivalent constructions with different communicative import; textual, for decisions of textual organization. Each stylistic item is of little import in itself, but taken together the entire set is indicative of systematic differences. A set of documents with a perceived

consistent tendency to make the same stylistic choices is called a genre or, specifically, if it has an established communicative function, a functional style (see e.g. Enkvist, (1973). Stylistic variation between genres or language varieties can be detected reliably using a large battery of quite simple stylistic items such as pronoun counts or relative frequencies of certain types of constructions such as agentless passives (Karlgren and Cutting, 1994), utilized for authorship determination by simple calculations of average word length distributions (Mendenhall, 1887), and with some success predictively for information retrieval (Harman, 1996).

Balanced Corpora for Testing

There is no well-established genre palette for Internet materials, such as one can find for printed documents. We need to create one to know what test materials to collect. This involves the risk of circular evaluation of self-established criteria for success, the interpretation of vaguely expressed and imperfectly understood user expectations, and the need to face the very real engineering problem of putting genre distinctions to predictive use for retrieval purposes.

In most computational stylistics, genre has mostly been equated or based on text source: Wall Street Journal text archive, personal letters, technical documentation (Francis and Kucera, 1982; Källgren, 1990; Karlgren and Cutting, 1994). We find this unsatisfying, and wish to find a better foundation for analysis; we believe user perceptions are central to this task. We have built our genre palette (see Table 1) through interviewing users: trying to define genres that are both reasonably consistent with what users expect and observable and conveniently computable using measures of stylistic variation as outlined in the previous section. This work is d escribed in a previous report (Dewe et al, 1998).

Genre		**Sample**
1	Informal, Private Personal home pages.	128
2	Public, commercial Home pages for the general public.	197
3	Interactive pages Pages with feed-back: customer dialogue; searchable indexes.	73
4	Journalistic materials Press: news, editorials, reviews, popular reporting, e-zines.	94
5	Reports Scientific, legal, and public materials; formal text.	113
6	Other running text	160
7	FAQs	12
8	Link Collections	148
9	Other listings and tables	225
10	Discussions Contributions to discussions; Usenet News material.	24
11	Error Messages	184
Total		**1358**

Table 1: The current genre palette

Recognizing genres automatically

The genre palette, besides being intuitively understandable, needs to be workable for automatic analysis. We calculate a quite large number of textual features for each individual text and work them together for a categorization decision using a machine learning algorithm. The pioneering work by Douglas Biber(1989) on computational corpus-based stylistics has been descriptive rather than predictive, aiming to find distinctions between different registers opr varieties of spoken and written language. It has made use of large numbers of stylistic features collected from previous, non-computational work and weighing them together using standard methods from multivariate statistics. We use this work as a basis for ours. Most of Biber's features we use here are rather lexical in nature, for ease of processing: the relative frequency of certain classes of words such as personal pronouns, emphatic expressions, or downtoning expressions, for instance. We add more general textual and genre specific features: relative number of digits, or average word length, for instance. Others yet are vectored specifically to the Internet material we have been using for experimentation: number of images or number of HREF links in the document, for instance. We normalize the measurements by mean and standard deviation, and combine them -- 40 of them, at present -- into simple if-then categorization rules using C4.5, a non-parametric categorization tool (Quinlan, 1993).

If
- there are more "because" than average,
- longer words than average,
- type-token ratio is above average,
then
- the object is of class Textual
with
- a certainty of 90.0%.

We have a few dozen rules to categorize texts into one of the eleven genres defined in the above sections. The genres partition into two major hypercategories: textual (04, 05, 06, 07, 10) and non-textual (01, 02, 03, 08, 09, 11); each of them in turn splits to one of five or six sub-categories. These splits are of varying quality: the first does quite well, something like a ninety per cent success rate, while the subsplits make the wrong choice somewhere between once in three or four times.

With additional features and a better defined genre palette results will improve. However, to get really useful results the categorization should not be exclusive. Every object should potentially be of several genres.

Clustering By Content -- A Very Simple Clustering Algorithm

The similarity measure for comparing topical document representations with each other and with cluster centroids is in most respects based on a standard tfc metric, as defined by Salton and Buckley (1982): standard term frequencies, cosine length normalization, and a standard collection frequency (idf) measure to factor in collection and domain specific terminology variation.

Since the emphasis is on a high degree of interactivity, a quick and dirty clustering must be used for the initial document sets. We work on the assumption that low number, up to 5, of clusters in the interface is desirable (Hearst and Pedersen, 1996). Since the search engine

itself is not included in the system setup, there is no time to wait for all the data to arrive. If the algorithm would have recourse to the entire document set, the initial clusters could be formed from a random set, as in Scatter/Gather (Cutting et al, 1992); in our case the first clustering must proceed on the assumption that the first documents to arrive are a representative subset of the entire retrieved set. This is a daring assumption and most likely overly optimistic, but enables us to start clustering sooner, and to restrict the use of the computationally expensive -- on the order of N squared -- hierarchical clustering by defining the first clusters on a small number of documents: the first 10-50 documents, which number is adjustable in the interface. The clustering itself is a variant of the standard metrics: a hierarchical agglomerative group-average algorithm (e.g. Jain and Dubes, 1988).

After deciding the first i clusters (with i adjustable in the interface) the following documents are each routed to one of them. A simple assign-to-nearest algorithm is used to decide cluster membership. The clusters are represented by their centroids -- the N-dimensional centre points -- as a list of the terms with highest weight, and the matching distance is the same as for the hierarchical clustering algorithm above. Currently the clusters are not refined on the fly; this would be easier to implement with the entire clustering stage coupled tighter to the search engine itself -- these algorithms are substitutes for an integrated system.

Evaluating The Method And The Design

Developer satisfaction

With rather small effort, this both user- and technique-centered development method has pinpointed certain weaknesses in the design while at the same time encouraging us to pursue its strong sides further. Users liked the basic idea, and since most components are based on empirically evaluated knowledge, they proved immediately useful.

Subject satisfaction

The clustering algorithms are very simple; the genre classification just barely flies; some interface niceties not completely in place yet -- but the prototype runs well enough for the interaction to be tested and evaluated and the design to be refined.

We gave a small number of subjects two simple retrieval tasks each, one using Easify and the other using Altavista. The order between questions and interfaces was varied between subjects.

12 subjects were chosen to be a representative test population. 6 subjects were male, 6 female; all of age 25-30; and while averages are a poor measure for small populations, in this case the subject pool hit the mark quite well: they assessed themselves as averagely experienced internet who use search services occasionally, and have a reasonable understanding of how search services work.

The tasks were

- Find an album or a concert review about Oasis.
- Find a list of hotels on Malta.

The subjects were given an introduction to the main ideas with Easify, and shown an example search with the system: all users had used Altavista previously. They were then given the tasks. The experiment supervisor gave tips on query formulation if the subjects seemed to get stuck. The subjects used about 5 minutes on each task.

The subjects did not do well on the tasks: Altavista search produced better results. In spite of this, the subjects liked and understood the interface prototype, with some remarks. Some subjects were confused by the changing cluster texts, but were comfortable with it after a short explanation. Some wanted the genres to be refined as well, to sub-genres. In spite of the low response times, all but one of the subjects liked the interface in itself.

Most subjects used the interface as we had intended them to, and many searched for documents in the genres we intended the results to show up in. The genre determination algorithm in its current state makes a mistaken choice too often for some of the genres: the hit rate must be raised to nine out of ten for the concept to work better. Flexible genre determination would help here -- a document should be allowed to fall into several genres rather than exclusively one.

Better cluster headings would improve subjects' chances of finding their way through the document space. This will be absolutely necessary if Easify is to work with very large document bases, where inspection of documents will be impossible until after a large number of categorization iterations.

The search engine only delivers the 200 top ranked documents for each query, and this gives the formulation of the query too large weight in determining if a correct answer can be found. With a closer integration this bottleneck can be eliminated.

With these improvements -- all of them included in future planned project work and none of them surprising or dismaying to the design team -- Easify will be able to fulfil its promises to compete successfully with single-shot single-modal single-channel search machinery.

References

Douglas Biber. 1989. "A typology of English texts", Linguistics, 27:3-43.

Douglass R. Cutting, David R. Karger, Jan O. Pedersen, John W. Tukey. 1992. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Copenhagen. New York: ACM.

Johan Dewe, Ivan Bretan, and Jussi Karlgren. 1998. "Assembling a Balanced Corpus from the Internet". 11th Nordic Computational Linguistics Conference, Copenhagen. Copenhagen University.

Nils Erik Enkvist. 1973. Linguistic Stylistics. The Hague: Mouton.

W. N. Francis and F. Kucera. 1982. Frequency Analysis of English Usage. Houghton Mifflin.

Donna Harman (ed.). 1996. The Fourth Text REtrieval Conference (TREC-4). National Institute of Standards Special Publication 500-236. Washington.

Marti A. Hearst and Jan O. Pedersen, Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. 1996. Collections. Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich. New York: ACM.

Anil K. Jain and Richard C. Dubes. 1988. Algorithms for Clustering Data. Engelwood Cliffs, New Jersey: Prentice Hall.

Jussi Karlgren. 1996."Stylistic Variation in an Information Retrieval Experiment" In Proceedings NeMLaP 2, Bilkent, September 1996. Ankara: Bilkent University. (In the Computation and Language E-Print Archive: cmp-lg/9608003).

Jussi Karlgren and Douglass Cutting. 1994. "Recognizing Text Genres with Simple Metrics Using Discriminant Analysis", Proceedings of the 15th International Conference on Computational Linguistics (COLING 94), Kyoto. (In the Computation and Language E-Print Archive: cmp-lg/9410008).

Gunnel Källgren. 1990. The First Million is Hardest to Get: Corpus Tagging. Proceedings of the 13th International Conference on Computational Linguistics (COLING-90) Hans Karlgren (ed.), Helsinki.

T.C. Mendenhall. 1887. "The Characteristic Curves of Composition." Science 9: 237-49.

J. Ross Quinlan. 1993. C4.5: Programs for Machine Learning. San Mateo: Morgan Kaufmann.

Gerard Salton and Christopher Buckley. Term-Weighting Approaches in Automatic Text Retrieval. Information Processing and Management 24 (5) 513-23.

Tomek Strzalkowski, Louise Guthrie, Jussi Karlgren, Jim Leistensnider, Fang Lin, Jose Perez-Carballo, Troy Straszheim, Jin Wang, Jon Wilding. 1996. "Natural Language Information Retrieval: TREC-5 Report" Proceedings of The Fifth Text REtrieval Conference (TREC-5). Donna Harman (ed.). National Institute of Standards Special Publication. Washington.

Edward R. Tufte. 1983. The Visual Display of Quantitative Information.

The DELOS Working Group is a part of the ESPRIT Long Term Research Programme (LTR No. 21057) and is managed by ERCIM.

The DELOS Partners are:

ERCIM members:

CLRC, CWI, CNR, FORTH, GMD, INRIA, INESC, SICS, ETH-SGFI, SINTEF Telecom and Informatics, MTA SZTAKI, VTT

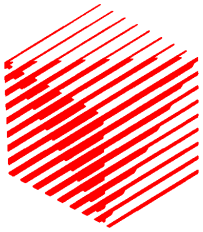
Non-ERCIM members:

University of Michigan, USA
Elsevier Sciences, The Netherlands

For additional information, please contact
Costantino Thanos
Istituto di Elaborazione della Informazione, Consiglio Nazionale delle Ricerche
Via Santa Maria 46
I-56126 Pisa
Tel: +39 50 593492, Fax: +39 50 554342, E-mail: thanos@iei.pi.cnr.i

DELOS web site: **http://www.iei.pi.cnr.it/DELOS/**

ISBN 2-912335-07-8



The European Research Consortium for Informatics and Mathematics (ERCIM) is an organisation dedicated to the advancement of European research and development, in the areas of information technology and applied mathematics. Through the definition of common scientific goals and strategies, its national member institutions aim to foster collaborative work within the European research community and to increase co-operation with European industry. To further these objectives, ERCIM organises joint technical Workshops and Advanced Courses, sponsors a Fellowship Programme for talented young researchers, undertakes joint strategic projects, and publishes workshop, research and strategic reports as well as a newsletter.

ERCIM presently consists of fourteen research organisations from as many countries:

Central Laboratory of the Research Councils

Rutherford Appleton Laboratory
Chilton, Didcot
GB-Oxon OX11 0QX

Tel: +44 1235 82 1900
Fax: +44 1235 445 385
http://www.cclrc.ac.uk/

Centrum voor Wiskunde en Informatica

Kruislaan 413
NL-1098 SJ
Amsterdam

Tel: +31 20 592 9333
Fax: +31 20 592 4199
http://www.cwi.nl/

Consiglio Nazionale delle Ricerche

IEI-CNR
Via S. Maria, 46
I-56126 Pisa

Tel: +39 050 593 433
Fax: +39 050 554 342
http://www.iei.pi.cnr.it/

Czech Research Consortium for Informatics and Mathematics

FIT-CTU
Thakurova 8a
CZ-602 00 Brno

Tel: +420 2 6884669
Fax: +420 2 6884903
http://www.uiam.cas.cz/CRCIM/bocomcol

Danish Consortium for Information Technology

DANIT c/o CIT
Anbogade 34
DK - 8200 Aarhus N

Tel: +45 8942 2440
Fax: +45 8942 2443
http://www.cit.dk/ERCIM/

Foundation for Research and Technology – Hellas

Institute of Computer Science
P.O. Box 1385
GR-711 10 Heraklion, Crete

Tel: +30 81 39 16 00
Fax: +30 81 39 16 01
http://www.ics.forth.gr/

GMD – Forschungszentrum Informationstechnik GmbH

Schloß Birlinghoven
D-53754 Sankt Augustin

Tel: +49 2241 14 0
Fax: +49 2241 14 2889
http://www.gmd.de/

Institut National de Recherche en Informatique et en Automatique

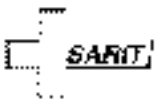
B.P. 105
F-78153 Le Chesnay

Tel: +33 1 39 63 55 11
Fax: +33 1 39 63 53 30
http://www.inria.fr/

Swedish Institute of Computer Science

Box 1263
S-164 28 Kista

Tel: +46 8 633 1500
Fax: +46 8 633 1510
http://www.sics.se/



Swiss Association for Research in Information Technology

Dept. Informatik
ETH-Zentrum
CH-8092 Zürich

Tel: +41 1 632 72 41
Fax: +41 1 632 11 72
http://www.vdf.ethz.ch/sarit/

Stiftelsen for Industriell og Teknisk Forskning ved Norges Tekniske Høgskole

SINTEF Telecom & Informatics
N-7034 Trondheim

Tel: +47 73 59 30 00
Fax: +47 73 59 43 02
http://www.informatics.sintef.no/

Slovak Research Consortium for Informatics and Mathematics

Dept. of Computer Science, Comenius University
Mlynska Dolina M
SK-842 15 Bratislava

Tel: +421 7 726635
Fax: +421 7 727041

Magyar Tudományos Akadémia – Számítástechnikai és Automatizálási Kutató Intézete

P.O. Box 63
H-1518 Budapest

Tel: +36 1 466 5644
Fax: +36 1 466 7503
http://www.sztaki.hu/

Technical Research Centre of Finland

VTT Information Technology
P.O. Box 1200
FIN-02044 VTT

Tel: +358 9 456 6041
Fax: +358 9 456 6027
http://www.vtt.fi/