Ph.D. Thesis, March 2007
Universidad Autónoma de Madrid

Realizing Interoperability of E-Learning Repositories

Daniel Olmedilla
olmedilla@L3S.de

March 2007

Supervisors:
Prof. Pablo Castells, Universidad Autónoma de Madrid, Spain
Prof. Wolfgang Nejdl, L3S & University of Hannover, Germany

Typeset by the author with the LaTeX 2$_\varepsilon$ Documentation System.

Contents

List of Figures

List of Tables

Chapter 1

Abstract (Español)

Tras la explosión del efecto Internet la Web ofrece una enorme cantidad de información. ¿ Significa esto que los usuarios pueden encontrar fácilmente y de manera efectiva la información que buscan? La respuesta es no. Por ejemplo, de acuerdo a recientes estudios los usuarios encuentran la información que buscan en tan sólo un 50% o menos de las veces (y dedican entre un 15% y un 35% de su tiempo a esta tarea). Esta situación se debe a la falta de interoperabilidad y a la sobrecarga de información. Por un lado, una gran parte de la información disponible no es fácilmente accesible sino que se encuentra protegida detrás del proveedor de información en el que se guarda. Usuarios deben acceder a cada proveedor de información de manera individual para encontrar los resultados buscados. Por otro lado, búsquedas basadas en palabras clave pueden producir un número de resultados relevantes difícil de manejar y por tanto demuestra la necesidad de lenguajes de búsqueda más precisos y su posterior ordenación.

Este documento contribuye a la mejora de la perspectiva global respecto a interoperabilidad en sistemas de gestión de aprendizaje y repositorios de objetos de aprendizaje online, así como cada uno de los pasos necesarios para conseguir dicho objetivo: lenguaje de búsqueda común, vocabulario global, integración semántica y ranking. Este trabajo mejora o soluciona los principales desafíos para la mejora de interoperabilidad y de esta manera mejorar soluciones existentes y aumentar su eficiencia y efectividad desde el punto de vista del consumidor y proveedor de información.

Chapter 2

Abstract (English)

After the boom of Internet a huge amount of information is available on the Web. Does that mean that users can easily and effectively find the specific information they seek? The answer is no. For example, searchers successfully find what they seek only 50% of the time or less (and they spend 15% to 35% of their time searching for information). This is due to the lack of interoperability and information overload. On the one hand, a big portion of the available information is not easily accessible for consumption but closed under each information source where it is stored. Users may need to access each information source individually in order to collect relevant information. On the other hand, keyword based queries may return an difficult to manage number of relevant results therefore showing the need for more accurate query languages and sorting mechanisms.

This document contributes to enhance the overall interoperability perspective in current e-learning management systems and on-line learning object repositories as well as each of the steps which need to be followed towards achieving such a goal, namely common query language, common schema, semantic integration and ranking. This work improves or overcomes the main challenges for interoperability in order to enhance existing approaches and increase their efficiency and effectiveness from both the provider's and consumer's perspective.

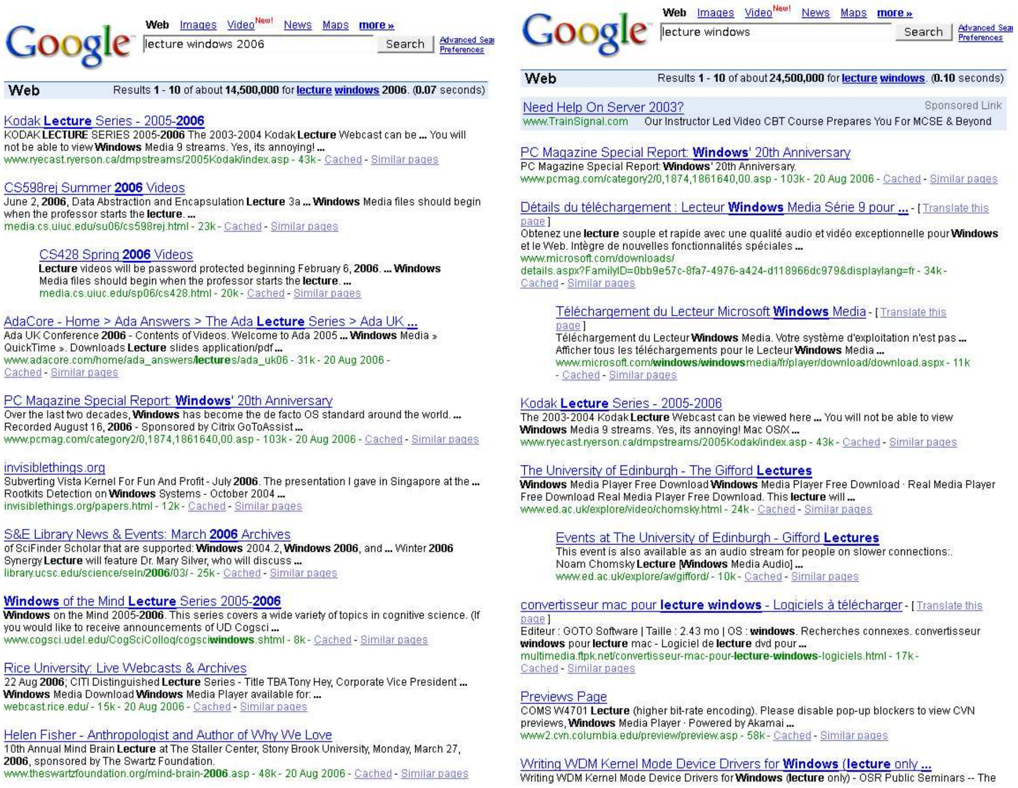
Chapter 3

Introduction

Nowadays, the digital world has become a reality and, as a consequence, the amount of information available is difficult to manage and is increasing rapidly. In such a world, centralized repositories[1] do not scale having as a consequence that information is distributed among many systems all over the world. As the Web has demonstrated with its success, distributed environments are highly powerful with respect to information sharing. However, making information available does not mean that it is easily accessible. Unfortunately, due to information overload on the Web, searching for information is more than a challenging task.

Imagine the following scenario: Alice is interested in learning about Windows and would like attend a lecture about it this year. She knows there are search engines where she can find a great deal of relevant material on the Web. She knows about Google and therefore she submits a query with the keywords “lecture windows 2006”. Unfortunately, such a query returns results which are completely irrelevant to her intended goal (see figure 3.1(a)). She tries to simplify her query by generalizing it to “lecture windows” but she does not have better luck (see figure 3.1(b)). She decides to replace “lecture” by “course” and tries with the query “windows course 2006”. This time she receives many results which refer to exactly the same resource (still off-topic): a book called “Windows on the World Complete Wine Course” (see figure 3.2(a)). She gives it a last try. This time she decides she will not look for lectures or courses but simply for resources with which she can learn Windows from home. She submits a query containing “learn windows home” and receives a set of different results (see figure 3.2(b)), mostly irrelevant. In all these cases, although there are no relevant results within the first 10 results, search engines return relevant material among the set of potential

[1] Centralized refers to the fact that only one entity has the control, even though replication among several computers could be used.



lecture windows 2006

Web Results 1 - 10 of about 14,500,000 for lecture windows 2006. (0.07 seconds)

Kodak Lecture Series - 2005-2006
KODAK LECTURE SERIES 2005-2006 The 2003-2004 Kodak Lecture Webcast can be ... You will not be able to view Windows Media 9 streams. Yes, its annoying! ...
www.ryecast.ryerson.ca/dmpstreams/2005Kodak/index.asp - 43k - Cached - Similar pages

CS598rej Summer 2006 Videos
June 2, 2006, Data Abstraction and Encapsulation Lecture 3a ... Windows Media files should begin when the professor starts the lecture. ...
media.cs.uiuc.edu/su06/cs598rej.html - 23k - Cached - Similar pages

CS428 Spring 2006 Videos
Lecture videos will be password protected beginning February 6, 2006. ... Windows Media files should begin when the professor starts the lecture. ...
media.cs.uiuc.edu/sp06/cs428.html - 20k - Cached - Similar pages

AdaCore - Home > Ada Answers > The Ada Lecture Series > Ada UK ...
Ada UK Conference 2006 - Contents of Videos. Welcome to Ada 2005 ... Windows Media » QuickTime ». Downloads Lecture slides application/pdf ...
www.adacore.com/home/ada_answers/lectures/ada_uk06 - 31k - 20 Aug 2006 - Cached - Similar pages

PC Magazine Special Report: Windows' 20th Anniversary
Over the last two decades, Windows has become the de facto OS standard around the world. ... Recorded August 16, 2006 - Sponsored by Citrix GoToAssist ...
www.pcmag.com/category2/0,1874,1861640,00.asp - 103k - 20 Aug 2006 - Cached - Similar pages

invisiblethings.org
Subverting Vista Kernel For Fun And Profit - July 2006. The presentation I gave in Singapore at the ... Rootkits Detection on Windows Systems - October 2004 ...
invisiblethings.org/papers.html - 12k - Cached - Similar pages

S&E Library News & Events: March 2006 Archives
of SciFinder Scholar that are supported: Windows 2004.2, Windows 2006, and ... Winter 2006 Synergy Lecture will feature Dr. Mary Silver, who will discuss ...
library.ucsc.edu/science/seln/2006/03/ - 25k - Cached - Similar pages

Windows of the Mind Lecture Series 2005-2006
Windows on the Mind 2005-2006. This series covers a wide variety of topics in cognitive science. (If you would like to receive announcements of UD Cogsci ...
www.cogsci.udel.edu/CogSciColloq/cogsciwindows.shtml - 8k - Cached - Similar pages

Rice University: Live Webcasts & Archives
22 Aug 2006; CITI Distinguished Lecture Series - Title TBA Tony Hey, Corporate Vice President ... Windows Media Download Windows Media Player available for: ...
webcast.rice.edu/ - 15k - 20 Aug 2006 - Cached - Similar pages

Helen Fisher - Anthropologist and Author of Why We Love
10th Annual Mind Brain Lecture at The Staller Center, Stony Brook University, Monday, March 27, 2006, sponsored by The Swartz Foundation.
www.theswartzfoundation.org/mind-brain-2006.asp - 48k - 20 Aug 2006 - Cached - Similar pages

(a) Keywords: "lecture windows 2006"

lecture windows

Web Results 1 - 10 of about 24,500,000 for lecture windows. (0.10 seconds)

Need Help On Server 2003? Sponsored Link
www.TrainSignal.com Our Instructor Led Video CBT Course Prepares You For MCSE & Beyond

PC Magazine Special Report: Windows' 20th Anniversary
PC Magazine Special Report: Windows' 20th Anniversary.
www.pcmag.com/category2/0,1874,1861640,00.asp - 103k - 20 Aug 2006 - Cached - Similar pages

Détails du téléchargement : Lecteur Windows Media Série 9 pour ... - [Translate this page]
Obtenez une lecture souple et rapide avec une qualité audio et vidéo exceptionnelle pour Windows et le Web. Intègre de nouvelles fonctionnalités spéciales ...
www.microsoft.com/downloads/details.aspx?FamilyID=0bb9e57c-8fa7-4976-a424-d118966dc979&displaylang=fr - 34k - Cached - Similar pages

Téléchargement du Lecteur Microsoft Windows Media - [Translate this page]
Téléchargement du Lecteur Windows Media. Votre système d'exploitation n'est pas ... Afficher tous les téléchargements pour le Lecteur Windows Media ...
www.microsoft.com/windows/windowsmedia/fr/player/download/download.aspx - 11k - Cached - Similar pages

Kodak Lecture Series - 2005-2006
The 2003-2004 Kodak Lecture Webcast can be viewed here ... You will not be able to view Windows Media 9 streams. Yes, its annoying! Mac OS/X ...
www.ryecast.ryerson.ca/dmpstreams/2005Kodak/index.asp - 43k - Cached - Similar pages

The University of Edinburgh - The Gifford Lectures
Windows Media Player Free Download Windows Media Player Free Download · Real Media Player Free Download Real Media Player Free Download. This lecture will ...
www.ed.ac.uk/explore/video/chomsky.html - 24k - Cached - Similar pages

Events at The University of Edinburgh - Gifford Lectures
This event is also available as an audio stream for people on slower connections:. Noam Chomsky Lecture [Windows Media Audio] ...
www.ed.ac.uk/explore/av/gifford/ - 10k - Cached - Similar pages

convertisseur mac pour lecture windows - Logiciels à télécharger - [Translate this page]
Editeur : GOTO Software | Taille : 2.43 mo | OS : windows. Recherches connexes. convertisseur windows pour lecture mac - Logiciel de lecture dvd pour ...
multimedia.ftpk.net/convertisseur-mac-pour-lecture-windows-logiciels.html - 17k - Cached - Similar pages

Previews Page
COMS W4701 Lecture (higher bit-rate encoding). Please disable pop-up blockers to view CVN previews, Windows Media Player · Powered by Akamai ...
www2.cvn.columbia.edu/preview/preview.asp - 58k - Cached - Similar pages

Writing WDM Kernel Mode Device Drivers for Windows (lecture only ...
Writing WDM Kernel Mode Device Drivers for Windows (lecture only) - OSR Public Seminars -- The

(b) Keywords: "lecture windows"

Figure 3.1: Alice searches for resources to learn Windows (1)

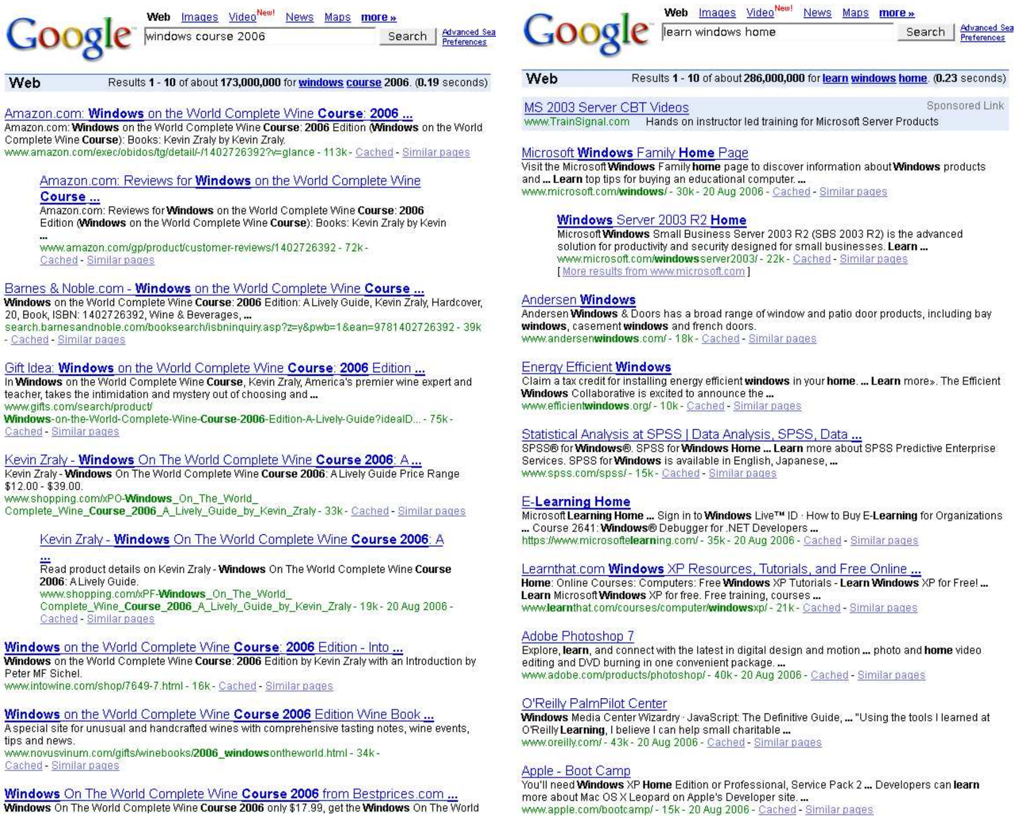
results (if it exists, it should match the query). However, the last query, for example, returned an estimation of 286 million matching documents. Such amount of information is unmanageable for users and finding the right information on the Web may take Alice too much time. Recent research [52, 51] states that:

- *knowledge workers spend from 15% to 35% of their time searching for information*

- *searchers are successful in finding what they seek 50% of the time or less*

what shows that current searching mechanisms are still limited and must be improved.

Current web search engines, such as Google [60], Altavista [5] or Yahoo! [158] are the typical place where users may search for information.



windows course 2006

Web Results 1 - 10 of about 173,000,000 for windows course 2006. (0.19 seconds)

Amazon.com: Windows on the World Complete Wine Course: 2006 ...
Amazon.com: Windows on the World Complete Wine Course: 2006 Edition (Windows on the World Complete Wine Course): Books: Kevin Zraly by Kevin Zraly.
www.amazon.com/exec/obidos/tg/detail/-/1402726392?v=glance - 113k - Cached - Similar pages

Amazon.com: Reviews for Windows on the World Complete Wine Course ...
Amazon.com: Reviews for Windows on the World Complete Wine Course: 2006 Edition (Windows on the World Complete Wine Course): Books: Kevin Zraly by Kevin ...
www.amazon.com/gp/product/customer-reviews/1402726392 - 72k - Cached - Similar pages

Barnes & Noble.com - Windows on the World Complete Wine Course ...
Windows on the World Complete Wine Course: 2006 Edition: A Lively Guide, Kevin Zraly, Hardcover, 20, Book, ISBN: 1402726392, Wine & Beverages, ...
search.barnesandnoble.com/booksearch/isbninquiry.asp?z=y&pwb=1&ean=9781402726392 - 39k - Cached - Similar pages

Gift Idea: Windows on the World Complete Wine Course: 2006 Edition ...
In Windows on the World Complete Wine Course, Kevin Zraly, America's premier wine expert and teacher, takes the intimidation and mystery out of choosing and ...
www.gifts.com/search/product/
Windows-on-the-World-Complete-Wine-Course-2006-Edition-A-Lively-Guide?idealD... - 75k - Cached - Similar pages

Kevin Zraly - Windows On The World Complete Wine Course 2006: A ...
Kevin Zraly - Windows On The World Complete Wine Course 2006: A Lively Guide Price Range $12.00 - $39.00.
www.shopping.com/xPO-Windows_On_The_World_
Complete_Wine_Course_2006_A_Lively_Guide_by_Kevin_Zraly - 33k - Cached - Similar pages

Kevin Zraly - Windows On The World Complete Wine Course 2006: A ...
Read product details on Kevin Zraly - Windows On The World Complete Wine Course 2006: A Lively Guide.
www.shopping.com/xPF-Windows_On_The_World_
Complete_Wine_Course_2006_A_Lively_Guide_by_Kevin_Zraly - 19k - 20 Aug 2006 - Cached - Similar pages

Windows on the World Complete Wine Course: 2006 Edition - Into ...
Windows on the World Complete Wine Course: 2006 Edition by Kevin Zraly with an Introduction by Peter MF Sichel.
www.intowine.com/shop/7649-7.html - 16k - Cached - Similar pages

Windows on the World Complete Wine Course 2006 Edition Wine Book ...
A special site for unusual and handcrafted wines with comprehensive tasting notes, wine events, tips and news.
www.novusvinum.com/gifts/winebooks/2006_windowsontheworld.html - 34k - Cached - Similar pages

Windows On The World Complete Wine Course 2006 from Bestprices.com ...
Windows On The World Complete Wine Course 2006 only $17.99, get the Windows On The World

(a) Keywords: "windows course 2006"

learn windows home

Web Results 1 - 10 of about 286,000,000 for learn windows home. (0.23 seconds)

MS 2003 Server CBT Videos Sponsored Link
www.TrainSignal.com Hands on instructor led training for Microsoft Server Products

Microsoft Windows Family Home Page
Visit the Microsoft Windows Family home page to discover information about Windows products and ... Learn top tips for buying an educational computer. ...
www.microsoft.com/windows/ - 30k - 20 Aug 2006 - Cached - Similar pages

Windows Server 2003 R2 Home
Microsoft Windows Small Business Server 2003 R2 (SBS 2003 R2) is the advanced solution for productivity and security designed for small businesses. Learn ...
www.microsoft.com/windowsserver2003/ - 22k - Cached - Similar pages
[More results from www.microsoft.com]

Andersen Windows
Andersen Windows & Doors has a broad range of window and patio door products, including bay windows, casement windows and french doors.
www.andersenwindows.com/ - 18k - Cached - Similar pages

Energy Efficient Windows
Claim a tax credit for installing energy efficient windows in your home. ... Learn more». The Efficient Windows Collaborative is excited to announce the ...
www.efficientwindows.org/ - 10k - Cached - Similar pages

Statistical Analysis at SPSS | Data Analysis, SPSS, Data ...
SPSS® for Windows®. SPSS for Windows Home ... Learn more about SPSS Predictive Enterprise Services. SPSS for Windows is available in English, Japanese, ...
www.spss.com/spss/ - 15k - Cached - Similar pages

E-Learning Home
Microsoft Learning Home ... Sign in to Windows Live™ ID · How to Buy E-Learning for Organizations ... Course 2641: Windows® Debugger for .NET Developers ...
https://www.microsoftelearning.com/ - 35k - 20 Aug 2006 - Cached - Similar pages

Learnthat.com Windows XP Resources, Tutorials, and Free Online ...
Home: Online Courses: Computers: Free Windows XP Tutorials - Learn Windows XP for Free! ... Learn Microsoft Windows XP for free. Free training, courses ...
www.learnthat.com/courses/computer/windowsxp/ - 21k - Cached - Similar pages

Adobe Photoshop 7
Explore, learn, and connect with the latest in digital design and motion ... photo and home video editing and DVD burning in one convenient package. ...
www.adobe.com/products/photoshop/ - 40k - 20 Aug 2006 - Cached - Similar pages

O'Reilly PalmPilot Center
Windows Media Center Wizardry · JavaScript: The Definitive Guide, ... "Using the tools I learned at O'Reilly Learning, I believe I can help small charitable ...
www.oreilly.com/ - 43k - 20 Aug 2006 - Cached - Similar pages

Apple - Boot Camp
You'll need Windows XP Home Edition or Professional, Service Pack 2 ... Developers can learn more about Mac OS X Leopard on Apple's Developer site. ...
www.apple.com/bootcamp/ - 15k - 20 Aug 2006 - Cached - Similar pages

(b) Keywords: "learn windows home"

Figure 3.2: Alice searches for resources to learn Windows (2)

They provide a solution to the distribution of content by gathering, storing and indexing as much information as they can. Some people may argue that thanks to search engines interoperability is not an issue. However, they unfortunately are not able to cover the whole Web. Coverage is even less for resources available on-line through different mechanisms other than the Web such as those available through Peer-to-Peer (P2P hereafter) networks or Web Services. In addition, although they cover a big percentage, still it is sometimes very difficult for users to find the right information they seek. Some of the reasons are:

- *Unstructured information and lack of semantics.* There is no structure on the Web which would allow search engines to organize and classify its content to make search more effective. For example, it is not possible for a search engine to distinguish wether a paragraph in a page contains a biography or a book summary; nor can search engines determine the topic of the paragraph. Keyword-based queries only search for occurrences of terms (e.g., “Windows” or “course”) in any part of the resource. In addition, matching is only syntactic and therefore terms may be ambiguous. For example, “Windows” may refer to an operating system, to a window in graphical system, to the name of an application (e.g., “Windows media player”) or to the glass element of a house or car. A system may not be able to disambiguate queries containing such terms (see figures 3.1 and 3.2).

- *Size and coverage of the Web.* Much information is not gathered and therefore not included in the indices of search engines. The reason is that the Web is much too large and search engines resources are limited. Crawlers not only must visit as many sites as they can, but also revisit them periodically in order to keep indices fresh. In addition, indices from search engines already require thousands of distributed computers in order to store such magnitudes of information.

- *Hidden Web (also Deep Web or Invisible Web).* Many pages are not linked by other pages (e.g., dynamically generated web pages) or protected (e.g., by registration mechanisms) and therefore search engines ignore their existence. BrightPlanet [11] estimates, using Google as a benchmark, that there are about 500 billion pages of information available on the Web, and only 1/500 of that information can be reached via traditional search engines [140]. Furthermore, there is a great deal of information available from different means besides the Web. For example, the boom of Web Services and P2P networks offer a huge amount of resources which are not considered in current search engines. Much of that information is public or at least could be searched (although consumption would not be for free) if there were standard means of interconnecting those different sources.

- *Ranking.* A query filters the information available according to some relevance criteria (e.g., keyword matching). However, due to the huge amount of existing information, it is usual the case that a query submitted to a search engine returns thousands (or even millions) of “relevant” results. A large number of results forces the user to manually explore them in order to actually find the relevant resources (or even the “best match”). This is a time consuming task. However, if the results are ordered, the time spent by users is dramatically reduced. The current problem is that ranking algorithms are global (same ranking for every user) and typically exploit the structure of the corpus (e.g., links among pages). Personalized ranking would save even more time for users (e.g., results to the query “Windows” should be different for a programmer than for a house constructor).

Many applications, such as content management systems, require users to register and sign in before offering full functionality. Therefore, its information is not available to search engines. In addition, many proprietary databases do not even publish their content on the Web. Still, it is common that institutions work together and therefore they need to interconnect their applications, allowing an application A to retrieve and use information available in application B. Since there is no common standard for the exchange of information among applications, these *coalitions* end up developing ad-hoc solutions for their needs, therefore increasing the cost of integrating new applications. Those applications typically connect by means of two mechanisms depending on the needs:

Replication ensures that each system contains a copy of other’s information. However, many institutions are not willing to replicate all their data because they could not charge users according to access to that data or they would loose control over it.

Federated Search provides a system with a list of sources that will be dynamically queried when requested. This requires an agreement between the system sending the query and the sources providing the information. Such an agreement is required at the institutional and technical level.

Regarding flexibility and autonomy, federated search is typically the choice. It provides a mechanism for searching a list of pre-configured sources therefore allowing those sources to keep control over their data. One of the main disadvantages is that due to the lack of a standard search interface, integration of new sources is typically done in a costly and non-reusable manner. Different coalitions define ad-hoc solutions, and as a consequence, costs are increased for new systems joining coalitions (for each coalition a new solution has to be developed); this reduces the number of systems interconnected and therefore a search engines coverage).

As an alternative, fully distributed environments like P2P networks have emerged. They have as main objective scalability and sharing of information. Their main advantages are: no single point of failure (some systems shutting down do not affect the whole network); owners of the information do not give its control away to any third party (they can keep it locally); dynamicity (peers often join and leave the network); and scalability. The main disadvantages are: the decrease of performance (it is more difficult to optimize services like search due for example to heterogeneity of the network); and usually a lack of interoperability with other systems/environments which do not follow their specific interfaces.

Providing transparent access to all available repositories would be easy if all players would use the same metadata profile, query language, storage layer and communication protocol. However, this is not the current situation and unlikely to happen in the very near future due to the lack of a standard and the proprietary solutions adopted by many of them. As extracted from a survey made among 38 industry associations in 27 different countries [77], *the most significant technology issues included integration (21%) and standards (20%).* Lack of standards within an specific sector typically means that interoperability among systems is achieved by investments among parties, and in pairwise manner if unlucky. This lack of reusability of interoperability solutions produces extra costs and often inhibits investment in interoperability [142].

In particular, E-Learning repositories are not a different case. There currently exist a significant amount of learning material available on the Web, on learning management systems and on users computers (e.g., professors and students). However, such material is not shared appropriately (or not at all) and therefore it is not made available to users and learners. As described above, lack of standards and appropriate integration solutions prevent users from easily and effectively finding relevant resources to their needs.

Most of the work described in this document has been performed in the context of the EU/IST project Elena [48], the EU/IST Network of Excellence PROLEARN [118] and the EU/IST Integrated Project TENCompetence [144] and therefore many scenarios and descriptions as well as the title of this document are biased towards e-learning. However, most of this work can be generalized to any domain and, in fact, most of the contributions of this thesis are currently being used also in other contexts like P2P search and desktop sharing and ranking. Some chapters of this document are left context-free intentionally, thereby demonstrating its independence from any specific domain.

This thesis describes current existing challenges for interoperability of distributed environments in order to share information. The main goal of the thesis is to overcome the requirements specified above by improving existing solutions or creating new ones in order to enhance existing approaches to interoperability and increase its efficiency from users perspective. This way, the effort needed to "realize interoperability" among systems would be smaller and users would still benefit from greater advantages. The contributions of this thesis include

- identification of requirements for system interoperability
- specification and standardization of a simple query interface to be adopted by systems willing to be interoperable
- development of open source components (based on SQI) to reduce the effort and costs of information providers
- specification of a proxying architecture in order to open (typically) closed environments to other consumers and providers
- creation of ontologies to annotate learning material and represent complex competences
- presentation of mappings and its application via query rewriting mechanisms in order to provide components for effective and low-cost semantic integration
- description of new ranking algorithms, one providing personalized results to the user assuming the existence of relationships among resources and another to adequate for unlinked corpus
- integration of all the previous items in a system as a proof of concept interoperability demonstration
- demonstration of the interoperability achievement through several networks of learning resources providers and projects world wide

The rest of the thesis is organized as follows: chapter 4 motivates the need for interoperability, its definition and the scope in which it is used in this paper. It also identifies the requirements and steps towards its achievement as well as current challenges and related work. Later chapters extend each one of the challenges explained in this chapter. The specification of a new standard, called Simple Query Interface is presented in chapter 5. In addition, a proxying architecture in which this interface is used is presented. In addition to a common communication interface, a common vocabulary is needed. Chapter 6 introduces several schemas for the description of learning resources and competences. These descriptions provide different levels of detail allowing for more basic or advanced services using them. In order to translate among concepts in different schemas, a component for semantic integration is presented in chapter 7. This component allows property and value mappings among schemas as well as the definition of default values.

Once information is retrieved it needs to be ordered so that the most important results appear earlier, therefore reducing the time a user needs to find relevant information. Chapter 8 provides two new algorithms for ranking results returned by a user query. All previous sections describe different steps towards the goal of interoperability. Chapter 9 combines all these into several systems and scenarios in order to demonstrate the feasibility of the contributions of this document as well as an example of its application and success. Finally, the conclusions and open issues which require further research are presented in section 10.

Chapter 4

Interoperability: What is it and Why is it needed?

This chapter introduces the term "interoperability" and defines the scope in which it is used in the rest of the document. In addition, it motivates the need for interoperability among information providers, presents the main challenges to achieve it and describes current state of the art in the area.

4.1 What is it?

The way the term "interoperability" is used differs among different communities. However, all of them have in common that they use it to mean the ability to talk to each other or work together, therefore encompassing a meaning related to communication and/or sharing. The following are some of the available definitions of "interoperability":

1. able to operate in conjunction [30, 98].

2. Interoperability is the ability of products, systems, or business processes to work together to accomplish a common task. The term can be defined in a technical way or in a broad way, taking into account social, political and organizational factors [156].

3. Interoperability is the ability of multiple systems with different hardware and software platforms, data structures, and interfaces to exchange data with minimal loss of content and functionality [106].

4. The ability of two or more systems, or components to exchange information, and to use the information that has been exchanged [72, 4].

5. The ability of various types of computers and programs to work together [38].
6. The ability of information systems to operate in conjunction with each other encompassing communication protocols, hardware software, application, and data compatibility layers [71].
7. The ability of hardware and software from different vendors to understand each other and exchange data, either within the same network or across dissimilar networks [151].
8. ability of a system or a product to work with other systems or products without special effort on the part of the customer [155].
9. to be interoperable, one should actively be engaged in the ongoing process of ensuring that the systems, procedures and culture of an organisation are managed in such a way as to maximise opportunities for exchange and re-use of information, whether internally or externally [98].
10. achieved only if the interaction between two systems can, at least, take place at the three levels: data, resource and business process with the semantics defined in a business context [21].

Let us extract some of the most important elements from the above definitions. Some definitions refer to the ability of working together to accomplish a common task (def. 2, 5 and 8), work in conjunction (def. 1 and 6), and exchange of information and, really important, use it (def. 4 and 7). Furthermore, it is also suggested that interoperability must be provided at different levels (def. 6, 9 and 10). Finally, such interoperability should be done without increasing the effort of the user (def. 8).

Although some readers may be interpreting the above definitions exclusively from a technical point of view, it is important to note that the term interoperability encompasses several other notions [98]:

Technical interoperability. An agreement on the communication, transport, storage and representation is needed.

Semantic interoperability. Common ontologies or thesauri are required avoiding the use of different terms to mean the same things or same terms to mean different ones.

Political interoperability. Sharing resources may involve a change in the business process of the institution.

Inter-community interoperability. Partnerships and agreements among institutions are often required before information may be shared among them.

Legal interoperability. Sharing of information must follow legislation (e.g., resources to which data protection laws apply).

International interoperability. Languages and cultural issues may present new problems to be solved.

The rest of this document focuses, among these different notions, only on the first two items, that is, technical and semantic interoperability, and assumes that interoperability at the other different levels have been established or do not apply.

4.2 Why is it needed?

From an e-commerce perspective, interoperability has two main advantages: the first is *to present "one face to the customer"*, and the second is *to provide "global inventory visibility"* [36]. From a more general point of view, an information provider seeks to make its content available to the widest audience at the lowest cost possible. Although interoperability is a significant strategic direction, it is often however inhibited by cost [142]. Expenditure by organizations around the globe on external ICT[1] products and services amounts to $1.45 trillion ($1,450 billion) annually [77]. In particular, investment in technology in Europe accounted €6.4 billion, that is, 71% of the amount invested in e-commerce in 2004 (10% more than previous year). Computer software is the largest sector according to total investment with €1.7 billion or 27% of total technology investment (see figure 4.1). A survey in 1998 found that 76% of company executives considered information to be "mission critical" and their company's most important asset [52]. In a survey made among 38 industry associations in 27 different countries, *the most significant technology issues in e-commerce included integration (21%), standards (20%) and open systems (3%)*, as depicted in figure 4.2. This survey highlights the importance of integration and standards, and *identifies them as key technological issues*. Lack of standards within an specific sector typically means that interoperability among systems is achieved by investments among parties, and in pairwise manner if unlucky. This lack of reusability of interoperability solutions produces extra costs and often inhibits investment in interoperability [142]. On the other hand, if standards exist or interoperability solutions

[1] Information Communications Technology

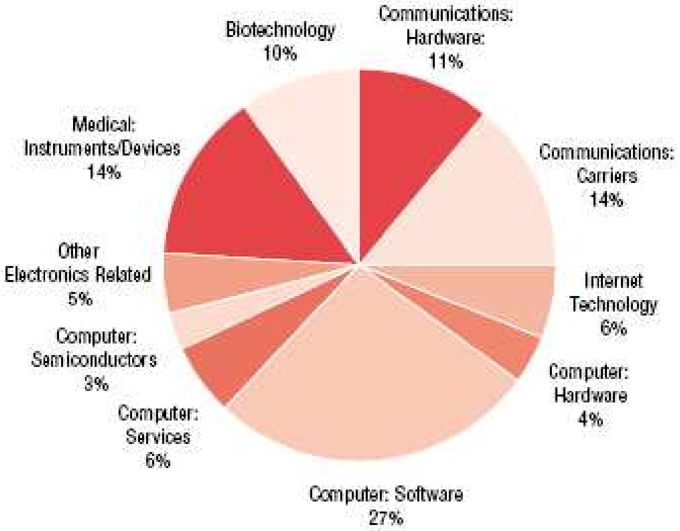


Figure 4.1: 2004 Technology Private Equity Investments Amount By Sector [99]

may be reused, the extra costs derived from providing interoperability are drastically reduced. Another solution could be to rely on an outsource service provider, but that would mean loosing control over the information and knowledge which may be much too valuable to share with third parties.

From a user point of view, lack of interoperability means incompleteness of information (restricted to that stored in the local repository) and therefore low recall. This typically leads to an inefficient process in which users spend too much time searching (possibly individually in different providers) and processing results in order to achieve their goals. Furthermore, it is typically the case that users either desist on their task without finding any solution or do not find the optimal solution (only partially achieving their goals), therefore reducing their performance. Institutions like IDC, Working Council of CIOs, AIIM, the Ford Motor Company and Reuters have found [52, 51] that

- *knowledge workers spend from 15% to 35% of their time searching for information*
- *searchers are successful in finding what they seek 50% of the time or less*

It is estimated that the total lost of not finding the right information is among \$2.5 to \$3.5 million per year for an enterprise with 1000 knowledge

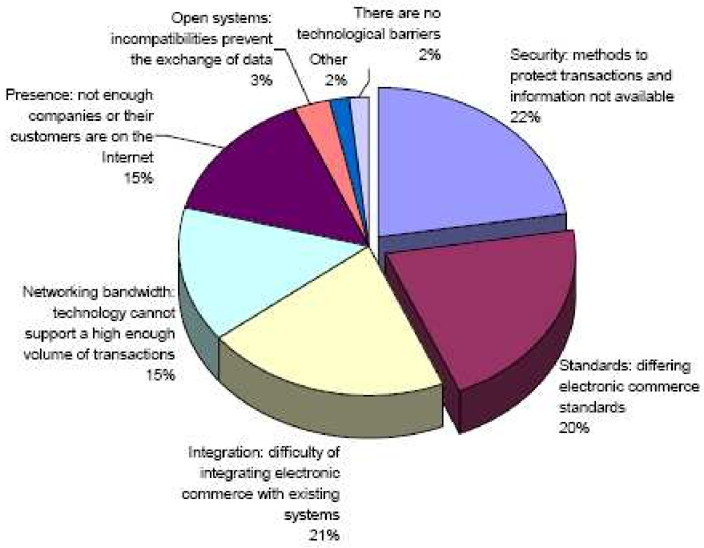


Figure 4.2: What are the most significant technology issues? [77]

workers [52]. The opportunity cost is even greater, with potential additional revenue of $15 million annually.

In summary, the goal of information providers is *to provide the right information, at the right time, to the right people, in the right context, in the right format* [136] at the lowest cost. However, *the sheer scale and volume of data, the variety of data sources and formats, the number of data owners, and the geographic distribution of the suppliers and consumers of data impose real challenges* [136] that are not easy to overcome. The lack of interoperability may lead [52] to:

- Poor decisions based on poor or faulty information
- Duplicated efforts
- Lost sales because customers give up before finding what they are searching for
- Lost of productivity when users (or employees) do not find the information they need and ask other colleagues[2].

In particular, e-learning is of course not different. In fact, e-learning is still behind other sectors regarding the adoption of existing standards

[2] Nonproductive information-related activities is estimated between 15% to 25% of employee's time [52].

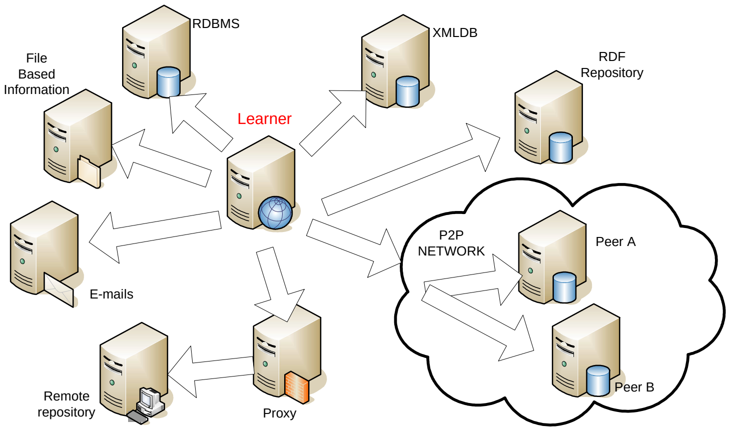


Figure 4.3: Different information sources with different communication protocols, schemas and query languages.

and therefore interoperability among learning repositories is still in its infancy (as demonstrated by the interest of the research community in the area [111, 112]). Providing interoperability would dramatically improve users satisfaction and performance and expand providers electronic distribution channels.

4.3 Challenges and Related Work

It is important to note that in this document only the sharing of metadata about learning resources, also known as learning objects, is considered. While this metadata is typically available and sharable, the learning object itself might not be. Therefore, this document does not deal with negotiations for the actual use of learning resources by users but only with the sharing, exchange and use of resource's metadata.

Providing transparent access to all available repositories would be easy if all players would use the same metadata profile, query language, storage layer and communication protocol. However, this is not the current situation (figure 4.3 gives a hint of the heterogeneity of systems on these terms) and unlikely to happen in the very near future due to the lack of a standard and the proprietary solutions adopted by many of them. In the following, the requirements learning repositories must satisfy in order to achieve technical and semantic interoperability are explained and an overview of the state of the art in the area is provided.

4.3.1 Common Communication Protocol and Interface

Different repositories provide different access methods and interfaces. Some examples are Web Services, different Remote Procedure Call methods, HTTP forms or even other proprietary solutions. In order to be able to communicate to each other, they need to agree on a common protocol and a common interface.

This document does not deal with the approaches for such communication protocols and assume that an agreement on the binding among parties already exists so they can communicate with each other. Currently, it seems that Web Services are gaining momentum as a standard for modular and interoperable communication protocol and therefore this solution was chosen. However, the second requirement is the interface. There are some available solutions and some interesting approaches currently available. However, all of them lack a wide adoption due to different reasons such as complexity of implementation (therefore increasing the cost for adoption) or restricted functionality (e.g., only keyword-based search or synchronous querying). The following describes the state of the art on query interfaces and highlights the limitations of existing approaches.

The ANSI/NISO Z39.50-2003 [159] is an application protocol for search and retrieval of information in databases. The specification presents eleven facilities for search and information retrieval related functions, namely, initialization, search, retrieval, result-set-delete, browse, sort, access control, accounting/resource control, explain, extended services, termination. It provides different query types (not query languages) based on attribute-value constraints and returns full records as results, therefore not allowing a selection of the attributes in records retrieved. Furthermore, Z39.50 requires the server to keep a state of the current communication with the client (within the so called Z-association) including an extra database for the explain facility and it does not allow for asynchronous queries, required to query, for example, distributed environments like P2P networks.

SRW [138] (the Search/Retrieve Webservice) is an XML oriented protocol for search and information retrieval on the Internet. SRU [137] (Search/Retrieve URL) is its homologous but uses parameters in a URL as transport (instead of SOAP, as in SRW). Both are built based on the experiences of Z39.50 and combine several of its features such as search, present, sort and scan services. SRW and SRU define a search, scan and explain operations. However, since SRW and SRU were designed to be used for Web Service and internet queries, they do not allow for asynchronous queries. In addition, a single query language is assumed, CQL, in order to look for a trade-off between expressiveness and user-friendly and simplicity. However, assuming one single query language restricts the expressivity of all queries by not allowing other simpler or more advanced query languages.

Open Knowledge Initiative (OKI) [107] defines a set of Application Programming Interfaces to be used as application interfaces in order to separate what a service does from how it is implemented. In particular, OKI Repository OSID is a general abstraction for the storage and retrieval of digital content searching technologies without specifying any in particular. It does not specify any query language or semantics and depends on Search Types which are created upon agreements, which specify search criteria semantics and search properties. OKI Repository OSID can be seen as a high level abstraction which may be implemented by using other query technologies such as the ones described above. In fact, there exists a component OKI2SQI which allows using OKI at the application level and SQI[3] at the communication level.

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) is a protocol that allows systems to collect metadata from other repositories. OAI-PMH provides two types of selective harvesting: datestamps (only records created, deleted or modified after a date are retrieved) or set-membership (only items belonging to a set are retrieved). However, OAI-PMH does not allow search or query but only collection. That would imply that a system should retrieve all possible resources and then execute a query locally, what is not acceptable given the scenarios presented in this thesis.

IMS Digital Repositories Interoperability (DRI) [73] suggests recommendations in order to establish interoperability among the different functions digital repositories provide, namely, Search/Expose, Gather/Expose, Alert/Expose, Submit/Store and Request/Deliver. Regarding Search, it is for example suggested the use of XQuery [157] as query language and Z39.50 [159] for searching library information. However, this specification only provides recommendations and only at a high level what provokes that different implementations of DRI may not be interoperable [75].

Google offers several services to be used by other systems, including Search, Cache requests and Spelling requests [61]. Within the Search service, Google offers the possibility to query the full content of its indices by using the Google query language[58, 59]. The API provided for search consist of a query request and a query response. The main drawbacks are that only Google query format and query result may be used, the paradigm followed included all parameters in the query request (therefore being restricted to those and are inflexible to the addition or removal of parameters) and it in-

[3] The Simple Query Interface will be presented in chapter 5.1

cluded some Google-specific parameters (e.g., safeSearch or filter). However, the simplicity of the query interface has inspired the work on a simple query interface described later in this document (in chapter 5.1).

4.3.2 Common Query Language

Independently of the communication interface, a language is needed in order to define which information is relevant for a given request. Metadata may be stored in different kinds of repositories, such as relational databases, RDF repositories, file systems, XML stores, etc. On top of this lower level data management stores, repositories expose their content through different search and query languages.

A query language may vary from simple keywords to a more structured and expressive approach (such as SQL [134], XQuery [157], QEL [104] or CQL [31, 32]). Therefore, a query language must be chosen. This language must be powerful enough to include most of the expressiveness of any other repository query language, in order to allow mappings without loosing information. Furthermore, it is required that such a query language provides enough power to develop advanced information retrieval on top of it. This thesis does not deal with the development of any particular query language. On the contrary, a language called QEL [104] was selected as a common query language due to its high flexibility and expressiveness as well as the availability of several wrappers implemented[4] in order to provide access to the most common repositories (relational databases, RDF repositories, RDF files, etc...). For all of them, the wrapper receives a query in QEL form, transforms it into the local query language and return the results back in RDF format. There already exists wrappers that translate QEL into other languages like SQL [134] (for relational databases), XQuery [157] (for XML stores), RQL [81] and RDQL [124] (for RDF repositories), and also for other more specific services like querying Amazon web services with keywords or a limited subset of the Google query language [58, 59].

In the following, QEL is briefly described and some examples are given in order to better understand the examples that will be presented in the rest of the document.

QEL

QEL [104] (Query exchange Language) is an RDF query language which is used in the EDUTELLA P2P network [47, 101]. This language was designed

[4] Some of them (e.g., a Sesame [129] wrapper) created as part of this work

with the idea that it should be relatively easy to translate it to other query languages, thus allowing an easier development of rappers and, as a consequence, a higher number of available repositories. This query language is based on Datalog [147] which allows expressing database queries based on predicate logic (allowing rules and recursion).

QEL is adapted as a dialect of datalog making use of the flexible RDF data model. RDF data consists of triples of the form

$$(Subject, Predicate, Object)$$

and therefore it is easier to query information sources without the need for knowing its internal representation (e.g., tables or views). QEL also provides some constructions in order to query RDF data. Among them (see [104] for a full description) it is important to mention the use of URIs and the predicate $qel : s$ for matching triples. For example, given a query

$$qel : s(Subject, Predicate, Object)$$

is true if

- *Subject and Predicate are anonymous or non-anonymous RDF resources,*
- *Object is a non-anonymous or anonymous RDF resource or an RDF Literal* and
- *the triple "$Subject, Predicate, Object$" exists in the RDF data.*

For example, the following query would return all resources available in the RDF data store together with their title and language.

$$\begin{array}{l}
@prefix\ qel : < http : //www.edutella.org/qel\# > .\\
@prefix\ dc : < http : //purl.org/dc/elements/1.1/ > .\\
?-\ \ qel : s(Resource, dc : title, Title),\\
\qquad qel : s(Resource, dc : language, Language).
\end{array}$$

Other QEL constructions include $qel : member$ (for RDF container membership), $qel : equals$ (checks if two RDF nodes are the same), $qel : like$ (check whether an RDF literal or URI contains a substring), $qel : greaterThan$ and $qel : lessThan$ (which checks if one string value is greater or less than another one). In addition, in QEL rules can be used therefore allowing not only for conjunctive queries but also disjunctive queries such as

$@prefix\ qel :< http : //www.edutella.org/qel\# > .$
$@prefix\ dc :< http : //purl.org/dc/elements/1.1/ > .$
$checkKeyword(X, Z, Y) \leftarrow qel : s(X, dc : title, Z), qel : like(Z, Y).$
$checkKeyword(X, Z, Y) \leftarrow qel : s(X, dc : description, Z), qel : like(Z, Y).$
$?(X, Y) - checkKeyword(X, Y,' Intelligence').$

which searches for the word “Intelligence” appearing either in the title or in the description of resources and returns both the identifier of the resource and the matching text (either title or description)..

More examples of queries specified in QEL are available in appendix A and a full reference of the QEL language as well as its datalog and RDF syntax is available in [104].

4.3.3 Common Metadata Schema

When searching and retrieving information, it is important that both parties are able to not only communicate, but also understand each other. They will not successfully exchange information if they use different terms to mean the same things or the same terms to mean different things. In particular, although IEEE LOM [93] is becoming a standard for e-learning resources metadata, many repositories are based on specific profiles that may include extensions and specific value spaces. A similar situation arises with competence representation, where IEEE RCD [123] and HR-XML [70] have recently appeared but still available systems and institutions have not adopted them yet.

In both cases, although there have been some efforts on standardizing some basic schemas, still it is recognized that they are much too general and they should improved. The challenge remains to provide a metadata profile (or profiles) that better suits current e-learning needs and which provides a good compromise between expressivity and ease of adoption.

Learning resources

When talking about metadata one of the most used schemas is Dublin Core [33]. Dublin Core is defined for cross-domain resource descriptions. The Dublin Core metadata element set includes 15 attributes including, for example, title, description, creator, language or date. Due to the cross-domain approach of Dublin Core, its schema is much too general to be uniquely applied to any learning context.

Learning Object Metadata (LOM) [93] is a standard by the Learning Technology Standards Committee (LTSC) of the IEEE since June 2002. It provides an extension to Dublin Core for the annotation of learning objects. LOM provides a schema subdivided in 9 categories: general, life cycle, meta-metadata, technical, educational, rights, relation, classification and annotation. Each category contains different subelements and all were made optional. Due to the high amount of attributes and to the fact that some needs are not included or specified enough, it is typically the case that learning management system define their own profile of LOM with a subset of LOM attributes and other newly created ones.

IMS Learning Design (LD) [74] is a specification which provides a framework for the description and execution of teaching related activities. It provides a teaching metamodel based on a theater play metaphor and provides three different levels: A, B and C. Level A includes the core vocabulary to support pedagogic diversity and provides the main entities of the teaching process like activity, outcome, environment, role and person. Level B adds properties and conditions in order to allow personalization and level C adds notification for communication among entities.

The ADL Sharable Content Object Reference Model (SCORM) [127] defines a web based learning content aggregation model and runtime environment. It provides three main components: content model, meta-data and content-packaging. SCORM adopted the LOM standard for the meta-data component, therefore bringing both its benefits and drawbacks.

Competenc(i)es

There exist some standardization efforts on modelling competenc*ies*. These efforts focus on different aspects related to competency: competencies descriptions, competency profiles and relationships among competencies.

The IMS Reusable Definition of Competencies or Educational Objective (RDCEO) [76] and the later IEEE Reusable Competency Definition (RCD) [123] (based on IMS RDCEO) focus on reusable competency definitions. The primary idea is to build central repositories which define competencies for certain communities. These definitions can be referenced by external data structures, encouraging interoperability and reusability. However, IEEE RCD lacks information on context and proficiency level and does not allow relationships or recursive dependencies among competencies.

HR-XML focuses on the modeling of a wide range of information related to human resource tasks (like contact data or aspects of the curriculum vitae). The work performed in HR-XML Measurable Competencies [70] tries to define profiles in order to use such competency definitions. It specifies data sets like job requirement profiles (which describe the competencies that a person is required to have) or personal competency profiles (which describe the competencies a person has). Such profiles are composed of evidences (either required or acquired) referring to competency definitions (e.g., IEEE RCD). Unfortunately, the proposed model does not clearly separate required and acquired profiles. The consequence is that an acquired competency could have mandatory and optional elements according to the model. Furthermore, it is unclear why a competency is composed of several evidences: since a competency is a reusable object, evidences should rather represent a requirement or demonstrate the acquisition of a competency. Hence, the evidences should refer to or contain competency definitions and not vice versa.

The Simple Reusable Competency Map (SRCM) [135] tries to model relationships between competencies. A map can contain information about dependencies/equivalences among competencies, including the composition of complex competencies from simpler ones. In SRCM, relationships are modeled using a directed acyclic graph. However, the semantics of the model proposed in SRCM is confusing. Relationships among different nodes may have different meanings: composition, equivalence or order dependency. This leads to confusion when modeling tasks as well as when creating algorithms to use such information. Furthermore, combination and weighting of competencies is not clearly defined, and external references to the maps (e.g., from profiles) must point to the root (and not to any node), therefore requiring the traversal of the graph until the appropriate node is found. Moreover, it is not possible to model relationships among competencies, because proficiency level and context should be considered. For example, statistics knowledge may be a requisite for becoming a computer scientist or a sociologist. However, the proficiency level required and the context in which the competency is applied are completely different, hence making impossible to create relationships directly among competencies.

In OntoProPer [141], profiles are described by flat vectors containing weighted skills (where weights grow from 0 to 3), which are expressed as labels. Weights represent importance if applied to requirements or skill level if applied to acquired skills. The system itself mainly focuses on profile matching and introduces an automated way of building and maintaining profiles based on ontologies. [29] describes an ontology-based semantic matchmaking (using Description Logics) between skills demand and supply. In [89], which also defines a competence ontology for domain knowledge dissemination and retrieval, a competence is related to capabilities, skills and expertise (measured by levels growing from 1 to 5). In this approach still the context is not tackled, the relationships are defined at the skill level and the proficiency levels are not flexible enough.

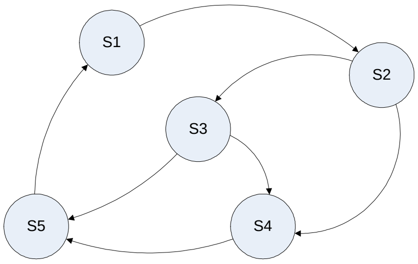


Figure 4.4: Graph Representing 1-to-1 Mappings

4.3.4 Semantic Integration

Although the previous section aims at the provision of a common metadata schema among all parties, I acknowledge the fact that this is probably never going to happen. It is probably not possible to have a single schema that suits all world needs. However, it is reasonable to assume that there may be a (hopefully small) number of standard schemas (or ontologies) and therefore a mapping among them needs to be provided [100]. This need even increases when content does not focus only on one domain but covers several of them. There are then two possibilities here: either each system maps its schema to a second system schema (in which case it reaches semantic interoperability by means of pair of mappings [1]) or a common global schema is provided and both systems provide a mapping to that common schema.

- If no virtual and unified schema is assumed in the network, systems within the network must provide pairs of mappings between each two systems. Subsequently, the distributed network can be seen as a directed graph (as shown in figure 4.4) in which each arrow represents an available mapping from one node to another. After that, they can be applied transitively in order to infer new mappings which were not explicitly defined. This is specially useful in P2P networks as it is usually not possible to enforce a unique and common schema. Authors in [1, 64] study this approach and provide algorithms to estimate the correctness of the inferred mappings.
- If a virtual and unified schema is assumed, there are two approaches for providing integration between the global schema and local schemas at the sources:
  - **Global As View (GAV)** [65]. In this approach, the global schema is expressed in terms of the data sources (an example is depicted in figure 4.5).

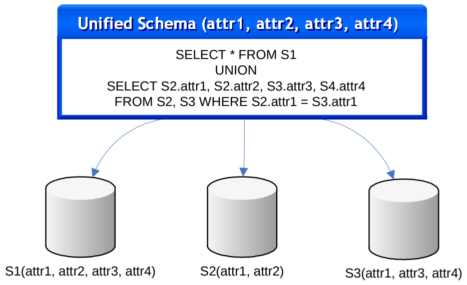

Unified Schema (attr1, attr2, attr3, attr4)
SELECT * FROM S1
UNION
SELECT S2.attr1, S2.attr2, S3.attr3, S4.attr4
FROM S2, S3 WHERE S2.attr1 = S3.attr1
S1(attr1, attr2, attr3, attr4)
S2(attr1, attr2)
S3(attr1, attr3, attr4)

Figure 4.5: Global As View Approach

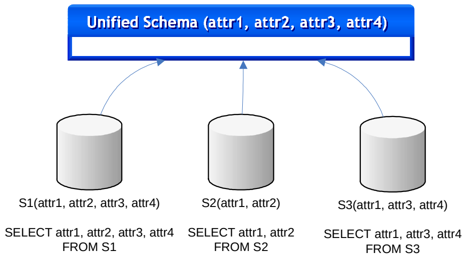

Unified Schema (attr1, attr2, attr3, attr4)
S1(attr1, attr2, attr3, attr4)
S2(attr1, attr2)
S3(attr1, attr3, attr4)
SELECT attr1, attr2, attr3, attr4
FROM S1
SELECT attr1, attr2
FROM S2
SELECT attr1, attr3, attr4
FROM S3

Figure 4.6: Local As View Approach

- **Local As View (LAV)** [148]. In this approach, each source is defined as a view over the global schema. This way, the global schema is specified independently from the sources (an example is depicted in figure 4.6).

A discussion of both GAV and LAV is provided in [91] as well as an introduction to "query rewriting" mechanisms. Query rewriting is the process in which a query expressed in the global schema is reformulated into another query according to a set of mappings [143].

There exists a large number of papers on ontology mapping, specially on the creation of such mappings in a (semi)automatic way. [122] provides a nice overview of the most relevant ones.

4.3.5 Ranking

Once search is realized, still too much information might reach the end user, that is, too many relevant results may be returned. Therefore, good ranking techniques are required in order to provide the user with those resources which are likely to be more relevant first. Ranking algorithms may be divided in two groups depending on the scenario in which interoperability is focused and the characteristics of the corpus distribution over information sources. The criteria to make this division depends on the answer to the question "does there exist relationships among resources in different sources or a big overlap of the resources contained among sources?

- If the answer is "Yes", in this case, like in the web, it is possible to analyze those relationships in order to rank the resources. Many algorithms exist on link analysis and rank aggregation but there is a lack of possibilities on personalized ranking.

- If the answer is "No", then existing techniques on link analysis or rank aggregation cannot be used and typically ad-hoc solutions (if any) are provided.

Ranking based on Link Analysis

PageRank [114, 13] is a method for computing a rank for every web page based on the graph of the web. The idea behind PageRank is that pages with many backlinks are more important that pages with only a few backlinks. Let us think of each link from page q to page p as a vote or recommendation. It means that highly linked pages are more "important" (voted or recommended). In order to solve the problem of malicious peers increasing their rank by simply creating many new pages pointing to his, PageRank is refined with the following intuitive description: "a page has high rank if the sum of the ranks of its backlinks is high".

In [83] Jon M. Kleinberg defines a relationship between two new entities in the web graph: *authorities* (pages that have "authoritative" information about topics) and *hubs* (pages that have links to many important pages on the same topic). Kleinberg uses an algorithm called HITS (Hypertext Induced Topic Selection) to discover "authoritative" sources about topics. The idea behind HITS is called *"Mutually Reinforcement"*: a good *hub* is a page that points to many good authorities and a good *authority* is a page that is pointed to by many good hubs.

The SALSA algorithm (Stochastic Approach for Link-Structure Analysis) [90] is an equivalent weighted in-degree analysis of the link-structure subgraphs, making it more efficient than the Mutual Reinforcement approach of HITS. In SALSA the subject that web sites pertain to a given topic is split into hubs and authorities is preserved. However, it replaces the Kleinberg's Mutual Reinforcement approach by a new stochastic approach. In SALSA an *informative link* is a link $p \rightarrow q$ where a page p suggests (can also be seen as a recommendation) surfers visiting p to follow the link and visit q. The idea behind this algorithm is that a random walk will surf more likely (with high probability) the t-authorities of our web subgraph.

The World Wide Web is changing continuously. If we have a rank algorithm that orders the results, we expect that this rank algorithm will return similar results with small perturbations into the source set. For example, if we send a query to a search engine and we receive one webpage as the second result, we would expect that in the next day, sending the same query, this webpage would still be between the best results. This is called *stability* [103, 102].

HITS [83] and PageRank [114] calculate the principal eigenvectors of the associated matrices based on the link structure of the web graph. These eigenvectors based methods are sensible to perturbations. [103, 102] use some ideas from matrix perturbation theory and Markov chain theory to test the stability of these algorithms. The results of their study show how HITS is highly unstable with specific changes of the web graph. PageRank shows to be more stable because of the dumping factor ("teleportation" factor), what authors in [103, 102] called "reset-to-uniform". This dumping factor makes PageRank almost immune to the perturbations. Based on these results they proposed a new algorithm called *Randomized HITS* where they make use of the apparent immunity of PageRank adding a dumping factor to the HITS algorithm. The idea is to have a random surfer that will walk forward and backward in our web graph.

Based on the previous study, another algorithm is proposed in [103] (and it was already mentioned in [83]). This algorithm is called *Subspace HITS* and it tries to combine several eigenvectors in order to increase the stability. In this new approach the idea is to find k eigenvectors and combine them (giving an appropriate weight to each one according to the eigenvalues). This approach is much more stable than the original in HITS.

If we look at the development of the search engines we can realize that at the beginning only text techniques were applied in order to satisfy user queries. Now several techniques are used including some related to graph theory. The same applies to the problem of finding related documents. Many text techniques has been investigated and are being investigated but in [79] a new technique based on a graph-theoretic model is presented. It is considered that "two objects are similar if they are related to similar objects". In the web two pages are related if they have hyperlinks between them. Therefore, if we have two pages and they are pointed to by similar pages then they are similar. The base case is that pages are similar to themselves.

The algorithm can also be used in recommendation systems if we define two different kinds of objects in the graph (users and items for example). If two persons purchase similar items we can conclude that these two persons are similar. Moreover, if some items are purchased by similar people we can also conclude that these items are similar. We then have a *mutually-reinforcement* relationship

Other relevant algorithms include Latent Semantic Indexing [35, 115] (exploits dependencies or "semantic similarity" between terms), the CLEVER algorithm [20] (a variation of HITS [83]), the PHITS Algorithm [27] (a statistical algorithm that produces a *likelihood function*), and WebQuery [18] (a link-based analysis implemented in a similar manner to HITS but lacking the notion of hubs and authorities).

Personalized Ranking

All the previous algorithms provide the same ranks for different users, that is, they are not personalized. Some initial efforts for personalization are described in the following.

In [114], the authors already gave a notion of this personalized ranking. They called it *Personalized PageRank*. PageRank has a source vector for the ranks and it is used to represent the random surfer jumping to a random page of the web graph with higher or lower probability. However, it is also a good method to adjust the ranks scores. Normally, PageRank is calculated with this vector initialized to a uniform vector so all the pages of the web graph will have the same importance and will contribute in the same way to the ranks. It is possible to bias the PageRank computation by varying this vector in such a way that some webpage recommendations are weighted higher than the rest. This method allows personalized rankings but has two big problems: it needs to have a different vector of rankings for each user (this is not possible due to potential amount of users and the limited storage resources) and a different calculation is required for each one of them (this is also not possible as the calculation is time consuming and time is also a limited resource).

Topic-sensitive PageRank [66] chooses the 16 main topics of the Open Directory Project [113] to compute personalized ranks for each of them and each page (so each page will have one rank for each topic associated). Then, at query time the search engine combines the ranks from the different topics with appropriate weights.

Scaling Personalized Web Search [78] uses *partial vectors* which are shared across multiple personalized views in order to scale their computation and storage with the numbers of views. Each user selects a subset of a given set of hubs. Each user's Personalize PageRank Vector can be expressed as a linear combination of basis vectors which are decomposed into partial vectors (which encode the part unique to each page, computed at run-time) and the hub skeleton (which captures the interrelationships among hub vectors, stored off-line) in order to make the algorithm scale.

Rank aggregation

In many cases, there are no relationships or links among resources at different repositories. Sometimes there is information about user ratings or consumption and then ranking methods based on link-analysis can be applied [79]. Otherwise, different approaches need to be explored.

Rank aggregation is a term that refers to the ability to retrieve ranks from different sources and aggregate them in a single ranking. The main requirement is that there exists overlap of the content in those different sources. Then, it is possible to merge the rankings from those sources into a single one. [43] presents different aggregation heuristics in the context of meta-search engines or combination of different ranking algorithms. In the case of web search, the overlap of resources has to be visible in the results returned from the search engine (e.g., top 500), normally limited because of privacy (to avoid reverse engineering on its algorithms) and performance. [49] describe how to compare partial ratings and propose several metrics. Also, many search engines return only qualitative ordering without any quantitative measurement. [86] presents a method for rank aggregation under this condition based on the representation of the meta-search problem in a directed graph and the execution of the Majority Spanning Tree algorithm [82].

[57] addresses the problem of merging results from different sources which do not provide any ranking at all. It relies on the expansion of the query into other similar queries which results are later weighted and combined in a single ranking. The main limitations of this approach correspond to the selection of the proper weighting values as well as the decrease of performance due to the multiplication of the number of queries.

Chapter 5

Common Query Interface

In order to achieve interoperability among learning repositories, implementers require a common communication framework for querying (see section 4.3.1). Although there exist some approaches to provide search protocols or interfaces, they are not yet widely adopted. The main reasons are the complexity of the interfaces which leads to high development cost and lack of adequacy to available scenarios (e.g., stateful or stateless communications, synchronous or asynchronous interactions, etc.). This chapter describes a set of methods referred to as Simple Query Interface (SQI) [133, 130, 8] as a universal interoperability layer for educational networks. In addition, a solution to integrate distributed networks inherently asynchronous like P2P networks with other systems or networks is presented. Such an approach benefits from the simplicity and power of the Simple Query Interface to perform such an integration and shows how similar integrations with other systems or networks may be performed.

The main contribution of this chapter is the specification and standardization of a simple query interface to be adopted by systems willing to be interoperable. SQI provides a simple solution in terms of number of methods and implementation costs as well as a flexible solution that targets different interoperability scenarios, including for example, synchronous, asynchronous, stateless and stateful communications. SQI is an official CEN/ISSS Workshop Agreement since October 2005. Furthermore, it is also one of the protocols listed in an official document published by IMS on Query Services [75] and it has been adopted by a large number of repositories making possible networks of repositories that did not exist before its creation (see chapter 9 for more details). In addition, a proxying architecture in order to open (typically) closed environments to other consumers and providers is specified. This architecture has been used by several systems (e.g., EDUTELLA) in order to bring interoperability to heterogenous networks of repositories.

5.1 Simple Query Interface

The Simple Query Interface (SQI) [133, 130, 8] is a protocol designed in order to provide a simple, powerfull and highly-flexible solution to interoperability problems[1]. During its design, different issues were especially taken into account as key points for its success:

- Some interfaces have tried to fix a query language, vocabularies or specific data formats therefore restricting its applicability and success on different domains. For that reason, SQI is defined to be agnostic respect to query language, result formats and vocabularies used.
- Most interfaces assume that queries will be sent to one single repository which is going to return results in a synchronous manner (as in database like approaches). However, as it will be described later in this section, it is envisioned that more complex kinds of information sources may be queried, which may require an asynchronous protocol for returning results, such as P2P networks or front-ends for federated search.
- Although due to performance and time response requirements many implementations of search mechanisms are developed in a stateful manner, some lightweight implementations may require stateless interfaces. SQI allows for both kind of implementations.
- Although a query service should be aware of security (authentication and authorization) mechanisms, it should be possible to separate both issues as much as possible (to enhance modularity). This way, security mechanisms can be changed without affecting the query service. In SQI, the concept of session management is exploited in order to provide such a separation between core services and application services.
- In order to allow for easy extensibility of the interface while preserving backwards compatibility, it was decided to follow an approach in which different methods with smaller number of parameters are created. This way, if new requirements arise, new methods can be created without changing the signature of existing ones.

According to these design principles and requirements, two sets of methods have been specified and are detailed below: one regarding the authentication and session management core service and another related to the query

[1] Since October 2005 SQI is an official CEN/ISSS Workshop Agreement. Furthermore, SQI is also one of the protocols listed in an official document published by IMS on Query Services [75].

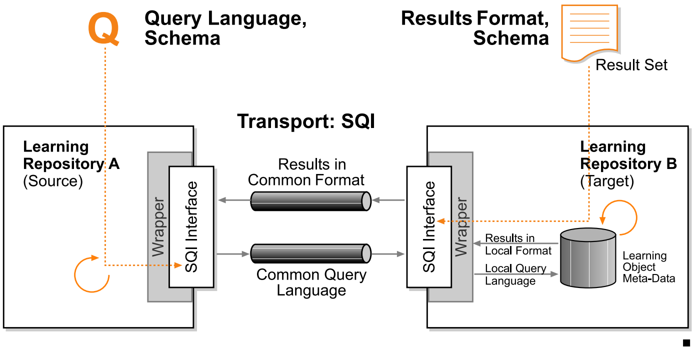


Figure 5.1: SQI Process [130]

application service. In the following, the system querying for information will be named *requester* (also called *source*) while the system offering the information is called *provider* (also known as *target*) (see figure 5.1). In first case, the requester needs to authenticate at the provider. After successful authentication a session is retrieved. This session is used during following interactions with the query service in order to avoid repeating the authentication mechanism (and therefore avoiding resending user names and passwords repeatedly). Before a query is sent, the requester may configure some parameters affecting the way the interaction will take place (e.g., query language to be used or maximum number of results to retrieve). If none of this parameters are specified within the session, defaults given by the provider are assumed. Finally, the requester has two different invocations of the query service: synchronously or asynchronously. The former sends the query and awaits till an answer is retrieved. This is especially suited for database-like repositories in which all results are known at once at the repository side. The latter sends the query and immediately returns the control to the requester, without waiting for any result. The provider will then later send new results (possibly in several messages) as soon as they are retrieved. This paradigm is especially interesting for information sources in which not all the results are retrieved at once but in several steps and without any time guarantee. Finally, once the requester is not interested anymore on querying the provider, it may destroy the session created before. Figure 5.2 summarizes this process and shows how the methods provided in the SQI specification are applied.

5.1.1 Authentication and Session Management Service

Authentication is a strong requirement when dealing with information since it may not necessarily be public and it enables traditional mechanisms for

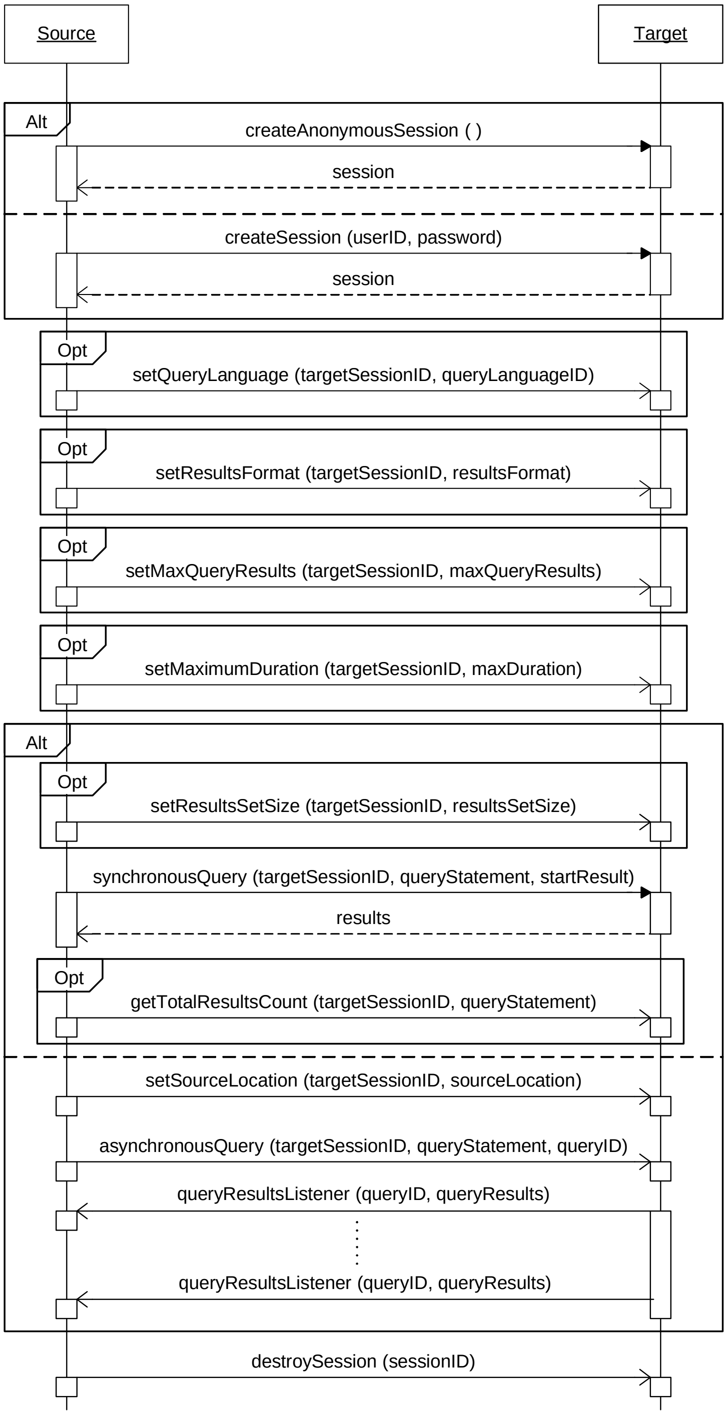

Source
Target
Alt
createAnonymousSession ( )
session
createSession (userID, password)
session
Opt
setQueryLanguage (targetSessionID, queryLanguageID)
Opt
setResultsFormat (targetSessionID, resultsFormat)
Opt
setMaxQueryResults (targetSessionID, maxQueryResults)
Opt
setMaximumDuration (targetSessionID, maxDuration)
Alt
Opt
setResultsSetSize (targetSessionID, resultsSetSize)
synchronousQuery (targetSessionID, queryStatement, startResult)
results
Opt
getTotalResultsCount (targetSessionID, queryStatement)
setSourceLocation (targetSessionID, sourceLocation)
asynchronousQuery (targetSessionID, queryStatement, queryID)
queryResultsListener (queryID, queryResults)
queryResultsListener (queryID, queryResults)
destroySession (sessionID)

Figure 5.2: UML 2.0 sequence diagram describing SQI

authorization. A repository may want to restrict access to the information depending on the requester or to limit the number of queries allowed in a given time frame (e.g., a day). Therefore, a means to identify a requester are needed. Furthermore, other application services should work independently of the way authentication is realized (e.g., via user name and password, provision of a credential or through a trust negotiation [139, 55, 9]). This flexibility is achieved by using sessions as the result of successful authentications. This way, once a requester successfully authenticates, the provider provides a session valid for a period of time. This session is then used by the requester for successive interactions. This way, the provider is able, given a session, to identify the requester and apply any authorization mechanism or policy to the current request while at the same time, the requester does not need to repeatedly authenticate (which may be costly as well as risky). Moreover, this separation between authentication and application services allow a new authentication method to be added in the future, its integration would be straight forward without the need of any change on the application services as well as its reuse for any other application service (e.g., for publishing).

SQI currently defines three methods for authentication and session management[2]:

createAnonymousSession

Args: *None*
Return value: *String session*
Description: This method may be used to create a session in repositories where no account is required or in which a *guest* account is enabled. It takes no argument and returns a valid session.

createSession

Args: *String userID, String password*
Return value: *String session*
Description: In case that authentication is an issue, then invoking this method may be required by repositories. It receives a user name and password and, if those map into a valid account at the provider, then a valid session is returned.

destroySession

Args: *String sessionID*
Return value: *Void*
Description: This method receives a valid session and destroys it,

[2] Defined exceptions are not included in this brief description as the reader is referred to [133] for a more detailed description

	Implemented at the target and called by the source	Implemented at the source and called by the target
Query Parameter Configuration		
setQueryLanguage	X	
setResultsFormat	X	
setMaxQueryResults	X	
setMaxDuration	X	
Synchronous Query Interface		
setResultsSetSize	X	
synchronousQuery	X	
getTotalResultsCount	X	
Asynchronous Query Interface		
setSourceLocation	X	
asynchronousQuery	X	
queryResultsListener		X

Table 5.1: Overview of Simple Query Interface Methods

meaning that such a session will not be accepted anymore at the provider.

5.1.2 Query Service

SQI specifies different methods in order to configure parameters related to the query service as well as to request synchronous or asynchronous queries (table 5.1 lists all these methods). The following provides a brief description of these methods (see [133] for more detailed information).

setQueryLanguage

Args: *String targetSessionID, String queryLanguageID*
Return value: *Void*
Description: SQI is query language independent, what means that it is not restricted to a specific query language. Therefore, this method establishes the query language to be used during the communication. It may vary from simple keyword based queries to more advanced and expressive query languages like QEL. Values of the *queryLanguageID* parameter are case insensitive.

setResultsFormat

Args: *String targetSessionID, String resultsFormat*
Return value: *Void*
Description: As with the query language, SQI does not specify any result format. This methods allows the requester to select a result format from those allowed by the provider.

setMaxQueryResults

Args: *String targetSessionID, Integer maxQueryResults*
Return value: *Void*
Description: This method specifies the maximum number of results that a query may produce. If *maxQueryResults* is set to 0 then no limit is established.

setMaxDuration

Args: *String targetSessionID, Integer maxDuration*
Return value: *Void*
Description: This methods defines a time out for a query request after which results are no longer needed. If *maxDuration* is 0 then the time out management is left to the provider.

setResultsSetSize

Args: *String targetSessionID, Integer resultsSetSize*
Return value: *Void*
Description: In many cases, too many results may be relevant to a given request (e.g., given some keywords Google may easily find million of relevant resources) but not all of them are needed (e.g., only the first 20 will be shown). In such cases, it is possible to configure the size of the result set retrieved from the provider. If *resultsSetSize* is 0, then all results are retrieved at once.

synchronousQuery

Args: *String targetSessionID, String queryStatement, Integer startResult*
Return value: *String resultSet*
Description: This method request a synchronous query and retrieves a result set starting from the result *startResult*. Therefore, this method may be called several times varying *startResult* in order to retrieve all available results in chunks given by the size of the result set (specified with the method *setResultsSetSize*). The results are returned as a String according to the specified results format.

getTotalResultsCount

Args: *String targetSessionID, String queryStatement*
Return value: *Integer totalResultsCount*
Description: This method provides the total amount of available results for a (possibly already given) query statement

setSourceLocation

Args: *String targetSessionID, String sourceLocation*

Return value: *Void*
Description: In asynchronous communications, it is needed to specify where the provider should send the results when retrieved. This method specifies the location of source's results listener.

asynchronousQuery

Args: *String targetSessionID, String queryStatement, String queryID*
Return value: *Void*
Description: This methods sends an asynchronous query to the provider. The query is specified via the *queryStatement*. A query identifier is also sent and will be used by the provider when sending back the results to such a query. This way the requester is able to distinguish the results for different (possibly parallel) asynchronous queries. The provider will return the results to the location specified by the *setSourceLocation* method.

queryResultsListener

Args: *String queryID, String queryResults*
Return value: *Void*
Description: This method is the only one that is invoked by the provider. It is used to send results of an asynchronous query to the requester. A *queryID* is provided as an identifier of the query submitted by the requester. Several calls to this method may be made for the same *queryID* as soon as the results are gathered by the provider.

5.2 Using SQI-based Proxies to increase Interoperability of P2P Networks

P2P networks are dynamic networks where peers can act as server and client indistinctly and peers might freely join and leave the network over the time. Obviously, peers must implement the specific P2P network protocol in order to connect to it.

General consumers and providers try to implement standard interfaces in order to maximize the effectiveness, implementation costs and effort. However, if they want to access or expose content in a P2P network, this requires the additional extra effort of implementing the specific network interface (one for each different P2P network to be accessed). This barrier makes P2P networks unable to interact with each other or with other systems and environments.

In order to solve this problem, we based our solution on proxies [145, 110, 15] that are used to connect peers in a P2P network with the "outside" world. These proxies bridge two systems with different capabilities by means of implementing the protocol and/or interface supported by each system respectively. This way, a proxy is able to forward requests and responses from one system to another.

Nowadays, systems try to provide their services/resources via standard interfaces like SQI [133] or OKI [107]. In our case, we have implemented proxies able to bridge the proprietary[3] Edutella/JXTA protocol and interface into a the Web Service binding of the Simple Query Interface.

Taking the P2P network as a reference, there are two different desirable scenarios [121]:

1. **An external consumer/client wants to query content in the P2P network.** For example, let us suppose that we would like to offer the content of a P2P network via Web Services and/or in a web site. The first solution would be to make the (web) server join the P2P network. However, the load of the server would increase considerably and even some problems could arise in case the server wants to provide content from more than one network (it would need to join all of them). A cleaner solution (and the one followed here) is to forward the query from the server to the P2P network by means of proxies and retrieve the answer with the same mechanism.

2. **An external provider wants to offer content to the P2P network.** It is assumed that providers that have already implemented a standard interface will not be happy spending more time and money in developing the specific interface(s) of the network(s) they want to join. In contrary, they would like to reuse the one they have which would also ease its administration (as only one interface needs to be maintained).

According to these two scenarios, there are two different types of proxies with different functionality. The former scenario requires the so-called "consumer proxy" and the latter the so-called "provider proxy" (names are assigned according to the role they play). A consumer proxy acts as a mediator between an external client that wants to query the network and the P2P network itself. A provider proxy acts as a mediator in order to provide the content of an external provider into the P2P network.

[3] Here we use the term "proprietary" to emphasize that this protocol is not standard for P2P networks but it does not mean it is not open. In fact, Edutella/JXTA is open-source and anyone can use it freely.

In the following, I first provide a brief description of Edutella [101], the P2P network chosen to interact with other systems. This P2P network was selected because of its ability to share and search for metadata. After this, a detailed description of the two scenarios and proxies described above is given.

5.2.1 Edutella

Often, learning object providers do not want to abandon control over their resources to a third party, not even among the members of a coalition. The same concern about abandoning control also often applies to individuals, who may not want to give away their content to any centralized repository. In order to deal with this issue, distributed environments have shown to be a feasible solution for interconnection, integration and access to large amounts of information. P2P networks are one example of the impact the distribution of information might have in the sharing of information. In such networks, peers can offer various services to the user ranging from search and delivery of content, to personalization and security services. In addition, they contribute to the solution of managing the information growth, and allow every learning resource provider to offer its information without loosing control over it.

The Edutella P2P network [101] was developed with these principles as main design requirements. Edutella is a schema-based P2P network for an open world scenario in which learning objects are freely offered (at no charge) and everybody is able to join (no agreement with an existing member of the network is required). It has various service facilities implemented, such as query or publishing/subscription. Schema-based means that peers interchange RDF meta-data (data about data) among each other but not the resources themselves, that is, they interchange information about e.g. title, description, language and authors of a resource. This information can be queried using the QEL query language [104] (based on Datalog). Metadata interchange and search services provide the basic infrastructure needed to retrieve information about resources and services.

5.2.2 Consumer Proxy

As described above in scenario 1, in some cases it is needed to be able to query a P2P network without the need of joining it. A consumer proxy is a peer which is part of the P2P network (and therefore it is able to send queries to and receive the answers from it) and which is also able to receive requests and send responses using a different protocol and interface. This way, an external client is able to query the P2P network through the proxy.

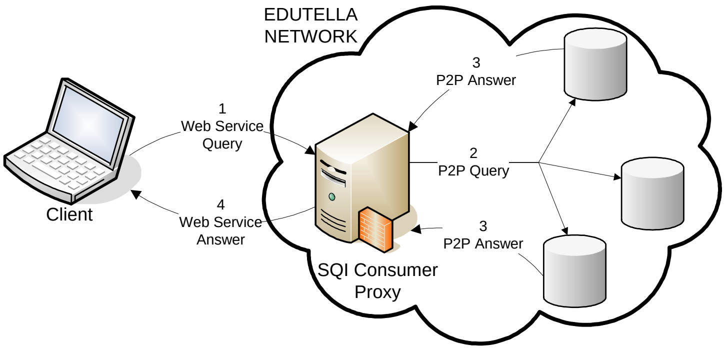


Figure 5.3: Consumer Proxy

In our implementation, a consumer proxy mediates between the Edutella/JXTA and the SQI protocol. As depicted in figure 5.3, it is responsible for

1. Receiving queries from external clients via SQI
2. Forwarding each query to the Edutella network using the Edutella/JXTA interface
3. Collecting the results sent from peers within the network using the Edutella/JXTA interface
4. Forwarding those results to each requester system via SQI

This simple mechanism allows any system to query the content of the Edutella P2P network without needing to implement its specific interface. In addition, the proxy can return the results to the client application

asynchronously. The results are sent to the client as soon as they arrive to the proxy. This is the typical mechanism in distributed environments as not all the results are generated at once but they must be gathered from the different systems in the network.

synchronously. The results are gathered at the proxy and sent in a single message to the client. Although this is not the intuitive way for a distributed environment it could be desirable in some scenarios (e.g., in mobile devices we do not want our device to receive a new message every time a new result arrives to the proxy but better ask for new results in a proactive manner).

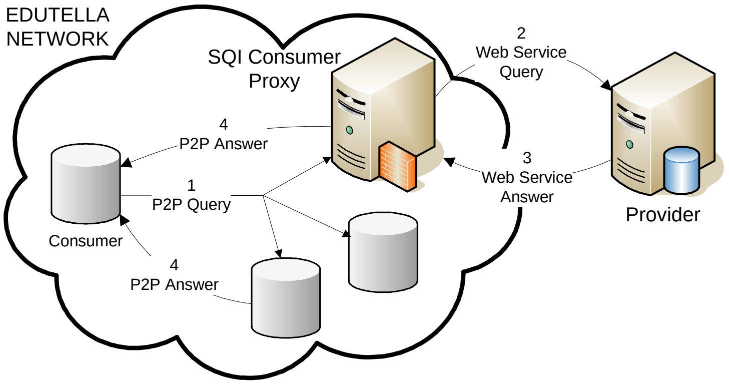


Figure 5.4: Provider Proxy

5.2.3 Provider Proxy

In order to fulfill the scenario 2, a second type of proxy has been developed. This provider proxy is a peer connected to the P2P network which also is able to send requests and receive responses by means of a different protocol and interface. Therefore, it is able to forward queries to external providers and receive their answers providing their content to the network.

As in the case of consumer proxies, the provider proxy mediates between the Edutella/JXTA and the SQI protocol. As depicted in figure 5.4, it is responsible for

1. Receiving queries from peers in the network using the Edutella/JXTA interface
2. Forwarding each query to the external provider via SQI
3. Receiving the results from the external provider via SQI
4. Sending those results back to each peer that had submitted the query using the Edutella/JXTA interface

Due to the asynchronous nature of a P2P network, it is possible for the provider proxy to receive the results from the external provider in a synchronous (e.g., in case the external provider is a relational database) or asynchronous (e.g., if the external provider is another distributed environment) way.

5.2.4 Combining Proxies

Thanks to the proxy types described above, it is now possible to open the P2P network to integration with other systems in both consuming and pro-

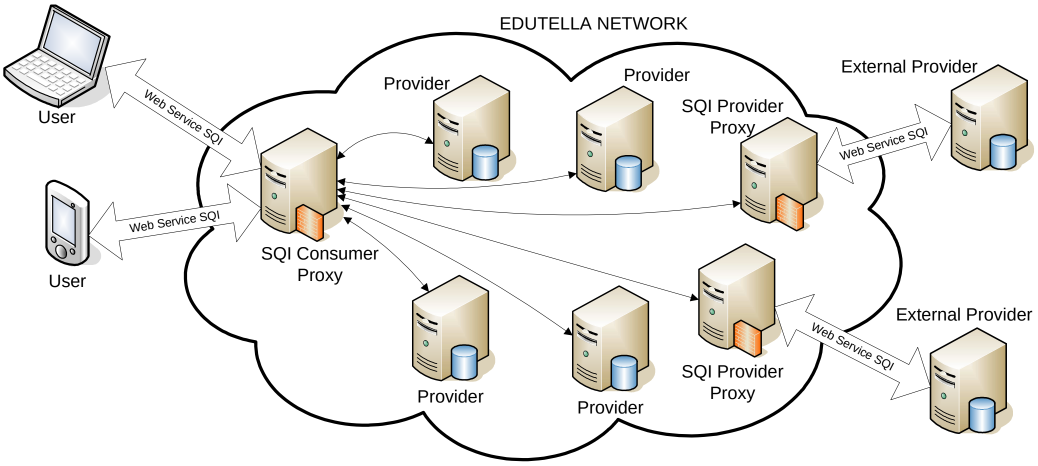


Figure 5.5: architecture

viding information (see figure 5.5). Joining the P2P network can now be done through a standard interface. This way, a consumer can use the same interface for querying the network or any other repository outside the network supporting the standard. At the same time, a provider may offer content to a network (P2P or federation) by implementing a single search interface.

Chapter 6

Common Metadata Schema

Once a common interface for communication has been specified an agreement on a common vocabulary is required. Otherwise, two systems would be able to talk to each other but that does not mean they are able to actually understand each other. As describe in section 4.3.3, there are several approaches which standardized vocabularies such as Dublin Core or LOM. However, on the one hand, Dublin Core is cross-domain and much too general to work alone on the e-learning domain. On the other hand, LOM describes up to 70 attributes (probably too many) where all are optional, therefore not ensuring an homogeneous set of metadata in order to provide advanced services using it. Furthermore, some attributes are not specified enough in order to be suitable, for example, for some business scenarios (e.g., no price attribute is given).

This chapter describes two complementary schemas[1]. The first provides a schema for learning resource annotations. It includes an attribute *competence* which is used to define prerequisites to access an activity as well as what is acquired once an activity is successfully completed. In this first schema, it is not defined what a competence is or what structure it has (if any). The second schema provides a model to describe rich competence descriptions in order to better suit learners' goals in different contexts and at different proficiency levels. In both cases, it was aimed to find a good compromise between expressivity and ease of adoption.

Ontologies are created in order to bring interoperability to the resources they are used to describe. The main contribution of this chapter is the creation of ontologies and data models to annotate learning material and to represent required or acquired competences. An ontology for learning resources is presented in order to classify and describe the metadata that

[1] In reality three since the first is composed of two different schemas, as the reader will see below.

Abbreviation	Namespace	Name
dc	http://purl.org/dc/elements/1.1/	Dublin Core
lom:rights	http://ltsc.ieee.org/2002/09/lom-rights#	LOM-Rights
openq	http://www.open-q.de/	Open-q
ln	http://www.hcd-online.com/nsv1/	HCD-Online

Table 6.1: Namespaces used in the learning resource schema

may be later used to find potential relevant material. In addition, a data model is provided in order to represent competences in an interoperable way allowing more advanced querying, filtering and ranking mechanisms.

6.1 Learning Resource Schema

This section briefly presents part of the learning resource schema [40, 41] mainly developed as part of the EU IST Elena project [48]. This project aimed at providing a Smart Space for Learning™ in which semantically enriched services would be composed to enhance the user experience.

There are however two problems here. On the one hand, in order to provide personalization and advanced services it is required to have a big set of annotations for each resource available. On the other hand, most institutions already have a set of metadata (typically in databases) and adding extra annotations may not be possible due to their current business workflows or an increase in their costs. As a decision, it was decided to provide two different schemas: a basic one each institution should be compliant with in order to ensure the basic functionality of the "space for learning" and a complete one, optional, for those institutions who want to benefit from the "smartness" providing, for example, advanced personalization.

After an analysis of the metadata available in learning management systems, universities and other small and medium companies in the e-learning domain, as well as the requirements to provide basic functionality that is useful for users (e.g., semantic search) the following attributes were selected: identifier, title, description, language, classification, category, additional information, learning goal, with cost and with restrictions. A description of each one of them is given in table 6.2. From these attributes perhaps the only one that requires an explanation is *ln:learning_resource_category*. Basically, this schema divides a learning resource in two subgroups: learning material and learning service. The former describes learning units, that is knowledge resources annotated with learning metadata that can be used in isolation (e.g., downloaded). Some examples include books or articles.

Attribute	**Description**
dc:identifier	A globally unique identifier of the resource (typically composed by a globally unique identifier of the provider and a local identifier of the resource)
dc:title	Title of the resource as free text
dc:description	Description of the resource as free text
dc:language	Language of the resource according to ISO639-2 `http://www.loc.gov/standards/iso639-2/langcodes.html`
dc:subject	Classification of the resource according to a specified taxonomy
ln:learning_resource_category	Either learning material or learning activity
ln:add_information	Any additional information. Typically will contain a link to the learning resource or a page with information about its access
openq:goal	Learning goal as free text
lom-rights:cost	Price flag specifying whether the learning resource has a cost (value "Yes") or is available for free (value "No")
lom-rights: copyright_and_other_restrictions	Restrictions flag specifying whether there are additional restrictions applicable to the learning resource (value "Yes") or not (value "No")

Table 6.2: Basic learning resource schema

The latter describe learning events which typically take place at a specific place/environment and/or time such as a course or a conference.

These attributes represent the minimum set of information that each institution must provide in order to join a learning network in which material will be searched for and retrieved. Only restricted personalization (e.g., based on language and learning goals in the profile) can be applied with this basic schema.

If an institution is interested in joining the actual *Smart Space for Learning*, then a more complete set of annotations must be provided (see figure 6.1). This schema is based on existing standards like, among others, Dublin Core, LOM or VCARD. This schema will not be described in detail here since it is out of the scope of this thesis (see [41] for more detailed information on the schema). However, an important point to notice is that any learning resource may have attached competences. They can be *required* to access or fully un-

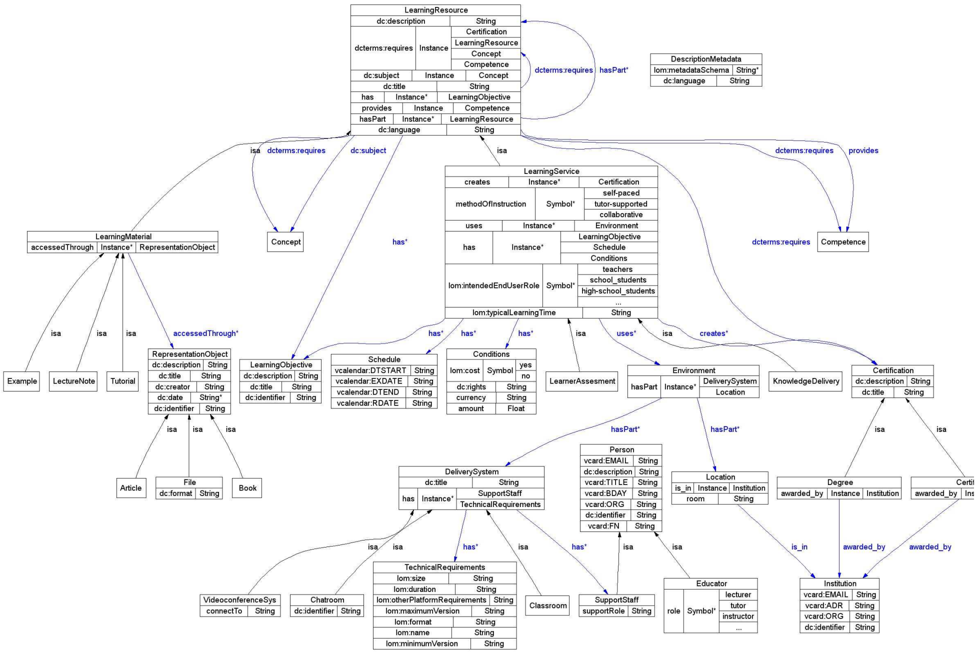


Figure 6.1: Elena common schema [41]

derstand the learning resource or *provided* after successfully completing the learning resource. In any case, a competence can be considered a simple free-text label, an entry in a taxonomy or a more complex structure in which reasoning can be performed. The following section presents a rich model for competence descriptions that could be exploited for gap analysis (e.g., for Human Resources analysis) or for finding relevant learning resources (e.g., according to the learner profile).

6.2 Competence Model

Nowadays, the mobility of people has increased. Learners may study abroad with the benefits of improving their language skills, receiving a better certification, or specializing in a topic not available in their regions. The same applies to the labour market. People do not need to restrict to their city or region while seeking for a job but may consider offers in other countries, too. This situation complicates the already difficult job of managers in learning organizations and Human Resource (HR) departments to decide who may have the right qualifications to join a project or the company itself. For learning organizations, requirements to join the program must be taken into account. For example, an applicant needs to possess a Bachelor degree to apply for Master studies; in order to attend an expert course on a topic, a certification on a basic level may be required. Furthermore, assuming that an applicant fulfills such requirements, exemptions could be granted for parts of the program that are similar to earlier followed courses. Imagine a mathematician starting a Computer Science degree. Most likely, courses like Algebra and Statistics could be exempted. In the case of Human Resource departments, the task is equally complex. HR experts need to match applicant or employee experience and knowledge with the requirements of a job offer or a project, including both mandatory requirements and desired ones (e.g., Business English is required and French would be a plus). The same applies to learners trying to find appropriate courses or learning resources in order to improve their knowledge or to fit their goals but without requirements which they do not yet know. (e.g., for example a student may want to learn Java programming and therefore an introductory course would be recommended while an expert course would be too far away from her current knowledge). Currently, all these competence matches have to be performed manually (therefore time consuming and error-prone), with hardly any guidelines or support, and therefore they are not suitable for automatic searching mechanisms (e.g., a user may not include competence information along with a query in order to further restrict potential relevant results). One important reason is the lack of interoperability of competence descriptions that may be exchanged among learners and systems. Competences are typically represented differently in user profiles stored in each repository and existing stardards are currently no sufficiently expressive for the representation of competences, which is needed for complex competence profiles and requirements.

Some initiatives, such as the IEEE Reusable Competency Definition (RCD) [123] and HR-XML [70], have done initial steps to define common models and schemas for interoperability, but their current work lacks some important information that is required for competence matching, like proficiency levels or context (see section 4.3.3), or for increasing reusability. In this section, the work that has been developed under these various initiatives is enhanced and extended and a model for representing competences with their relationships as well as some usage profiles (such as profiles for job requirements description or for learner achievements description) [28] is introduced. This model provides the basis for allowing advanced (semi-)automatic competence matching and gap analysis, which might be used, for example, by Human Resources departments or by tools e-learning systems provide in order to recommend courses or allow exemptions according to learners's profiles.

6.2.1 Motivating Scenario

Typically, a recruiter in an HR department would write a job offer[2] like

> Wanted: J2EE consultant
>
> - Completed Master's Degree (any faculty)
> - Expert Knowledge in Java J2EE, Servlets, JSP
> - Very good English and/or French

Among other drawbacks, such an advertisement does not indicate what is mandatory or optional and, more importantly, it is not machine-understandable. Performing a manual matching (as widely performed now from the recruiters), the recruiter will have a hard time matching applications against this offer. An interoperable representation for competences is therefore required.

6.2.2 What is a competence?

In this work the following definition of *competence* is adopted: "effective performance within a domain/context at different levels of proficiency" [54]. Note that there exists some confusion on the term *competency*[3] in the literature. [123, 76] define the stricter term of competency as "any form of knowledge, skill, attitude, ability, or learning objective that can be described in a context of learning, education or training". This definition is insufficiently expressive for competence gap analysis. For example, it is not clear if "piloting" covers both the ability to pilot a small plane and to pilot a big passenger airplane. Or if the competency "English writing skills" represents a specific level such as intermediate, fluent, native or simply the existence of the competency. In fact, if that information becomes part of the competency definition, its reusability is drastically reduced (with the consequence of, e.g. having different competency definitions for each context in which a competency is applied, and for any proficiency level and proficiency level scale). The definition given in [70] tries to extend the previous one: "A specific, identifiable, definable, and measurable knowledge, skill, ability and/or other deployment-related characteristic (e.g., attitude, behavior, physical ability) which a human resource may possess and which is necessary for, or material to, the performance of an activity within a specific business context". In

[2] Excerpt extracted from a newspaper

[3] The reader is alerted for the distinction between the two terms, *competence* and *competency*

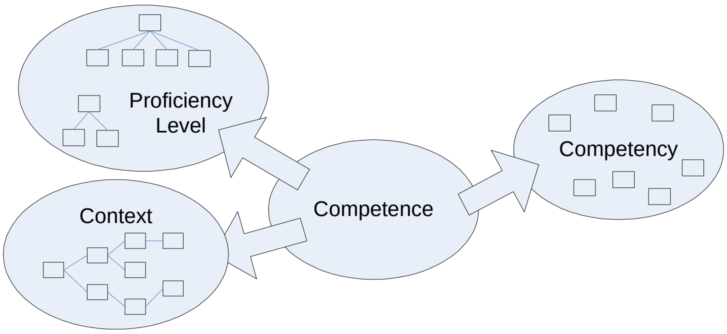


Figure 6.2: Competence as a combination of competency, proficiency level and context

this case, "measurable" indicates a relationship with a specific proficiency level[4] and competency now applies only to the business context. In any case, since context is implicit, the models proposed from these specifications do not include context information.

As stated above, current approaches to modeling competencies do not explicitly address proficiency level and context. On the contrary, competency, proficiency level and context are three different dimensions that should be modeled separately in order to maximize their reuse. For example, the same competencies may be used in different contexts, or the same proficiency level scales may be reused among different certifications. The same applies to contexts (or "domain models"), which in many situations already exist and therefore may be reused by competences. Therefore, according to what stated above, a competence (plural:competences) is here modeled as a three-dimensional variable, made up of a competency (plural:competencies), a proficiency level and a context (see figure 6.2). For example, "Fluent Business English" would be composed of the competency "English", the proficiency level "Fluent" and the context "Business".

For sake of clarity, and in order to avoid confusion between the terms competence and competency, competency and skill may be used interchangeably hereafter. However the reader should be aware that skill is not a synonym for competency, as it only covers part of its scope.

6.2.3 Modelling a Competence

In this section, a model for representing a competence with a broader and clearly defined view is introduced. This model is based on the three di-

[4] Although they later refer to it as "grade", which is different from proficiency level (see section 6.2.4)

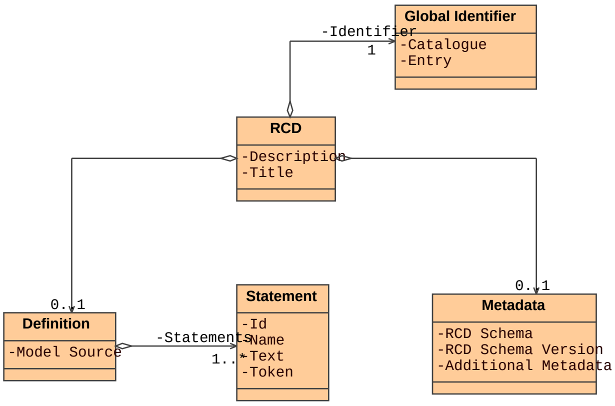


Figure 6.3: IEEE RCD Model

mensions that a competence is composed of: competency, proficiency level and context. First each dimension is described separately and finally it is presented how they are combined in order to build a competence and how competences may be composed of sub-competences. Several issues encountered during the modeling process, and possible solutions (eventually with a trade-off between expressiveness and complexity) are described. The decisions we have taken as well as their features and the limitations derived from them are also discussed.

Competency

The IEEE Reusable Competency Definitions (IEEE RCD) [123] provides a model for the representation of competencies (figure 6.3). This model does not include proficiency level or context information. In addition, as stated in the specification, IEEE RCD is "intended to meet the simple need of referencing and cataloging a competency, not classifying it", that is, it does not provide any means to specify relationships between competencies. We agree upon this view and believe that relationships should not be modeled at this level because they also depend on the other two dimensions: proficiency level and context. For example, piloting cannot be related to other competencies without knowing if it refers to helicopters, small planes or passenger planes.

The ideas described in [123] meet our requirements, with the advantage that this work is already acknowledged from the community as a draft standard. Therefore, IEEE RCD's model is reused to represent competencies (see model depicted in figure 6.3).

The RCD identifier provides the basis for referencing and reusing such RCDs, while title and description provide free text to represent them. It is assumed the existence of repositories of RCD elements which may be referenced from different competences by their global identifier. In addition, [123] includes information which is not thought to be machine-processable[5] but for human interpretation. Such information includes structured descriptions for more complete definitions.

Proficiency Level

Different scales (qualitative and quantitative) may be used in order to represent proficiency levels. For instance, a computer science curriculum may simply want to specify whether a student has acquired a competence or not, whereas an English certification institution may want to classify students into intermediate, advanced or proficient. Many different scales may be used but it should be possible to reuse them within and across the borders of the institution. For example, scales are typically the same for most certifications given by one institution and even among them (e.g., all curricula in Spanish universities). Hence, they can be modeled once and referenced many times.

Proficiency levels are not simply a flat set of elements. There are implicit relationships among elements within one scale. For example, a proficiency level may be subsumed by another ("proficient" subsumes "advanced" which subsumes "intermediate" and so on). We need to model such relationships because they will be needed for competence matching. For instance, a job requiring someone with intermediate English skill typically has the implicit quantifier "with at least", meaning that anyone with advanced English would also be accepted (and maybe even preferred). In order to represent these relationships, an ordered list provides a reasonable means to represent a proficiency level scale (see figure 6.4). In such a list, the minimum value (subsumed by any other in the list) is given by the first element and the maximum is given by the last one. Therefore, the order in the list represents subsumption relationships, that is, the first element is subsumed by the second one which is as well subsumed by the third one and so on.

In order to improve interoperability and matching among scales, an optional field is included for mapping to a universal scale (e.g., [0,1]). The reason why this mapping field is optional is that even though it would be useful to include it, in some contexts it may not be possible to find a suitable mapping or it may not even be necessary (e.g., if a scale is used only within an institution and no interoperability is intended).

Competence descriptions can refer to specific items of these scales in order to represent the proficiency level acquired/required. Algorithms could take

[5] Do not confuse with *machine-exchangeable*

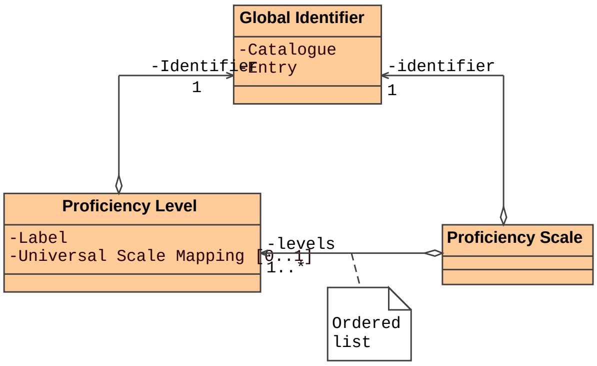


Figure 6.4: Proficiency Level Model

relationships among proficiency levels into account in order to find out how much training/learning is required to reach a determined employee/learner proficiency level. For example, if advanced English skills are required, training an employee who already acquired intermediate English skills will cost less time and money than training another employee who has only beginner English skills.

Context

Context is defined as "the interrelated conditions in which something exists or occurs" [154], which includes "the circumstances and conditions which surround it" [156]. Regarding competences, context may refer to different concepts. It might be the specific occupation in which a competence is acquired (e.g., driving as an ambulance driver or as a pizza delivery employee), a set of topics within a domain (e.g., telecommunications or tourism, or theoretical vs. applied physics) or even the personal settings related to the learner (e.g., competences are different if acquired in a group-based learning setting than individually). All these (and possibly more) are contexts which may be part of a competence. What actually makes up sufficient context descriptions can not be defined in general, but depends on the scope and purpose of the competence descriptions to which they are attached. As with the skill definitions and proficiency levels, context definitions may be reused.

Modeling contexts may be a complex task, as it may coincide with modeling the entire domain knowledge of an institution. Ontologies can capture such knowledge [88] and use arbitrary complex structures, from simple sets or tree structures to directed acyclic graphs. Up to date, our investigations of existing relationships between context elements (regarding its use within competences) do not show the need for providing a graph representation or

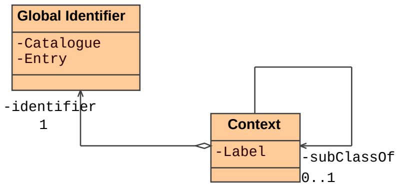


Figure 6.5: Context Model

multiple inheritance. For this reason, it was decided to first restrict the modeling of context to trees (see model[6] depicted in figure 6.5). This has multiple benefits:

- it reduces the computation complexity of competences
- it is easier to understand by users
- it avoids the need for cycle-detection mechanisms while modeling is done
- it simplifies the algorithms for competence matching.

We are still investigating the advantages and drawbacks of this decision and an extension of the model may be required in case some scenarios for which such a structure would be beneficiary are found. Allowing for more advanced algorithms could also be a reason for choosing a more expressive context model. Furthermore, the relationship among context concepts may also be used by algorithms analyzing competence gaps. For example, assume that a context models all occupations of an airline company within an airport. If it is needed to train a new pilot for passenger flights, it would be preferred to train some of the pilots of cargo planes instead of a person from the check-in counter. This information could be extracted from, for instance, distances between the occupation "pilot" and the rest of occupations in the tree/graph.

Competence

Competences are described as reusable domain knowledge. Any model representing competences describes what a competence is and how it is composed of sub-competences. These competences are general descriptions, independent of specific learners or job descriptions. For example, being a good taxi

[6] The set of attributes in the context structure is the minimum one allowing reference and reuse. This model may of course be extended with more data specific for the areas in which it is used

driver or an expert Oracle database administrator are concepts with fixed meaning (domain knowledge), independent of which person possesses such competences. This is important to be noticed, because competences are to be referenced from certifications or job descriptions, in order to stimulate their reuse. For instance, a company may define required, relevant or desirable competences for their business, which are included in job offers or projects descriptions. The exact meaning of these competences is provided by a company-wide competence model. Using this approach, the explanation of a competence needs not to be explicitly included every time it is used[7]. These explanations may cover a broad range of aspects, such as:

- how a competence may be achieved, for example by acquiring some sub-competences;
- to which level each competence should be acquired. As an example, scientific research in a University may require only basic knowledge of mathematics while at NASA, expert knowledge is needed;
- whether sub-competences must be all achieved or simply a subset of them. For instance, it is typical in curricula that in order to get a degree, some topics are mandatory and some other are optional, from which a subset has to be chosen (e.g., pass k optional courses out of n offers);
- if the sub-competences must be acquired in a specific order. Some companies may require that an applicant acquired a competence on personal task organization *before* becoming a good team leader. Otherwise, they may assume that the performance related to the competence of being a good team leader is reduced.

In order to model all these elements an object model derived from the Composite design pattern [126] (see figure 6.6) has been created. In this model, a competence can be either simple, an aggregation of children, or a selection from children alternatives. `Competence` models a competence, with references to a skill (RCD id), a proficiency level and a context. It can be a `SimpleCompetence` (an atomic description) or a `CompositeCompetence`. The latter can be either be an `AggregateCompetence` or `AlternativeCompetence`. An `AggregateCompetence` can be used to define a competence which consists of several sub-competences, all of them required. The sub-competences can be either an ordered set (meaning that the sub-competences must have been

[7] As with the use of ontologies, whose classes can be simply referenced without the need of copying the whole ontology every time they are used

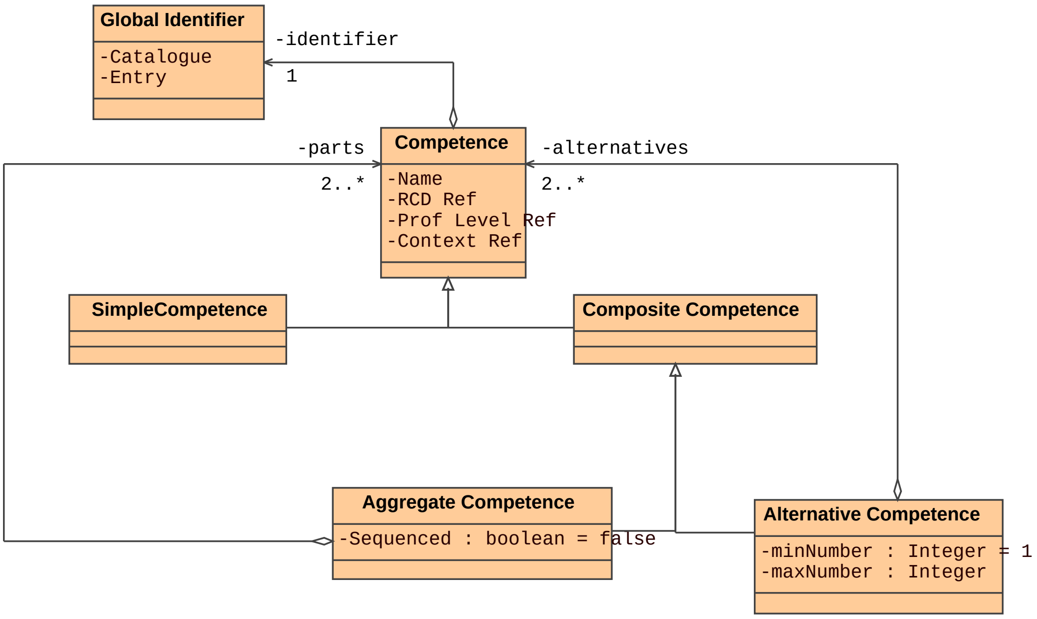


Figure 6.6: Competence Model

acquired in such an order) or unordered (default). An `AlternativeCompetence` can be used to construct a set of alternative sub-competences. It is possible to specify a minimum and a maximum number of alternatives that must be acquired (e.g., minimum k out of n). "Exactly" k sub-competences might be specified by setting both minimum and maximum to the same number. By default minimum is set to 1 so at least one subcompetence of the set is required.

Such a model allows for the representation of atomic competences, (un)ordered aggregation (all sub-competences must be acquired), alternative composition (a subset of sub-competences must be acquired) and any combination of all of them.

It is important to notice that if a competence is composed from several sub-competences, the proficiency level referenced in each subcompetence represents the minimum level required. For example, if it is required to have intermediate English skills in the context of science in order to be a good researcher, then anyone with advanced skills fulfills such a requirement. The subsumption relationship modeled within the proficiency levels is used for this purpose, and the proficiency level on the competence itself needs not to include all possible subsumers.

Our model is open to the addition of new relationships, among them, an equivalence relationship. This is especially interesting if competence repositories of two communities are joined and mappings between overlapping competences have to be modeled.

6.2.4 Competence Profiles

Previous sections described how competences, and relationships among them, can be modeled. In real world applications, competence definitions are used to support different tasks such as creating job profiles for hiring or selecting people for a particular project; creating personal competence profiles showing the abilities of a person; and modeling the prerequisites and expected results of joining a learning or training program. These tasks require modeling collections of required or acquired competences. Furthermore, the requirements specified by a job offer must be matched by the acquired competences an applicant provides. It therefore indicates that the model should be similar for all the cases enumerated in order to ease its matching. This model will be referred to as "competence profile" hereafter.

Two types of competence profiles can be distinguished, depending on their purpose:

Required Competence Profile: Specifies the requirements (in terms of competences) to be fulfilled by an applicant. These are typically used for job descriptions or program prerequisites[8].

Acquired Competence Profile: Specifies the accomplishments (in terms of competences) of employees and learners. These are typically used in order to show (and possibly prove) which competences have been acquired or to represent the expected accomplishment after successful completion of a program.

Each kind of profile is composed of a set of *ProfileElements*[9]. These profile elements may be *required* or *acquired*, depending on the type of the profile container (see figure 6.7). A profile element contains data which

- may be part of the criteria a company or a learning program uses to decide whether an applicant is appropriate

[8] Although this section focuses on gap analysis for a company or a learning institution, the same analysis can be used as a tool to help a learner find appropriate learning resources according to her goals and current knowledge.

[9] For clarity, the term *evidence* introduced in [70] was not kept. A *ProfileElement* represents a requirement or a statement of an acquired competence but not necessarily a proof. Therefore, evidence could be misleading since it may be confused with proof or certification

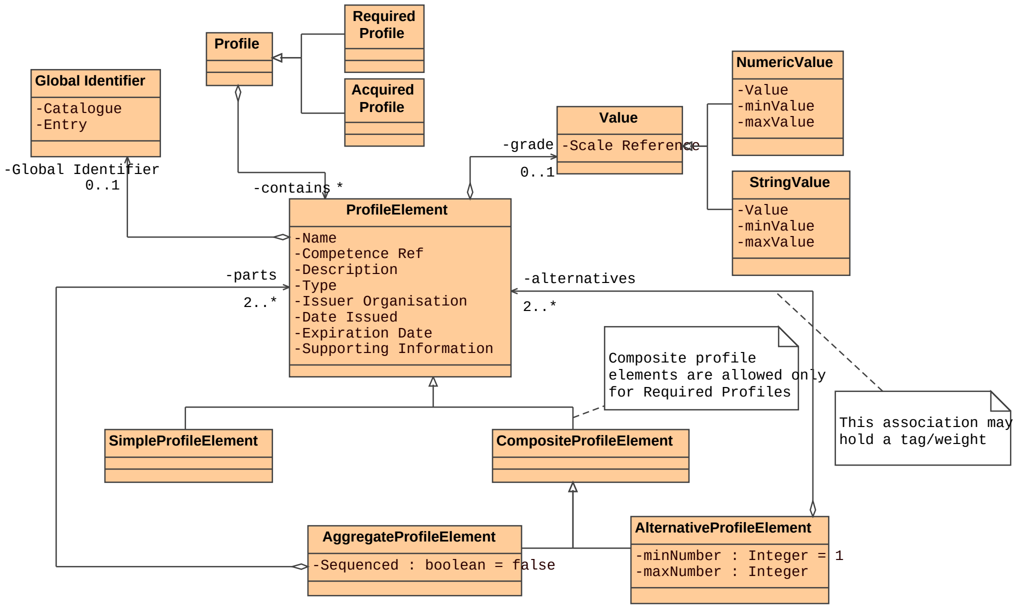


Figure 6.7: Competence Profile

- an institution providing degrees or certifications issues to learners as a prove of the acquired competence
- a learner uses to describe acquired competences in her CV (not necessarily with a proof or certification, e.g.,based on her experience)

Such information includes a type (e.g., driving license or university degree), the competence required or acquired and (possibly) a grade[10], the issuer organization, issue date and expiration date (i.e., from when the driving license is not valid anymore). All these fields are optional since not all are always needed. Typically, requirement profiles do not need to specify all fields of expected profile elements, but only part of them. In these cases, some fields may be left empty, ensuring comparison only on those fields which specify constraints. For example, being expert computer scientist may be a requirement but it may not be relevant where the competence was acquired (only competence field is filled in) or any applicant with a master degree may be sought but it does not matter in which field (only "type" is filled in and competence is left empty). In contrary, acquired profile elements should typically be filled in to a larger extent, specially if provided by certifications.

Note that the structure of a "ProfileElement" is different for required and acquired profiles. On the one hand, required profiles need to represent

[10] Note that grade and proficiency level represent different concepts (see section 6.2.4)

mandatory (English and French) and alternative requirements (either English or French) or even desired requirements (English mandatory and French is a plus). For that, the same composite model (meta-model) as the one specified for competences in section 6.2.3 (with the addition of tagging relationships with e.g. 'desired') is used, thus easing understanding and simplifying the tools needed to process these models. On the other hand, acquired profiles do not need such complex relationships and will therefore be represented as sets, that is, a flat collection of “SimpleProfileElement” elements.

Competence Proficiency Level vs. Grade in Competence Profile Element

Proficiency levels are part of competences, for example “Fluent English”. This is different from grades provided by institutions (e.g., 250 in TOEFL test). While the former represents that “any person who has such a competence is supposed to perform effectively”, the latter provides a “way to rate persons having such competence at a specific level of proficiency, by means of some sort of assessment”. For example, two people having successfully completed an “Advanced Oracle Database Administrator” program are able to perform effectively. However, they may have different grades in their final certification, which may be considered by HR departments before accepting any of them. In other words, proficiency levels (which are not bound to specific profiles) represent the scope of the competence acquired (advanced database administration vs. basic database administration) independently of whether a specific learner or employee (bound to a profile) learned the content perfectly or sufficiently to acquire the competence (higher or lower grade). For instance, being a proficient computer scientist requires to have advanced knowledge on databases, be intermediate software engineer and have basic knowledge on economics. Those represent the content (scope) required to acquire the competence, independently of the grade received by learners.

6.2.5 Motivating Scenario (revisited)

It is assumed the existence of repositories with information about skills, proficiency levels, context and competences as depicted in figure 6.8. This work does not deal with the problem of ontology heterogeneity and therefore assumes that there either exist appropriate standards for this information or there are available mappings between different ontologies (see e.g. [122, 34] and section 7). In addition, how these models are instantiated is also out of the scope of this paper. It is also assumed the existence of appropriate tools

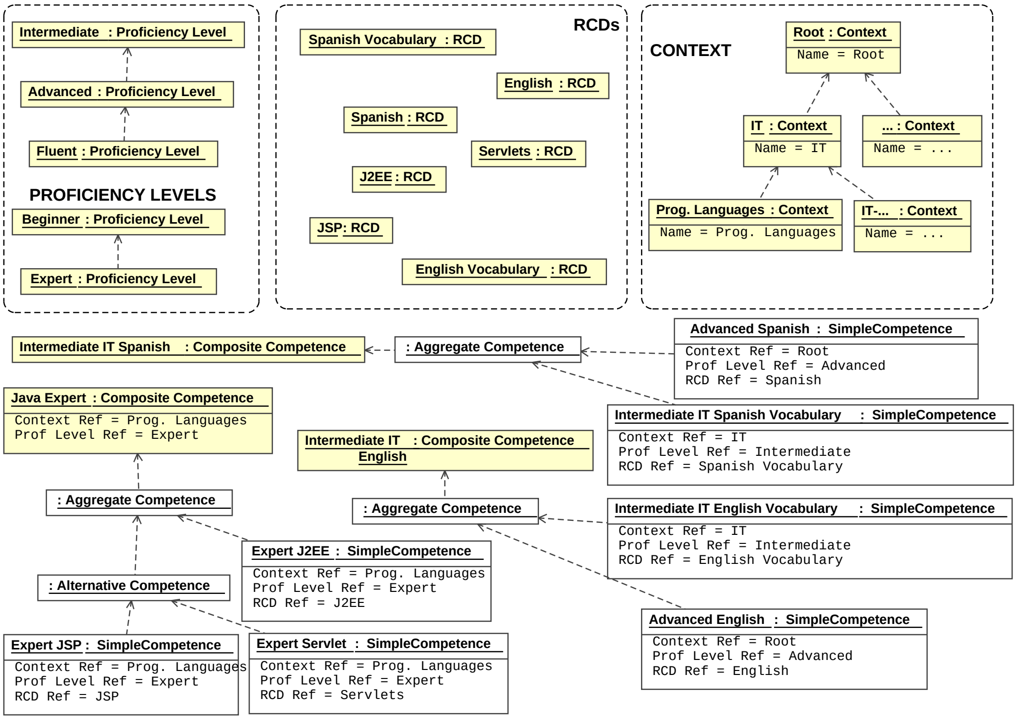

Intermediate : Proficiency Level
Advanced : Proficiency Level
Fluent : Proficiency Level
PROFICIENCY LEVELS
Beginner : Proficiency Level
Expert : Proficiency Level
RCDs
Spanish Vocabulary : RCD
English : RCD
Spanish : RCD
Servlets : RCD
J2EE : RCD
JSP: RCD
English Vocabulary : RCD
CONTEXT
Root : Context
Name = Root
IT : Context
Name = IT
... : Context
Name = ...
Prog. Languages : Context
Name = Prog. Languages
IT-... : Context
Name = ...
Intermediate IT Spanish : Composite Competence
: Aggregate Competence
Advanced Spanish : SimpleCompetence
Context Ref = Root
Prof Level Ref = Advanced
RCD Ref = Spanish
Java Expert : Composite Competence
Context Ref = Prog. Languages
Prof Level Ref = Expert
Intermediate IT Composite Competence English
Intermediate IT Spanish Vocabulary : SimpleCompetence
Context Ref = IT
Prof Level Ref = Intermediate
RCD Ref = Spanish Vocabulary
: Aggregate Competence
: Aggregate Competence
Intermediate IT English Vocabulary : SimpleCompetence
Context Ref = IT
Prof Level Ref = Intermediate
RCD Ref = English Vocabulary
Expert J2EE : SimpleCompetence
Context Ref = Prog. Languages
Prof Level Ref = Expert
RCD Ref = J2EE
: Alternative Competence
Advanced English : SimpleCompetence
Context Ref = Root
Prof Level Ref = Advanced
RCD Ref = English
Expert JSP : SimpleCompetence
Context Ref = Prog. Languages
Prof Level Ref = Expert
RCD Ref = JSP
Expert Servlet : SimpleCompetence
Context Ref = Prog. Languages
Prof Level Ref = Expert
RCD Ref = Servlets

Figure 6.8: Competence Profile Example

to hide the model from end users (e.g., competence management profile or CV creation).

Typically, a recruiter in an HR department would write a job offer like

> Wanted: J2EE consultant
>
> - Completed Master's Degree (any faculty)
> - Expert Knowledge in Java J2EE, Servlets, JSP
> - Very good English and/or French

An alternative would be to use the model proposed here, to encode the job advertisement (see left side of figure 6.9). The model not only enforces a well-structured profiling, it also saves the information in a machine-readable and machine-understandable way. The recruiter can as well reuse information created from previous job advertisements (e.g., reuse the definition of Java Expert for her company, as well as use the well-accepted definition of Master). This "indexable" representation also has significant advantages compared to the manual approach for the applicants: the applicants can now quickly seek on the advertisements, filter out advertisements for which their

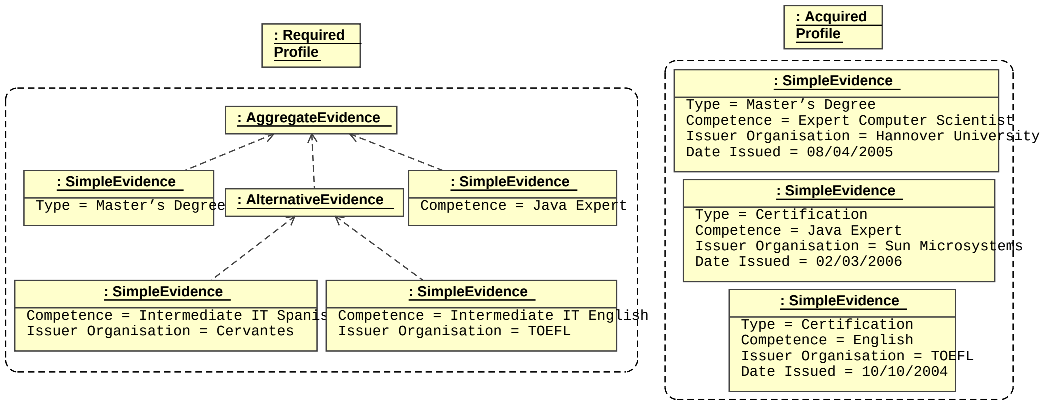


Figure 6.9: Personal Profiles Example

profile does not satisfy the requirements. In an even more advanced scenario, the profile representation can enable some ranking of the advertisements for which the applicant satisfies the requirements and some of the *optional* competences. Finally, the cycle is concluded when the applications come back to the recruiter. The recruiter can use a (semi-)automatic matching engine to filter the non-satisfactory applicants according to their profiles, and rate the suitable applicants. For example, an applicant profile as depicted in the right side of figure 6.9 would be a perfect match for such an offer. More complex techniques could be used for partial matches and rankings/ratings, as they have been hinted along this paper or in [29]. However, elaborating on the matching techniques themselves is out of the scope of this work.

Chapter 7

Semantic Integration

In previous sections, some of the basics for interoperability, namely common protocol and interfaces and a common query language (or the use of appropriate wrappers) have been presented. Although these elements ensure that two systems are able to talk to each other, they still do not guarantee that they will be able to understand each other unless they both understand the same vocabulary (they use the same schemas/ontologies). Even if a common schema exists, it still may be needed to convert from the local schema used and such a common schema (they are typically different).

Nowadays, there is a big effort on standardization of domain ontologies. For example, Dublin Core [33] is intended to be a standard for cross-domain information resource description and LOM [93] describes attributes required to fully/adequately describe a Learning Object. Unfortunately, still many proprietary schemas are used in each domain (e.g., database schemas within companies). For example, Dublin Core suggests using the attribute “creator” to describe the responsible of making or writting a resource. While many repositories probably follow this suggestion when annotating their resources, others might use, for instance, their own attribute “author” instead. In order to bring interoperability among them, a translation, that is, a data integration approach in the form of semantic mappings [100, 1] is needed. In this context, a semantic mapping is a transformation from one data model to another data model according to a set of rules (mappings).

This chapter defines three kinds of mappings and describes how they can be applied using a query rewriting mechanism performed by a mediator component. Such a mediator component is able to make transformations from one data model to another by rewriting queries. It would, according to a table of mapping rules, convert a query made with attributes from one data model to another query semantically equivalent only with attributes that may be found in the second data model.

The main contribution of this chapter is the presentation of three types of (datalog) mappings and its application by means of query rewriting mechanisms. These mappings allows the connection of repositories at the schema level, given a simple set of mapping rules. An opensource version of a mediator using such techniques has been implemented and is currently used by many repositories therefore demostrating its need and success.

7.1 Mediator Module

In order to provide semantic interoperability in our network, a module which transforms a query q_1 into a query q_2 according to a set of specified mappings has been developed [110, 15]. This module is intended to work on pairs of mappings without a unified schema, or in GAV or LAV integration approaches (see section 4.3.4).

QEL, the language used in our network, is based on datalog. In addition to standard datalog constructs, QEL includes some built-in predicates. Taking into account that in our network only metadata (in RDF) is queried and exchanged, the most important one is

$$qel : s(Subject, Predicate, Object)$$

which according to the QEL specification [104] "is true if Subject and Predicate are anonymous or non-anonymous RDF resources, and Object is a non-anonymous or anonymous RDF resource or an RDF Literal and the triple Resource Predicate Object exists in the RDF data". For example, a query like

$$? - qel : s(X, dc : title, 'Artificial\ Intelligence').$$

will return all the resources which title is "Artificial Intelligence". Other useful built-in predicates are *qel:like(X,Y)* ("used to determine whether an RDF literal or URI contains a string as a substring"), *qel:lessThan(X,Y)* and *qel:greaterThan(X,Y)* which are used to compare two RDF literals.

Given this short introduction to the language (more information in section 4.3.2), the following query that will be used for the examples in the rest of this section is presented:

$@prefix\ qel : < http : //www.edutella.org/qel\# > .$
$@prefix\ dc : < http : //purl.org/dc/elements/1.1/ > .$
$@prefix\ lom : < http : //ltsc.ieee.org/2002/09/lom\text{-}rights\# > .$
$? - qel : s(X, dc : title, Title),$

$$\begin{aligned}&qel : s(X, dc : description, Description),\\&qel : s(X, dc : creator, Creator),\\&qel : s(X, lom : cost, Cost),\\&qel : s(X, dc : subject, Subject).\end{aligned}$$

This query retrieves all the resources with title, description, creator and subject attributes from Dublin Core and the cost from LOM. The first lines of the query with prefix "@" define the namespaces.

Given such a query, the following requirements were identified:

- The query specifies a property (property and attribute are used indistinctly in this document) that does not exist in the source but the source has an equivalent property which could be used instead. For example, if one data source has a schema which uses the property "abstract" instead of the property "description" from the Dublin Core standard.

- The query specifies a property and one value according to a specific taxonomy and the source uses a different taxonomy (possibly also a different property). For example, if the query searches for resources with "dc:subject" following the ACM classification [3] and the data source does have "dc:subject" but it follows the Dutch Basic Classification [42].

- In general, if one of the attributes is not available at the data source, the whole query fails[1]. However, it might happen that although the source does not have explicitly such an attribute, all its resources would share the same value if it existed. For example, assume a repository where all the resources are offered for free. This repository does not have the property "lom:cost" because it is not needed. However, in case one query contains this attribute, the whole query would fail (even if the constraint in the query is "lom:cost = No" which is actually true though it is not annotated). In such a case, it is desirable to assign a default value to all the resources in the data source without having to explicitly annotate all the resources of the repository.

In order to satisfy these requirements a module that performs two types of mappings and one extra transformation was developed: property mapping, property-value mapping and default value transformation (see table 7.1 for the whole list of mappings and [108] for technical details).

[1] Here it is assumed that only conjunctives queries are sent. QEL support disjunctive queries but they will be omitted here because of simplicity.

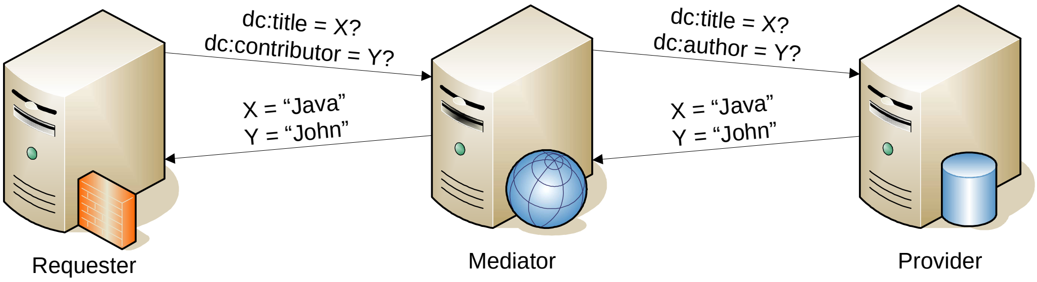


Figure 7.1: Property Mapping Example

7.1.1 Property Mapping

A property mapping specifies how one property in the query must be reformulated. When the mapping module receives a query that contains the triple $qel : s(X, p_1, Z)$ it rewrites it into $qel : s(X, p_2, Z)$.

Using our example query and taking into account the requirement in which the source does not contain the property "dc:description" but "own:abstract" (where "own" stands for the namespace of the local schema), it is possible to define the following mapping[2]:

$$(X, dc : description, Z) \leftarrow (X, own : abstract, Z)$$

Another example would be

$$(X, dc : contributor, Z) \leftarrow (X, dc : creator, Z)$$

which returns any contributor of a resource as its creator (see figure 7.1).

This mapping is currently a 1-to-1 mapping, that is, there is only one triple at each side of the mapping (separated by the left arrow) but it is also possible to specify 1-to-2, 2-to-1 and 2-to-2 mappings (see table 7.1). For example, suppose the author in the source is encoded using the property full name from the vcard ontology [150]. In such a case, the following mapping is needed

$$(X, dc : creator, Z) \leftarrow (X, dc : creator, Y), (Y, vcard : fn, Z)$$

in order to abstract from the internal representation at the source.

[2] Note that similar notation to the one in [148] is used, where the left side of the rule is considered a view defining new semantics of the properties available in the local schema (on the right side). Therefore, the right side of the mapping rule is rewritten into the left side.

Mapping type	Description
1-to-1 property mapping	$(R, p_1, O) \leftarrow (R, p_2, O)$
1-to-1 property-value mapping	$(R, p_1, v_1) \leftarrow (R, p_2, v_2)$
2-to-1 property mapping	$(R, p_1, O), (O, p_2, L) \leftarrow (R, p_3, L)$
2-to-1 property-value mapping	$(R, p_1, O), (O, p_2, v_1) \leftarrow (R, p_3, v_2)$
1-to-2 property mapping	$(R, p_1, L) \leftarrow (R, p_2, O), (O, p_3, L)$
1-to-2 property-value mapping	$(R, p_1, v_1) \leftarrow (R, p_2, O), (O, p_3, v_2)$
2-to-2 property mapping	$(R, p_1, O), (O, p_2, L) \leftarrow (R, p_3, O), (O, p_4, L)$
2-to-2 property-value mapping	$(R, p_1, O), (O, p_2, v_1) \leftarrow (R, p_3, O), (O, p_4, v_2)$
Default value	$(p \leftarrow v)$

Table 7.1: Types of Mappings

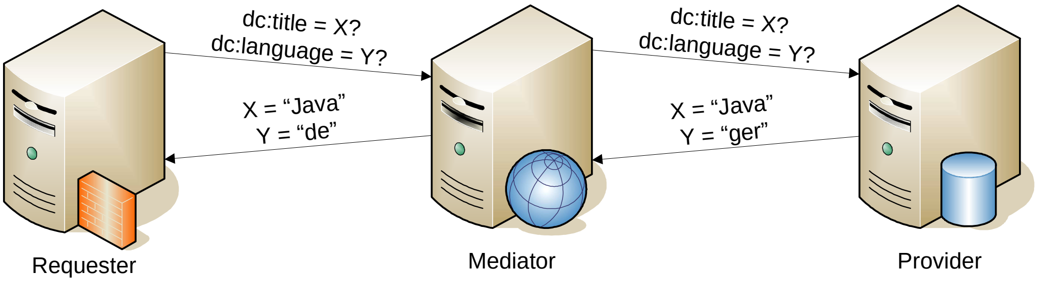


Figure 7.2: Property-Value Mapping Example

7.1.2 Property-Value Mapping

The mapping described above assumes that one property is completely mapped onto another one. However, mapping can be brought to the granularity of values. A property-value mapping applies only when a query contains not only a specific property, but also a specific value for that property and then both of them map into other (possibly the same) property and value. For instance, assume that our example query uses the ACM classification in the property “dc:subject” and our source does have the property “dc:subject” but annotated with the Dutch Basic Classification taxonomy. Several mappings of the form

$$(X, dc : subject,' Software/Programming\_Languages') \leftarrow \\ (X, dc : subject,' Computer\_Science/Programming\_Languages')$$

could be used to specify how the different values from the ACM taxonomy map into the Dutch Basic Classification.

Another example (see figure 7.2) is

$$(X, dc : language,' ger') \leftarrow (X, dc : language,' de')$$

which transforms attributes withe the language of resources encoded with the standard ISO 639-1 (alpha-2 codes) onto ISO 639-2 (alpha-3 codes).

In the same way as the property mapping, it is possible to extend this 1-to-1 to 2-to-1, 1-to-2 and 2-to-2 mappings (see appendix C for examples of mappings used to expose university courses on a network which requires a different schema).

7.1.3 Default Value

Property and property-value mappings provide rules which define how source triples are reformulated into another equivalent triples corresponding to the destination schema.

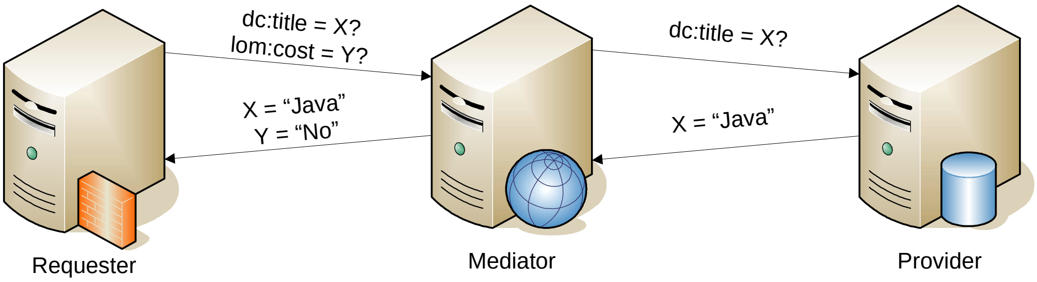


Figure 7.3: Default Value Example

The "default value" mapping works differently. The properties specified in default values do not exist in the source repository and therefore they must be removed (not just reformulated) in the new query. This process[3] is formalized in Appendix B.

Following this approach, when a query is received by our mapping module, if there exists in the query any occurrence of a property specified in the default values, this occurrence is temporarily removed. This way, the query is sent to the local repository without that property (otherwise the query would fail) and a result set is returned. However, this result set still does not contain the default values that were requested (the properties previously removed) and therefore they must be added. Default values are added to each of the rows in the result set returned by the repository. For example, using our example query, suppose that our source repository does not have the property "lom:cost" but all the resources in the repository are free of charge. The following default value can then be defined

$$(lom : cost \leftarrow' No')$$

This way, any triple in the query referring to the property "lom:cost" would be removed before the query is sent to the repository and added subsequently to the returned result set together with the default value "No" (see figure 7.3). In contrary, the query may specify that only elements which are not free of charge should be returned. In such a case, since it does not match the default value, the query is not executed and an empty result set is returned.

[3] This process is similar to a view in a database which specify constants in its definition.

Chapter 8

Ranking

There are three different types of queries on the Web according to their intent [14]: *navigational* (to get a particular site), *informational* (to retrieve information assumed to be present on one or more web pages) and *transactional* (to perform some web-mediated activity). The same may be extended to any distributed environment in which resources can be searched, although in distributed networks like P2P or business coalitions typically informational and transactional queries are mostly used. For these two kind of queries (informational and transactional) the amount of resources (size of the corpus) may be very large and diverse and many resources could be related to a given query. For this reason a method to sort all resources potentially relevant to a user query is needed.

However, ranking resources that are distributed over a network is not an easy task. Different solutions exist depending on whether there exists any kind of relationships among resources of different providers [114, 83, 80] (either explicit links like in the Web or made via user consumption or recommendation) or whether there is an overlap in the coverage of resources and search engines [43, 86, 50, 49] (like in meta search engines). However, in a network coalition where resources are distributed over the providers, they are typically unique within each provider and do not link to other providers' resources, therefore making ranking an even harder challenge.

This chapter provides two different ranking algorithms based on the two different scenarios described above. For the case where relationships are available among resources in different providers, an algorithm focused on personalization is presented. For the case where no relationships exist, an algorithm for semantic ranking integrated with some personalization based on user profiles is presented. It is important to note that both algorithms are complementary and could therefore be used together.

The main contribution of this chapter is the specification of two ranking algorithms: one to provide personalized results to the user assuming the existence of relationships among resources and another to adequate for unlinked corpus. The former retrieves information from the user in order to find out her preferences and interests and to provide a personalized ranking. The latter provides a weighted mechanism to rank results retrieved from heterogeneous sources based on metadata attributes, ontologies and user profiles.

8.1 PROS: A Personalized Ranking Platform

To solve the problem of too many results being returned to a user, a new concept associated with every resource is introduced: importance. For each web page[1] a value will be associated measuring its importance. Built on top of the text of the web pages the hypertext (link structure and link text) provides an important way to retrieve information. Based on that information it is possible to build such a scoring function. This way, once the search engine retrieves the results that match the user query, it will order them according to their importance score presenting first the most important (supposed to be likely more interesting to the user).

Some of current web search engines apply basic personalization techniques such as personalized user interfaces based on a user's physical location (e.g., a user interface in Spanish if a user accesses from a Spanish IP address). However, this refers only to how the search engine communicates with the user (user interface) but it does not take into account user interests during search and results ranking. It applies the same process and ranking independently of the user who submits the query without taking into account user interests. This information can be extracted from bookmarks and from frequently visited pages. This information, and global ranks, helps our framework to create a personalized view of these global ranks according to user preferences.

This section describes the design and implementation of PROS [109, 25], a personalized ranking platform which uses the algorithm presented in [78] (called the *"PPR algorithm"* – Personalized PageRank – hereafter) as well as new algorithms for automated input generation to drive and optimize it. Our platform is based on HubFinder [24, 23], an algorithm developed to find related pages (or hubs, depending on the user) and on a proxy server meant to (temporarily) capture user's surfing behavior. Hubs in this context are Web pages pointing to many other important pages (i.e., with a high rank). Their counterpart are authorities, which are high quality pages pointed by

[1] These algorithms are typically applied to the Web but they can be extended to any network where relationships among resources are available. Any reference to *the Web* here should be understood as such.

many hubs.

In the original paper [78], PPR user profiles, used as input for building personalized ranks, are gained by presenting users a set of pages/hubs with high PageRank (as computed using PageRank [114]) from which they can choose a set of preferred pages. The disadvantage of this procedure is that this operation takes time and might often be superfluous as most Internet users have some bookmarks of their own already, which could be used to derive their user profile. We therefore wanted to build such a preference set *automatically*, using user's bookmarks and/or most surfed pages (i.e., pages read for a longer period of time, or voted for by a user). This resulting set can then be extended using an algorithm which finds high quality related pages.

The contributions of PROS are:

1. a platform which automates the computation of personalized ranks by generating more comprehensive input data with less user effort, and which consists of two modules: one based on user's bookmarks and the other based on the output of a specialized proxy server which computes the pages most likely to be considered interesting by the user.

2. both modules use HubFinder (a fast and flexible algorithm for finding related pages using the link structure of the World Wide Web) and HubRank (a modified PageRank algorithm which combines the authority value with the hub value of Web pages, in order to further extend these sets of Web pages into the input data needed by the PPR algorithm).

3. first experimental results from integrating PROS into a personalized Web search system.

8.1.1 Notation

Let $G = (V, E)$ denote the *web graph*, where V is the set of all web pages and E contains a directed edge $< p, q >$ iff page p links to page q. For a page p, $I(p)$ denotes the set of in-neighbors (pages pointing to p) and $O(p)$ the set of out-neighbors (pages pointed by p). Individual in-neighbors are denoted as $I_i(p)$ $(1 \leq i \leq |I(p|)$, and individual out-neighbors are denoted analogously [78]. For convenience, pages are numbered from 1 to n, and it is possible to refer to a page p and its associated number i interchangeably. For a vector $\mathbf{v}$, $v(p)$ denotes *entry* p, the *p-th* component of $\mathbf{v}$. Vectors will be always typeset in boldface and scalars (e.g., $v(p)$) in normal font. All vectors are $n-$dimensional and have nonnegative entries. They should be thought of as distributions rather than arrows.

Let A be the matrix corresponding to the web graph G, where:

- $A_{ij} = \frac{1}{|O(j)|}$ for PageRank and HubRank
- $A_{ij} = 1$ for HITS

if page j links to page i, and $A_{ij} = 0$ otherwise. The difference between the algorithms is that for PageRank, the matrix should be normalized in order to make the algorithm converge. In the other algorithms, a different normalization is done if necessary.

8.1.2 Overview

As explained before, personalized rankings can improve current Web search systems by adapting results to user preferences. The algorithm presented in [78] is the most recent step in this direction. An open issue is how a set of highly rated hubs, needed as input for the adaptation process, is selected by the user. The personalization (and therefore success) relies on the user's ability to choose such high quality hubs which match her preferences.

This section describes how to exploit information collected from the user to derive the highly rated hubs that represent the user profile. The computation is performed automatically based on the following input:

Most surfed pages. Pages visited by the user are tracked using a specialized proxy we implemented. The proxy records information about the duration of time the user looked at a page and how frequently she returned to it.

User's bookmarks. Additionally, the user's bookmarks are used as an indication for user preferences. Currently, bookmarks are directly provided by the user, but this interaction could also be automated (e.g., using a browser plug-in).

These two sets of pages represent the user interests but it is usually too specific. The PROS platform consists of two main modules, which use the two input sets described above. They use HubFinder and HubRank, two algorithms developed for finding related pages using the Web link structure and for ranking Web pages, respectively.

HubFinder finds the related hubs to a given set of pages. HubFinder is applied to the most frequently surfed pages to obtain the hubs related to them. Using the same mechanism, it finds the related hubs to the user's bookmarks. Therefore, HubFinder finds two sets of highly rated hubs which represent a generalized view of the specific user interests (frequent surfed pages and bookmarks respectively). This information, together with a merge of the original bookmarks and surfed pages, is combined to build the *User' Hub Set* which represent the *User's Web Profile*. This *User's Web Profile* is used by the search engine to build a personalized rank for each user. The whole process is depicted in figure 8.1.

The first module consists of applying the following operations:

1. Get bookmarks from the user.

2. Add bookmarks to the *preference set*.

3. Apply HubFinder, using the user's bookmarks as input and HubRank scores as trimming criterion. HubRank is the best criterion in this situation, because the PPR algorithm needs hubs with high PageRank as input and HubRank has been designed as a biasing of PageRank towards hubs, as discussed later in this section.

4. Add the preference set and the output from the previous step to the *hub set*.

The second module is based on using a proxy server for a limited period of time in order to capture user's "surfing behavior". Its modus operandi is described below:

1. The user surfs the Web using a given proxy. The proxy will output the pages examined by the user for a certain period of time (there must be both a lower threshold and an upper one to avoid the situation when the user leaves the browser open for a long period of time without using it), as well as those most frequently revisited. The more time it is used, the better ranking accuracy will be acquired.

2. Add the user's most surfed pages (as recorded by the proxy) to the *preference set*.

3. Apply HubFinder with HubRank as criterion and a small radius and number of output pages. We want the pages related to user's bookmarks to be more important than the pages related to his/her most surfed ones and using a smaller radius is a way to achieve this.

4. Add user's most surfed pages, as well as the pages related to them to the *hub set*.

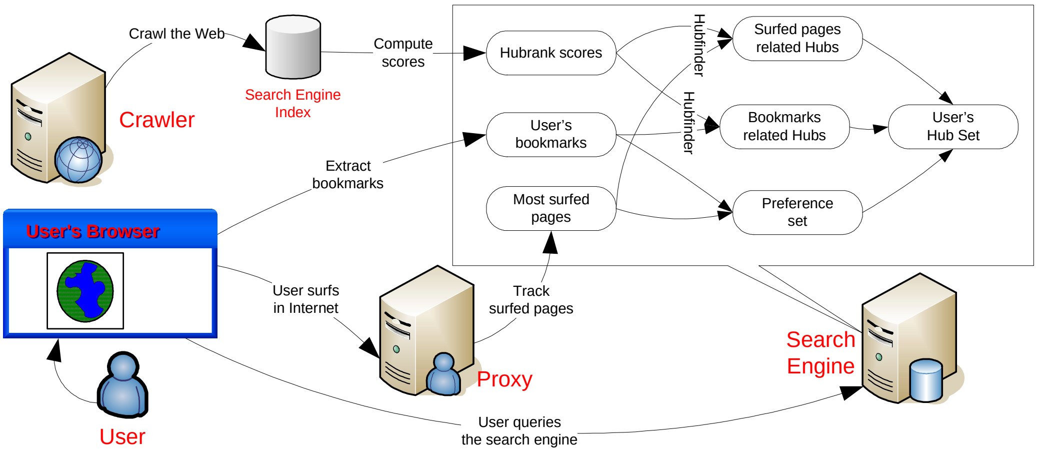


Figure 8.1: Personalized Ranking Platform

Finally, the PPR algorithm is executed using the newly computed preference and hub sets. The complete process is depicted in figure 8.1.

8.1.3 HubRank

PageRank has demonstrated to be a successful ranking algorithm although it focuses on authority values exclusively. On the other hand, HITS computes hub ranks according only to hub values. However, if a user searches for "travel agency", she would probably be interested in a high-quality hub page with a list of all the travel agencies, result which can not be provided by either one of these algorithms (PageRank output would not be a hub and HITS hub output would not consider authority value). HubRank addresses this problem by combining both approaches into a single score which biases PageRank [114] towards hubs.

We started from the idea that a page pointing to a good hub is a candidate for having a high hub rank as well. Often we encounter pages (perhaps good authorities) with only a few out-going links, but towards very important hubs. We consider such pages more important than the hubs themselves, the reason being that while a hub can cover lots of topics, such a page will usually contain information about the content addressed by the hubs it is pointing to, about the value of their content (e.g., author opinions), etc.

To compute these hub scores, the PageRank personalization vector was modified in order to consider the out-degree of the pages. Intuitively, the random surfer when bored prefers pages with a big out-degree. This way, the global importance of the pages will play an important role in defining general scores, as the random surfer will follow the out-going links with a higher probability than the random ones, and on the other hand, the out-degree of pages will always be considered. In PageRank, given a Web page p, the following formula is used:

$$PR(p) = (1 - c) \sum_{q \in O_p} \frac{PR(q)}{\|O(q)\|} + cE(p)$$

where the dumping factor $c < 1$ (usually 0.15) is necessary to guarantee convergence (A is not irreducible, i.e., G is not strongly connected) and to limit the effect of rank sinks [12]. Intuitively, a random surfer will follow an outgoing link from the current page with probability $(1 - c)$ and will get bored and select a different page with probability c. In addition, the vector $\mathbf{E}$ is a uniform distribution with $\frac{1}{NP}$ in each entry (where NP is the total number of pages).

To achieve these hub scores, we modify the PageRank personalization vector ($\mathbf{E}$) to consider the out-degree/in-degree of the pages. More intuitively, the random surfer will always prefer pages with a big out-degree when she gets bored. This way, the global importance of the pages will play an important role in defining general scores, as the random surfer will follow the out-going links with a higher probability than the random ones, and on the other hand, the out-degree of pages will always be considered. We set the value of each entry i of the vector $\mathbf{E}$ to $E_i = |O(i)|\frac{NP}{|O|}$ where $|O|$ is the summation of the out-going links over the whole Web graph and $O(i)$ the set of out-links of page i. Appendix D provides a brief overview of the benefits of the algorithm as well as experimental results.

Analogously, authority scores can be computed setting the components of the personalization vector to $E_i = |I(i)|\frac{NP}{|I|}$, where $|I|$ is the summation of all the in-degrees of the pages in the Web. However, when computing authority values, one might use a different matrix than the one used in PageRank (the row-out-going links matrix of the Web graph normalized on columns), depending on how much importance needs to be given to hub values and how much to authority values (e.g., the transposed row-out-going links matrix normalized on rows, as in [103]).

8.1.4 HubFinder

HubFinder is an algorithm for finding hubs, related to an initial base set of Web pages. *Related* is defined similarly to [80], i.e. using only link information as input. Two pages are related if one is accessible from the other via the link structure of the Web graph (following either in-going or out-going links). We should also add that the distance (the number of links followed)

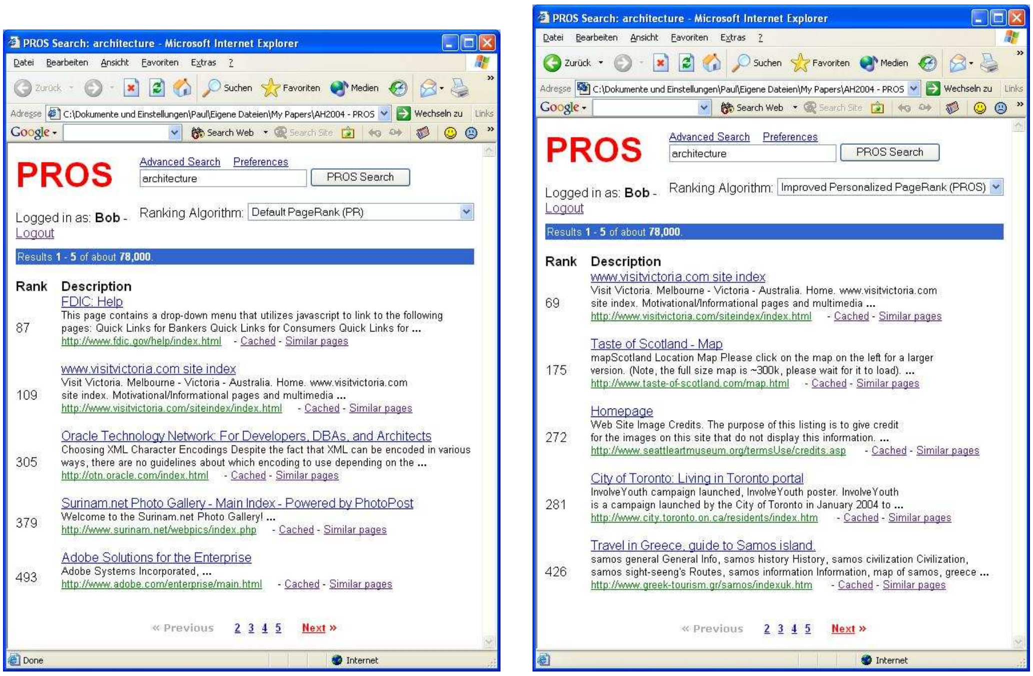
```
Let $\Gamma$ be the Base Set of pages whose related hubs we are looking for
$\Gamma \leftarrow$ Apply the Kleinberg Extension on $\Gamma$ once
$\Gamma' \leftarrow \Gamma$
For $i = 1$ to $\sigma$ do:
    $\Gamma'' \leftarrow$ Apply the Kleinberg Extension on $\Gamma'$ once
    Trim $\Gamma''$ to contain only *interesting* pages, *not* contained in $\Gamma$
    $\Gamma \leftarrow \Gamma + \Gamma''$
    $\Gamma' \leftarrow \Gamma''$
End For
Trim $\Gamma$ to contain as many interesting pages as desired
Return $\Gamma$
```

Figure 8.2: HubFinder pseudo-code

between such two pages is usually less than 6 (according to our experiments, in cases where the distance is bigger, the link information becomes insufficient to say that pages are similar in context with a high enough probability), and thus the related hubs are in the vicinity of the starting page. The maximum distance (noted σ and also called radius) is a parameter for HubFinder.

In order to get a good set of related pages the following aspects were taken into account: the set has to be small, rich in relevant pages and it should contain many of the strongest authorities. [83] extracts the top results of a query sent to a search engine and builds a focused sub-graph of the WWW around them. It then extends this base set by adding all pages these results point to and at most d pages pointing to each of such results. This operation is called *Kleinberg extension*. The author extends the initial set only once, and focuses on computing Hub and Authority scores, whereas we were focusing on finding related pages or hubs. Therefore we iteratively apply the Kleinberg extension several times on the resulting set of each previous iteration in order to obtain more pages and thus more representative results. As this scenario leads to very big output sets (up to 500,000 pages), trimming is necessary after each intermediate step. The pseudo-code of the HubFinder algorithm is described in figure 8.2.

Two aspects have to be considered: how many pages should be kept after each iteration and which are the *interesting pages*? Regarding the former question, one percent of the current set size is kept, whereas the best crite-



Adresse: C:\Dokumente und Einstellungen\Paul\Eigene Dateien\My Papers\AH2004 - PROS

architecture

Logged in as: **Bob** - Logout

Ranking Algorithm: Default PageRank (PR)

Results **1** - **5** of about **78,000**.

Rank	Description
87	FDIC: Help This page contains a drop-down menu that utilizes javascript to link to the following pages: Quick Links for Bankers Quick Links for Consumers Quick Links for **...** http://www.fdic.gov/help/index.html - Cached - Similar pages
109	www.visitvictoria.com site index Visit Victoria. Melbourne - Victoria - Australia. Home. www.visitvictoria.com site index. Motivational/Informational pages and multimedia **...** http://www.visitvictoria.com/siteindex/index.html - Cached - Similar pages
305	Oracle Technology Network: For Developers, DBAs, and Architects Choosing XML Character Encodings Despite the fact that XML can be encoded in various ways, there are no guidelines about which encoding to use depending on the **...** http://otn.oracle.com/index.html - Cached - Similar pages
379	Surinam.net Photo Gallery - Main Index - Powered by PhotoPost Welcome to the Surinam.net Photo Gallery! **...** http://www.surinam.net/webpics/index.php - Cached - Similar pages
493	Adobe Solutions for the Enterprise Adobe Systems Incorporated, **...** http://www.adobe.com/enterprise/main.html - Cached - Similar pages

Adresse: C:\Dokumente und Einstellungen\Paul\Eigene Dateien\My Papers\AH2004 - PROS

architecture

Logged in as: **Bob** - Logout

Ranking Algorithm: Improved Personalized PageRank (PROS)

Results **1** - **5** of about **78,000**.

Rank	Description
69	www.visitvictoria.com site index Visit Victoria. Melbourne - Victoria - Australia. Home. www.visitvictoria.com site index. Motivational/Informational pages and multimedia **...** http://www.visitvictoria.com/siteindex/index.html - Cached - Similar pages
175	Taste of Scotland - Map mapScotland Location Map Please click on the map on the left for a larger version. (Note, the full size map is ~300k, please wait for it to load). **...** http://www.taste-of-scotland.com/map.html - Cached - Similar pages
272	Homepage Web Site Image Credits. The purpose of this listing is to give credit for the images on this site that do not display this information. **...** http://www.seattleartmuseum.org/termsUse/credits.asp - Cached - Similar pages
281	City of Toronto: Living in Toronto portal InvolveYouth campaign launched, InvolveYouth poster. InvolveYouth is a campaign launched by the City of Toronto in January 2004 to **...** http://www.city.toronto.on.ca/residents/index.htm - Cached - Similar pages
426	Travel in Greece, guide to Samos island. samos general General Info, samos history History, samos civilization Civilization, samos sight-seeng's Routes, samos information Information, map of samos, greece **...** http://www.greek-tourism.gr/samos/indexuk.htm - Cached - Similar pages

Figure 8.3: Prototype of the PROS Web Search System

rion tackling the latter issue are global rank scores (e.g., as computed with PageRank or a similar algorithm). [24, 23] contains an in-depth discussion of the algorithm, a formula on the exact number of pages to keep, as well as a proposed extension based on text analysis. Appendix E presents some experimental results.

8.1.5 Prototype

Current Web search systems apply only basic personalization techniques (e.g., presenting a user interface in Spanish if the access is from a Spanish IP address). However, this refers only to how the search engine interacts with the user, but it uses the same ranking process no matter who submits the query. To exemplify this problem, let us imagine that a user searches using the keyword "architecture". Output topics may vary from computer architecture to building architecture or even something else. By extracting user's interests from her bookmarks (if she likes building architecture she would have some bookmarks on it) and from her most visited pages (she would check building architecture pages often), we can create a personalized view of the global ranks, and thus provide tailored output for each user. A screenshot of our prototype can be seen in figure 8.3. As a comparison, the

Algorithm	Preference Set	Hub Set
PPR	30 user defined bookmarks.	User's bookmarks (30) plus top ranked PageRank pages. Totally about 1200 pages.
PROS	30 user defined bookmarks plus 78 pages selected tracking user's surfing behavior (108 pages in total).	The preference set plus its related pages plus top ranked PageRank pages. Totally about 1700 pages.

Table 8.1: Input Data for the PPR algorithm experiments

results obtained when ranking URLs with the PageRank algorithm [114] are presented on the left side, and with PROS on the right. Our tester was interested in building architecture. While with PageRank only two output URLs were relevant, all five generated by PROS were worth checking.

8.1.6 Experiments and Results

Tests on several small Web crawls (3 to 40 thousand pages) and on two bigger ones were performed, one with one million and one with three million Web pages. The results presented in this section use the largest set. Furthermore, PPR and PROS was run using several data sets as input and several users, but only the most representative experiments were selected to be described here.

Our first experiment follows all guidelines of [78]. It has 30 user bookmarks as preference set and a hub set mixing user's bookmarks with top PageRank documents. The second experiment uses the input obtained with our ranking platform. A tester surfed the Web for about two weeks using our proxy and 78 pages were selected as her "fingerprint". These were merged with her 30 bookmarks (same as in the first experiment) into the *preference set*. Then, HubFinder with HubRank as criterion was applied on both the set of bookmarks and the set of most surfed pages, obtaining about 900 pages from the former one and about 400 from the latter one (using a radius of 5 for the bookmarks and a radius of 2 for the most surfed pages). To these 1300 pages some top PageRank pages were added and the resulting set was used as *hub set*. A description of the input data used can be found in table 8.1. Our tester was an architect, having traveling and software as other hobbies, and sports as a secondary interest. Her bookmarks were distributed accordingly: 15 on architecture, 7 on traveling, 6 on software and 2 on sports.

To analyze the resulting ranks, some general keywords were selected (see

Query	**PageRank**			**PPR**			**PROS**		
Keywords	R.	P.R.	I.	R.	P.R.	I.	R.	P.R.	I.
architecture	5	3	2	3	7	0	8	2	0
building	3	2	5	2	3	5	4	1	5
Paris	6	0	4	2	3	5	6	2	2
park	6	0	4	8	0	2	10	0	0
surf	3	0	7	4	2	4	7	2	1
Total	23	5	22	19	15	16	35	7	8

R. ≡ Relevant, *P.R.* ≡ Partially Relevant, *I* ≡ Irrelevant

Table 8.2: Relevancy value for different search keywords and different algorithms

table 8.2) and Web searches performed, exactly as in a search engine. Results were sorted with respect to their ranks, without considering term frequency of keywords in output documents. The ranking algorithms used were PageRank, PPR, and PROS. Although the first one does not involve any personalization, it is was decided to implement it too, as it is the most popular algorithm and useful as background for our evaluation.

The top 10 URLs obtained by ranking the search results with each algorithm were classified into the following categories: (a) Relevant (denoted by "R." in table 8.2) if the URL was on one of the four topics of interest of our tester; (b) Partially Relevant ("P.R.") if it was on a topic related to one of the above-mentioned four ones (e.g., an URL on hardware architectures was considered partially related to computer software); or (c) Irrelevant ("I.") if it was not in any of the previous categories. A detailed list with all the output URLs can be found in [119].

The most important issue is that, as expected, the original PageRank algorithm provides top results on several topics, even though the searcher is almost always interested in only a specific one. This behavior is understandable, as the algorithm cannot disambiguate results based on user preferences.

The PPR algorithm performs only slightly better in this experiment (the total number of possibly relevant URLs is 34, whereas for PageRank it is 28), mostly because the input sets were too sparse and qualitatively not very good. This might be improved by adding additional top PageRank pages to the preference set, but this approach was not used, as it would have definitely damaged the personalization aspect (remember that top PageRank pages can be on any topic).

Finally, significant improvements are seen when using PROS. The number of relevant pages is much higher than for the other two algorithms. However, still some bad output URLs (e.g., for the search keyword "building") were received. We think this happened because of the general profile of our tester. Other tests performed with more focused testers (i.e., with a profile on exclusively one topic, such as "architecture") provided even better results, but we consider the experiment presented in this paper to be the closest to a real-life situation.

As we know from [78], pages from the preference set may have different importance. In our tests, all pages had the same weight, that is $\frac{1}{|PS|}$, where PS is the preference set. Let us denote by B the set of bookmarks and S the set of user's most surfed pages. In this case, we can give for example $\frac{2}{3*|B|}$ importance to bookmarks and $\frac{1}{3*|S|}$ to user's most surfed pages. We think this approach (or a more complex one which automatically gives an importance value to each of the most surfed pages, depending on the amount of surfing time or revisits) would provide even more accurate ranks. Further experiments are needed to define the correct biasing of these pages.

Generally, our experiments allow us to conclude that PROS increases the ranks of pages similar to user's bookmarks, as well as those that are most likely to be considered interesting by the user (and thus the granularity of relevant results when performing a Web search). If the tester uses the proxy server for longer periods, the accuracy of the latter set is proportionately bigger (i.e., the size of the "most surfed pages" set is bigger, and therefore the rankings are more accurate).

8.2 Ranking based on Semantics and User Profiles

Ranking resources in a centralized repository is a "simple" task because the whole corpus (set of documents) is known. Ranking resources that are distributed over a network is not an easy task. Different solutions exist depending on whether there exists any kind of links between resources of different providers [114, 83, 80] (either explicit links or made via user consumption or recommendation) or whether there is an overlap on the covering of resources and search engines [43, 86, 50, 49] (like in meta search engines). For example, in a distributed environment like the web, if it is assumes that we have relations among documents (e.g., hyperlinks on web pages), link analysis techniques like PageRank [114] or HITS [83] can be used. If there exist manually created relationships (e.g., user accessed or voted document A), similar techniques like the SimRank algorithm [80] (which are more like recommendation techniques in this case) can be applied once enough data is available. This may not be feasible in many scenarios (as the one described in section 9.1), where no such relationships exist.

As stated above, it is also possible to apply rank aggregation among different data-sources. That is, each data-source provides one ranking of the results which then can be combined into a global one [43, 86, 50, 49]. This is possible by doing it quantitatively (taking into account the exact score for the rank of each resource) or qualitative (taking into account only the ordering given by each data-source). The problem is that for these techniques an overlap between the different data-sources is required, that is, it is needed that some resources (not one or two but at least 20%) or their metadata are not only in one repository but in several of them. This is applicable to web search engines which have a big overlap on the amount of web pages they crawl but again this may not be a valid assumption (as in the scenario described in section 9.1).

Some networks have the properties that the different data-sources do not provide any links among repositories and that there is no overlap among resources on them (both of them are allowed but not necessarily both assumed). That is, in these networks resources are distributed over the providers, they are unique within each provider and do not link to other providers' resources. In addition, we assumed the existence of user profiles with information about goals, interests, the history of the user and preferred language among others. This ensures that we provide some basic level of personalization instead of basing our algorithm only on text measures. In such a case, the following criteria may be applied:

Occurrence and proximity. How many times the keyword occurs in the resource (at the metadata level) and how far are the different keywords (in case the query has several) among them in the text.

Provider reputation. Users could rank the different providers according to previous experiences. This way, at the beginning all the providers would have the same weight, but if any of them start to misbehave, a user could apply a lower weight to all the results coming from it.

Ontology Classification. Some properties (e.g., dc:subject) may point to a taxonomy/hierarchy specifying the classification of a resource. It is possible to calculate the distance between the elements of the hierarchy the user is interested in (according to her goals, history or interests) and the one from the resource (allowing three kinds of comparisons: exact match, generalization or specialization). For example, imagine that results for java are wanted. All the results fitting perfectly in that category may be ranked higher but still it is possible to retrieve specialized resources (e.g., one level less in the hierarchy) and generalized resources (e.g., one level more in the hierarchy: programming languages). The matches in the same category would get a higher rank than others in a different level but all of them would be aggregated. This provides three different semantic rank mechanisms based on user goals, interests and history.

Therefore, the final rank is a (possibly weighted) combination of the previous criteria. In our implementation the user is able to select in the search interface or in her profile which ones must be applied and which ones must not.

- Occurrence - ranking
- Proximity ranking
- Goal-based ranking
- Interests-based ranking
- History-based ranking
- Provider Reputation-based ranking

In addition, to these criteria, it is possible to add link analysis or rank aggregation methods in case explicit links between resources are added or there exist an overlap between the resources available in the providers.

8.2.1 Implementation of the ranking algorithm in a network search client

The search results can be ranked by two different filter types [10], which are combined to get a weight for each learning resource to rank:

- Text
- Categories

The functionality of the text filter is simple: The specified text is searched in the metadata of all results and for each one a weight is calculated. The higher the weight, the better the resource meets the search criteria. For category filtering, the distances from the specified classifications (e.g., interests) in the ontology to the entries specified in the dc:subject field from each resource are evaluated. The weights of the text filter and the distances in the ontology are then combined and normalized to get a weighting value between 0 (bad accordance) and 1 (good accordance). In the following the filters and their combination is explained in detail.

The text filter

The text filter uses Lucene. Lucene is a free text-indexing and -searching API written in Java which offers a simple, yet powerful core API. It is a technology suitable for nearly any application that requires full-text search. The ranking algorithm indexes all resource metadata received from the network query and holds the index in a so-called RAMDirectory (memory persistence) for faster access.

Before the resource metadata is indexed, it is passed through an analyzer. An analyzer is in charge of extracting tokens out of text to be indexed and eliminating the rest. In Lucene several analyzers are available. Some of them deal with skipping stop words (frequently used words that don't help distinguish one document from the other, such as "a", "an", "the", "in", and on); some deal with conversion of tokens to lowercase letters, so that searches are not case-sensitive. More sophisticate analyzers supports stemming. Often, a user desires a query for one word to match other similar words. For example, a query for "jump" should probably also match the words "jumped", "jumper", or "jumps". Reducing a word to its root form is called stemming. If an analyzer does stemming, it has to know the language of the resource. At the time of the implementation, Lucene stemming was just supported for German and Russian.

To achieve good ranking results the choice of the correct analyzer is a critical step. For all results, but German and Russian ones, Lucene's StandardAnalyzer is solely used, which extracts stop words (common English words), but does not support stemming. For German search results the GermanAnalyzer is additionally used. This analyzer uses German stop words and supports, in contrast to the StandardAnalyzer, also stemming. Due the fact that metadata is likely described in English, the resources are not just analyzed by a language sensitive analyzer (like the GermanAnalyzer), but by the StandardAnalyzer too.

After the indexing of the received metadata has been finished, a query is created. For this the query string of the original request is tokenised by an analyzer. To achieve good search results the analyzer for tokenising the query string must be the same as the analyzer used for the index. The preferred language of the learner is used to choose the analyzer. If it is German (or Russian), the GermanAnalyzer (or RussianAnalyzer) and the StandardAnalyzer are used. The split tokens of both analyzers are combined by an OR operation. To maintain the weight of the query the boost factors are halved. If the preferred language is different from German or Russian only the StandardAnalyzer is used.

All search terms (e.g., the learner's interests, goals and learning history) are evaluated separately and combined with an OR operation. The wildcard “*” is added to the start and end of each word (it substitutes for a string of characters of any length). So not only the exact search word, but also any word, which includes the search word, is found and therefore weighted. For example, a query for “class” also match the words “subclass” or “classtree”. Due to the fact that Lucene makes no difference in scoring exact word matches or just closely matching words, both words are equally scored.

The category filter

The category filter is based on the distances from the specified classifications in the ontology. The property dc:subject points to a taxonomy/hierarchy according to the classification of a resource. It is possible to calculate the distance between the elements of the hierarchy the user is interested in (according to her goals, history or interests) and the one from the resource (allowing three kind of comparisons: exact match, generalization or specialization). Imagine a user wants to have results for java. All the results that fits perfectly in that category will be ranked higher but still it is possible to retrieve specialized resources (e.g., one level less in the hierarchy) and generalized resources (e.g., one level more in the hierarchy: programming languages). The matches in the same category would get a higher rank than others in a different level but all of them would be aggregated.

To get a weighting factor out of the calculated distances the following formula is computed:

$$wc_i = \frac{\sum_j \frac{MaxDist_j - Dist_{ij} + 1}{MaxDist_j + 1} Boost_j}{\sum_j Boost_j}$$

- $i \equiv$ Resource index
- $j \equiv$ Classification index
- $wc_i \equiv$ Weight of resource i based on classification
- $Dist_{ij} \equiv$ Distance of resource i in the classification j
- $MaxDist_j \equiv$ Maximum of calculated distances of classification j
- $Boost_j \equiv$ Boost factor of classification j

This formula calculates a weighting factor between 0 (bad accordance) and 1 (good accordance) for each resource based on all classifications to filter.

At the end two weighting factors for each resource are used, one based on the text filter, the other based on the category filter. What still has to be done is to combine these two values to get one value for each resource. To do that, the following formula is used:

$$w_i = \frac{wt_i \sum_j Boost_j + wc_i \sum_k Boost_k}{\sum_j Boost_j + \sum_k Boost_k}$$

- $i \equiv$ Resource index
- $j \equiv$ Classification index
- $w_i \equiv$ Weight of resource i based on text and classification
- $wt_i \equiv$ Weight of resource i based on text
- $wc_i \equiv$ Weight of resource i based on classification
- $Boost_j \equiv$ Boost factor of text j
- $Boost_k \equiv$ Boost factor of classification k

If this formula is applied a value between 0 (bad accordance) and 1 (good accordance) is computed for each result based on the metadata and classifications to filter. These values provide the ranking that is finally provided to the user. An important thing to mention here is that the user can also personalize this complex ranking function and even some parts of it can be enabled or disabled, as will be later be shown in section 9.1.

Chapter 9

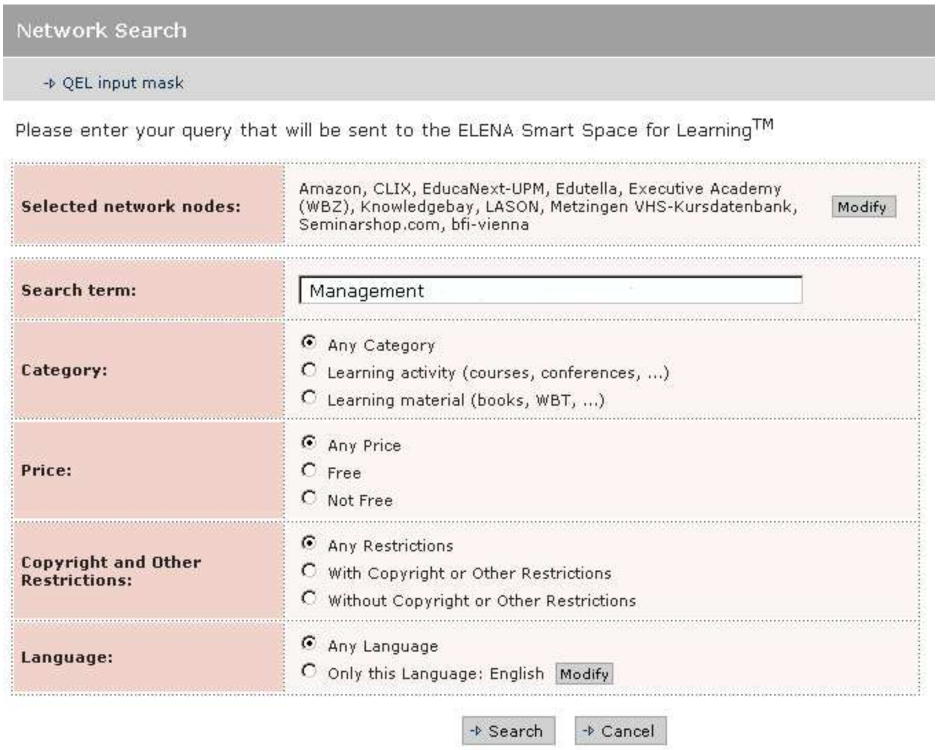
Bringing it all together toward Interoperability

This chapter describes three different scenarios in which the concepts and results from previous chapters are brought together in order to improve system interoperability. This chapter aims at demonstrating how the previously described contributions help to improve each of the crucial steps to be achieved in order to ensure interoperability. Especially, section 9.1 provides an example in which most of them are integrated in a system where an advanced search service is provided in order to query heterogeneous e-learning repositories. The increasing amount of repositories being connected to such a network and the interoperability achievement through several networks of learning resources providers and projects world wide demonstrate the enhancement to interoperability and the success of the contributions of this thesis. Futhermore, this work is not limited to the achievements described in this thesis and it is being continued in the context of several other projects.

9.1 Advanced Network Search

The goal of the EU IST Elena project [48] was the design and implementation of a Smart Space for Learning [131, 39] that allow personalized access to heterogeneous learning services. Such a goal for a Smart Space for Learning relies completely on a network where heterogeneous sources of information, learning resources and services in general are integrated. Furthermore, a study analysis within the training-life-cycle in several companies across Europe [62] discovered the importance of the following requirements:

Retrieving learning services from a wide variety of providers in order to help to get a critical mass of good content in the training depart-



Network Search

QEL input mask

Please enter your query that will be sent to the ELENA Smart Space for Learning™

Selected network nodes:	Amazon, CLIX, EducaNext-UPM, Edutella, Executive Academy (WBZ), Knowledgebay, LASON, Metzingen VHS-Kursdatenbank, Seminarshop.com, bfi-vienna Modify
Search term:	Management
Category:	● Any Category ○ Learning activity (courses, conferences, ...) ○ Learning material (books, WBT, ...)
Price:	● Any Price ○ Free ○ Not Free
Copyright and Other Restrictions:	● Any Restrictions ○ With Copyright or Other Restrictions ○ Without Copyright or Other Restrictions
Language:	● Any Language ○ Only this Language: English Modify

Figure 9.1: HCD-Online network search

ments. This study adds that this task is currently performed manually.

Search heuristics in order to provide results not only for the formulated query, but also results about related topics.

Metadata queries in order to create, for example, queries for resources specifying whether it has a cost associated or not, or whether they fall within a certain budget in order to assist in making training decisions and budget control.

Matching Skill Gaps with Learning Service Selections supporting the learner in selecting the right courses.

Matching personal development goals with learning services in order to maximize the effectiveness of the courses selected as well as the motivation of the learner.

It highlighted as well the importance of the several hypothesis regarding information management [63]: Variety of Learning Resources Offered, Quality of Information and Personalization of Workplace Learning.

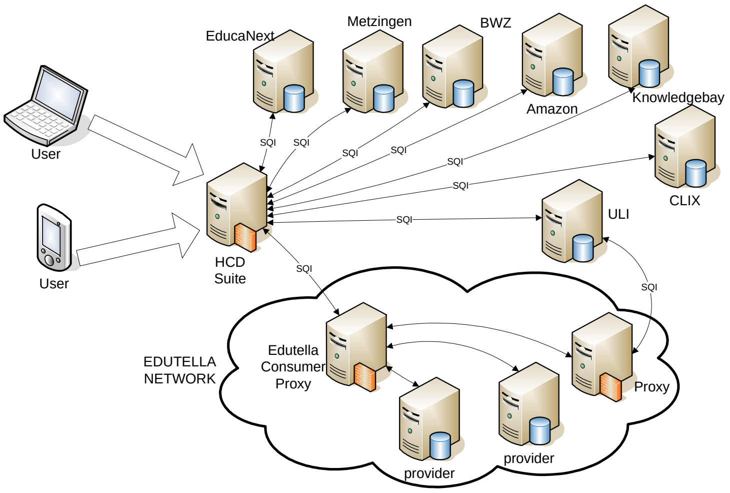


Figure 9.2: ELENA Network Architecture

These requirements showed the need for advanced searching mechanisms within a wide collection of available repositories in order to ease and minimize the time needed to perform tasks that many times are even performed manually and, not less important, to improve the outcomes of those tasks. Towards this goal the ELENA project built an advanced network search mechanisms as part of the HCD-ONLINE system [67] (the Educanext system [45] has also adopted the same advanced network search here described).

In HCD-ONLINE users may specify keywords to be matched in free-text attributes of resources such as title, description or learning goal as well as some other attributes like learning resource category, price, copyrights and restrictions and language (as depicted in figure 9.1). Additionally, it is possible to select on which repositories the search should be performed. Current supported repositories include, among others, Amazon [6], CLIX [26], EDUTELLA [47] (see section 5.2.1, EducaNext [45], EduSource [46], Executive Academy (WBZ) [152], Knowledgebay [85], LASON [87], Seminarshop.com [128] and ULI [146] (see figure 9.2).

Interoperability among this systems is achieved thanks to the implementation of the Simple Query Interface for the network communication, QEL as common query language and a minimal common schema (see section 6.1). Queries like

$$@prefix\ qel : < http : //www.edutella.org/qel\# > .$$
$$@prefix\ dc : < http : //purl.org/dc/elements/1.1/ > .$$
$$@prefix\ ln : < http : //www.hcd\text{-}online.com/nsv1/ > .$$
$$@prefix\ openq : < http : //www.open\text{-}q.de/ > .$$

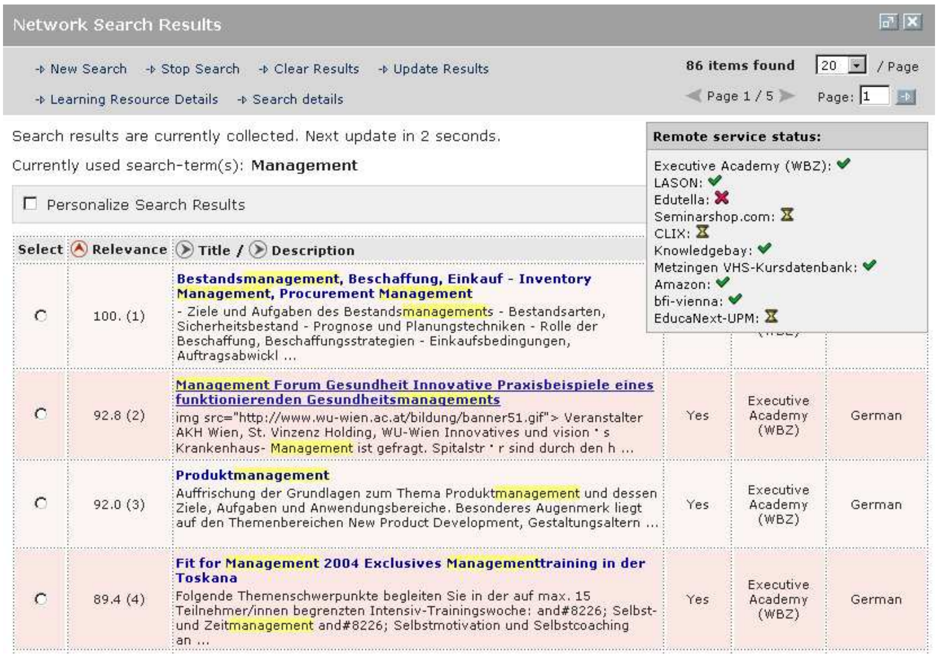
$$
\begin{array}{l}
@prefix\ lom\text{-}rights : < http : //ltsc.ieee.org/2002/09/lom\text{-}rights\# > .\\
?- \quad qel : s(Resource, dc : identifier, Identifier),\\
\qquad qel : s(Resource, dc : subject, Subject),\\
\qquad qel : s(Resource, ln : add\_information, AddInfo),\\
\qquad qel : s(Resource, openq : goal, Goal),\\
\qquad qel : s(Resource, dc : language, LanguageLR),\\
\qquad qel : s(Resource, ln : learning\_resource\_category, CategoryLR),\\
\qquad qel : s(Resource, lom\text{-}rights : cost, Price),\\
\qquad qel : s(Resource, lom\text{-}rights : copyright\_and\_other\_restrictions,\\
\qquad\qquad\qquad Restrictions),\\
\qquad qel : s(Resource, dc : title, Title),\\
\qquad qel : s(Resource, dc : description, Description),\\
\qquad qel : like(Title,' \%keyword\%').
\end{array}
$$

are sent to all the systems in order to retrieve relevant resources. In some cases, translations to local query languages (like XQuery for example) and semantic integration among the common schema and different local schemas supported at each system is needed [132] (Appendix C shows one example for the ULI provider). Moreover, due to the amount of available resources in each system, a general query such as "Management" may return too many results. In order to sort these results ranking, as described in section 8.2 is performed. As depicted in figure 9.3, results are sorted according to relevancy, which is shown on a column on the left. The full algorithm is depicted in figure 9.4.

Furthermore, it is possible for users to tweak the personalization of the system. For that purpose, we provided a personalization panel where the user may select which aspects of her profile (interests, goals and/or history) should be taken into account and to what extent during the generation of the ranking. Figure 9.5 shows on the upper part such a personalization panel.

HCD-Online has been evaluated in two rounds with users from different countries in Europe. The main goals were to test whether the search component finds relevant learning resources in the Elena network, to evaluate implemented ranking algorithms, to test how personalization changes the search results and their ranking, and how user friendly is the use of the search component. This evaluation proved the feasibility of our prototype and generated satisfactory results. In addition, some interesting feedback was provided regarding the ranking algorithm in order to improve its implementations in future releases[1].

[1] Some users reported for example that when using specific queries, personalization using multiple goals from the profile may provide slightly worse rankings if they are not too related.



Network Search Results

→ New Search → Stop Search → Clear Results → Update Results
→ Learning Resource Details → Search details

86 items found 20 / Page
Page 1 / 5 Page: 1

Search results are currently collected. Next update in 2 seconds.

Currently used search-term(s): **Management**

☐ Personalize Search Results

Remote service status:

Executive Academy (WBZ): ✓
LASON: ✓
Edutella: ✗
Seminarshop.com: ⧗
CLIX: ⧗
Knowledgebay: ✓
Metzingen VHS-Kursdatenbank: ✓
Amazon: ✓
bfi-vienna: ✓
EducaNext-UPM: ⧗

Select	Relevance	Title / Description			
○	100. (1)	**Bestandsmanagement, Beschaffung, Einkauf - Inventory Management, Procurement Management** - Ziele und Aufgaben des Bestandsmanagements - Bestandsarten, Sicherheitsbestand - Prognose und Planungstechniken - Rolle der Beschaffung, Beschaffungsstrategien - Einkaufsbedingungen, Auftragsabwickl ...			
○	92.8 (2)	**Management Forum Gesundheit Innovative Praxisbeispiele eines funktionierenden Gesundheitsmanagements** img src="http://www.wu-wien.ac.at/bildung/banner51.gif"> Veranstalter AKH Wien, St. Vinzenz Holding, WU-Wien Innovatives und vision ' s Krankenhaus- Management ist gefragt. Spitalstr ' r sind durch den h ...	Yes	Executive Academy (WBZ)	German
○	92.0 (3)	**Produktmanagement** Auffrischung der Grundlagen zum Thema Produktmanagement und dessen Ziele, Aufgaben und Anwendungsbereiche. Besonderes Augenmerk liegt auf den Themenbereichen New Product Development, Gestaltungsaltern ...	Yes	Executive Academy (WBZ)	German
○	89.4 (4)	**Fit for Management 2004 Exclusives Managementtraining in der Toskana** Folgende Themenschwerpunkte begleiten Sie in der auf max. 15 Teilnehmer/innen begrenzten Intensiv-Trainingswoche: and#8226; Selbst- und Zeitmanagement and#8226; Selbstmotivation und Selbstcoaching an ...	Yes	Executive Academy (WBZ)	German

Figure 9.3: HCD-Online network search results

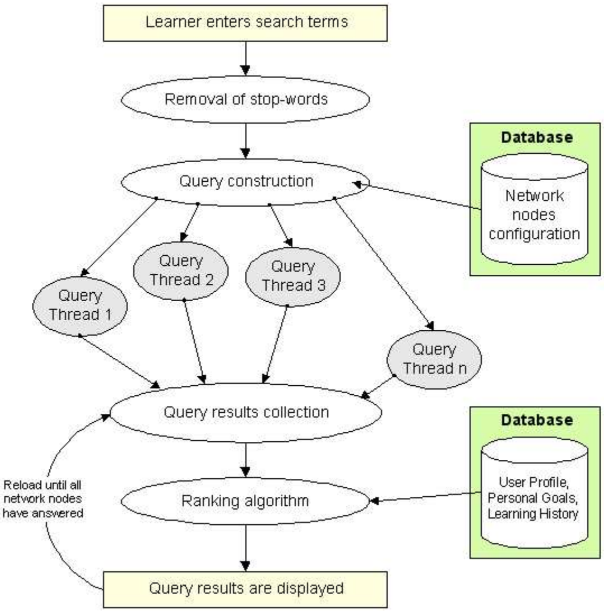
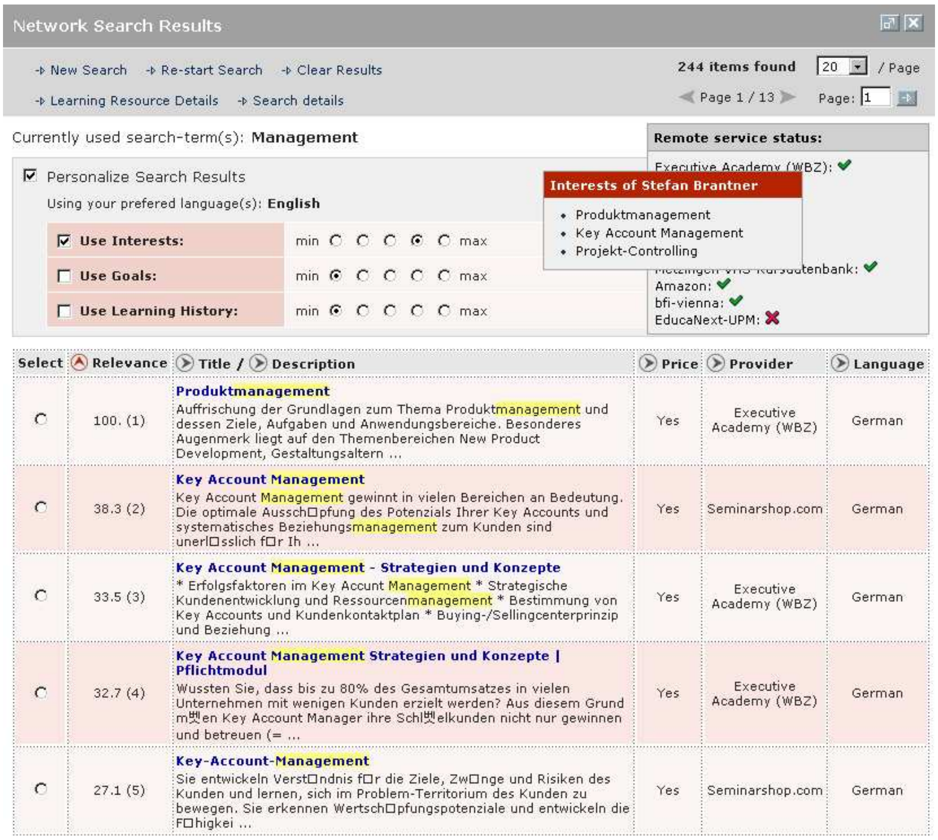

Learner enters search terms
Removal of stop-words
Query construction
Database
Network nodes configuration
Query Thread 1
Query Thread 2
Query Thread 3
Query Thread n
Query results collection
Database
User Profile, Personal Goals, Learning History
Reload until all network nodes have answered
Ranking algorithm
Query results are displayed

Figure 9.4: HCD-Online network search and ranking algorithm



Network Search Results

New Search · Re-start Search · Clear Results · Learning Resource Details · Search details

244 items found 20 / Page

Page 1 / 13 Page: 1

Currently used search-term(s): **Management**

☑ Personalize Search Results

Using your prefered language(s): **English**

- ☑ **Use Interests:** min ○ ○ ○ ● ○ max
- ☐ **Use Goals:** min ● ○ ○ ○ ○ max
- ☐ **Use Learning History:** min ● ○ ○ ○ ○ max

Interests of Stefan Brantner

- Produktmanagement
- Key Account Management
- Projekt-Controlling

Remote service status:

Executive Academy (WBZ): ✓
Metzingen VHS Kursdatenbank: ✓
Amazon: ✓
bfi-vienna: ✓
EducaNext-UPM: ✗

Select	Relevance	Title / Description	Price	Provider	Language
○	100. (1)	**Produktmanagement** Auffrischung der Grundlagen zum Thema Produktmanagement und dessen Ziele, Aufgaben und Anwendungsbereiche. Besonderes Augenmerk liegt auf den Themenbereichen New Product Development, Gestaltungsaltern ...	Yes	Executive Academy (WBZ)	German
○	38.3 (2)	**Key Account Management** Key Account Management gewinnt in vielen Bereichen an Bedeutung. Die optimale Aussch□pfung des Potenzials Ihrer Key Accounts und systematisches Beziehungsmanagement zum Kunden sind unerl□sslich f□r Ih ...	Yes	Seminarshop.com	German
○	33.5 (3)	**Key Account Management - Strategien und Konzepte** * Erfolgsfaktoren im Key Accunt Management * Strategische Kundenentwicklung und Ressourcenmanagement * Bestimmung von Key Accounts und Kundenkontaktplan * Buying-/Sellingcenterprinzip und Beziehung ...	Yes	Executive Academy (WBZ)	German
○	32.7 (4)	**Key Account Management Strategien und Konzepte | Pflichtmodul** Wussten Sie, dass bis zu 80% des Gesamtumsatzes in vielen Unternehmen mit wenigen Kunden erzielt werden? Aus diesem Grund m□en Key Account Manager ihre Schl□elkunden nicht nur gewinnen und betreuen (= ...	Yes	Executive Academy (WBZ)	German
○	27.1 (5)	**Key-Account-Management** Sie entwickeln Verst□ndnis f□r die Ziele, Zw□nge und Risiken des Kunden und lernen, sich im Problem-Territorium des Kunden zu bewegen. Sie erkennen Wertsch□pfungspotenziale und entwickeln die F□higkei ...	Yes	Seminarshop.com	German

Figure 9.5: HCD-Online network search personalization options

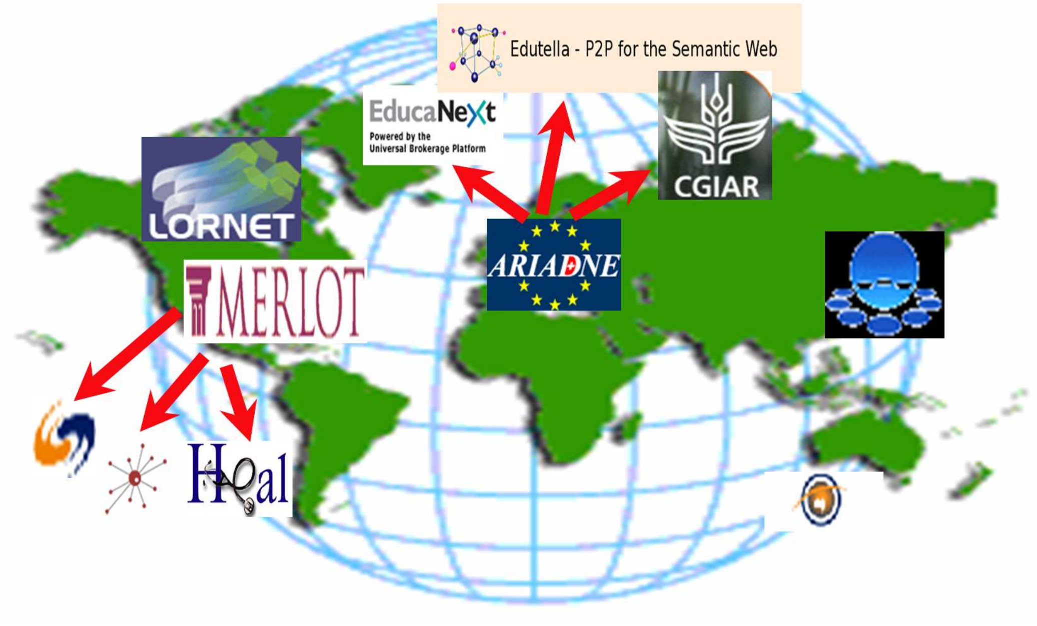

Edutella - P2P for the Semantic Web
EducaNext
Powered by the Universal Brokerage Platform
CGIAR
LORNET
ARIADNE
MERLOT
HEal

Figure 9.6: The Globe Network

9.2 Bringing Learning Repositories to a Global Network

The Network of Excellence PROLEARN [118] and the GLOBE project [56] have spent quite some effort towards interoperability. PROLEARN's mission is to bring together the most important research groups in the area of professional learning and training, as well as other key organizations and industrial partners. Here professional learning is understood as any form of learning in and for the professional life of a citizen including respective individual as well as organizational aspects of employers and employees, due to the rapidly change professions with regard to their specialized knowledge, skills and required competences. The "Learning Objects, Metadata and Standards" PROLEARN workgroup focuses specially on learning object interoperability and supports SQI for that purpose (see `http://www.prolearn-project.org/lori/` for more information). The Global Learning Objects Brokered Exchange (GLOBE) is an international consortium that strives to make shared on-line learning resources available to educators and students around the world. It aims to connect the world and unlock the "deep web" of quality on-line educational resources through brokering relationships with content providers relying on the SQI as the basis for communication (see figure 9.6).

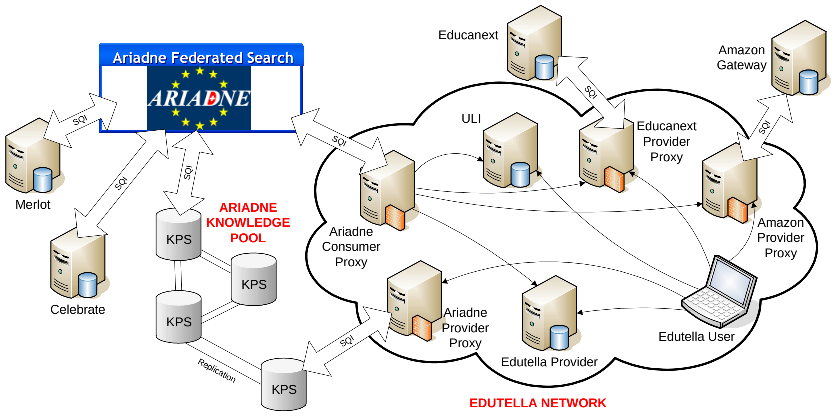


Figure 9.7: ARIADNE and EDUTELLA integration

Within these two projects, SQI has been evaluated by several prototype implementations demonstrating its universal applicability. An open registry with information about all this providers is available at `http://ariadne.cs.kuleuven.be/SqiInterop/free/SQIImplementationsRegistry.jsp`. A non-exhaustive list of the repositories involved in these initiatives includes Ariadne [7], CGIAR [19], EdNA Online [44], EducaNext [45], EDUTELLA [47], Fire [53], Lion [92], LORNET [94], Merlot [97], Nime[105] and PlanetDR-URV [116].

As an example, figure 9.7 shows how ARIADNE and EDUTELLA has been connected [145]. Additionally, via EDUTELLA proxies it is now possible to expose content of previous EDUTELLA providers to external systems via SQI. Some examples of such providers include the Media Library (a joint project between the KMR group [84] and the Swedish Educational Radio and Television [149]), Nature and Technology (Swedish National Agency for Education), Confolio System (portfolios hosted at Royal Institute of Technology) and University teaching network for computer science (ULI) [146].

9.3 Knowledge Resource Sharing for Life Long Learning

Many other projects such as TENCOMPETENCE [144] (Building the European Network for Lifelong Competence Development), MACE [95] (Metadata for Architectural Contents in Europe), MELT [96], ACKNOWLEDGE [2] (competencies in eLearning) or PUBELO [120] (development of LOM application profile for Flemish publishers) are also currently using SQI. In particular, TENCOMPETENCE aims at meeting the needs of individuals, groups

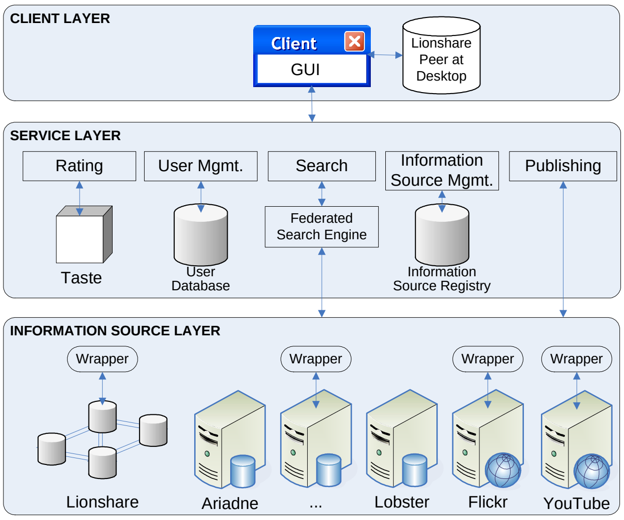


Figure 9.8: TENCompetence Knowledge Sharing Infrastructure

and organizations in Europe for lifelong competence development [68, 17] by establishing the best infrastructure which is possible today, using open-source, standards-based, sustainable and extensible technology [16]. Such an infrastructure relies on a knowledge management layer in which information is made accessible in order to better support lifelong learning and at the same time enhance the learning experience [37]. Such a layer would bring together that information stored for example in institutional servers and learning management systems (centralized repositories), locally on learner desktops (by means of P2P networks) and online community-sharing systems like online-storage applications, wikis or blogs (see figure 9.8). In order to achieve this aim, SQI has been the choice for the search interface [37] and similar specifications (together with ARIADNE) are ongoing in order to standardized, among others, publishing services (storage, update and deletion), user management and repository management services.

Chapter 10

Conclusions and Open Issues (English)

This document provides a review of the state of the art in the field of interoperability of distributed systems with a special focus on the e-learning community. It describes in detail not only the motivational aspects of providing interoperability from both the information provider's and consumer's point of view but also the different components that are required in order to ensure it. Moreover, the results presented in this document contribute to enhance the overall interoperability perspective in current e-learning management systems and online learning object repositories as well as each of the steps that are to be followed towards such a goal. The objective of the work described in this document was to improve or overcome the main challenges for interoperability in order to enhance existing approaches and increase its efficiency and effectiveness from both provider and consumer perspective, and therefore increasing the amount of information available to users while reducing provider costs.

The main contributions of this document are:

Identification of requirements for system interoperability.
Which are the aspects two or more systems need to agree upon in order to be able to exchange and use their information. Those aspects include a common communication protocol and interface, common query language, common schema (and/or semantic integration techniques) and advanced ranking algorithms. All of them are crucial to ensure interoperability.

Specification and standardization of a simple query interface to be adopted by systems willing to be interoperable.

The Simple Query Interface has been presented. This standard specification provides a simple solution in terms of number of methods and implementation costs as well as a flexible solution that targets different interoperability scenarios, including for example, synchronous, asynchronous, stateless and stateful communications. SQI is an official CEN/ISSS Workshop Agreement since October 2005. Furthermore, it is also one of the protocols listed in an official document published by IMS on Query Services [75] and it has been adopted by a large number of repositories making possible networks of repositories that did not exist before its creation.

Development of open source components (based on SQI) to be adopted by information providers.
Components have been developed in order to reduce the effort and costs of information provides using RDBMS or RDF backend repositories. These components are available as open source in order to maximize the number of providers using them and the amount of systems using them demonstrates its success.

Specification of a proxying architecture in order to open (typically) closed environments to other consumers and providers.
Some environments require the implementation of a specific interface in order to be able to share and exchange information. One example of this kind of environments is P2P networks. This document presents a proxying architecture based on SQI in order to overcome this limitation and access and share information along the borders of the "closed" environment. This architecture has been used by several systems in order to bring interoperability to heterogenous networks of repositories.

Creation of ontologies and data models to annotate learning material and to represent complex competences.
In order to be able to understand each other, two systems need to use the same vocabulary. The richer this vocabulary is the more advanced the communication might be. Ontologies are created in order to bring interoperability to the resources they are used to describe. An ontology for learning resources is presented in order to classify and describe the metadata that may be later used to find potential relevant material. However, it is difficult to agree on a single universal vocabulary and therefore this document suggests a two-level split: one for basic sharing of information and a more complex one for advanced services. In addition, a data model is provided in order to represent competences in an interoperable way allowing advanced gap analysis and

searches based on competence goals and requirements and achievements.

Presentation of mappings and its application via query rewriting mechanisms in order to provide components for effective and low-cost semantic integration.
The impossibility of having a single universal schema at all levels is readily accepted. However, vocabularies may have similar concepts, may be able to be translated from one to another and there might even be some implicit concepts within the data. The mappings presented in this document allow the connection of repositories at the schema level, given a simple set of datalog mapping rules. An opensource version of a mediator using such techniques has been implemented and is currently used by many repositories therefore demostrating its need and success.

Description of new ranking algorithms, one providing personalized results to the user assuming the existence of relationships among resources and another to adequate for unlinked corpus.
Given a specific query, a large number of results might be considered relevant. In order to reduce the time users need to filter those large result sets, ranking algorithms are needed. These ranking algorithms need to take into account user preferences and interests in order to personalize those results. This document presents two algorithms: one for personalized ranking in linked corpus (e.g., the Web) and another for (optionally personalized) ranking on unlinked corpus. The former retrieves information from the user in order to find out her preferences and interests and to provide a personalized ranking. The latter provides a weighted mechanism to rank results retrieved from heterogeneous sources based on metadata attributes, ontologies and user profiles.

Integration of all the previous items in a system as a proof of concept interoperability demonstration.
All previous contributions help to improve each of the crucial steps to be achieved in order to ensure interoperability. This thesis also provides a example as a proof of concept in which most of them are integrated into a single system where an advanced search service is provided in order to query heterogeneous e-learning repositories.

Demonstration of the interoperability achievement through several networks of learning resources providers and projects world wide.
The work performed has successfully brought interoperability to several networked repositories demonstrating its feasibility and success. The increasing

amount of repositories being connected to such networks networks of learning resources providers and projects world wide demonstrate the enhancement to interoperability and the success of the contributions of this thesis. Futhermore, this work is not limited to the achievements described in this thesis and it is being continued in the context of several other projects.

Although this document describes several improvements to the state of the art in interoperability, they are just small steps. There are many issues which are still subject of open research. This document has presented a standard for a search interface. However, there is a lack of standards for other basic services related to interoperability like publishing interfaces, repository management and user management. Collaboration between ARIADNE and TENCompetence has taken steps towards establishing a standard interface definition for these services [37].

In addition, this document has used QEL, a powerful and expressive semantic language, as a common query language. However, the performance of some operations, for example, full text queries, should be improved. Furthermore, some repositories do not require having such an expressive language and it would be satisfactory to have a simpler one (e.g., keyword based). Work in PROLEARN is currently being focused on the "PROLEARN Query Language" (PLQL) [117], a query language for learning repositories, which is envisioned to have five layers of increasing expressiveness.

Work on schemas will still be performed in order to adapt to user and market needs. However, I believe it is not possible to have a single universal ontology (not even for each domain) and therefore effort is required on components that allow the specification of mappings among two schemas as well as the (semi-)automatic discovery of such mappings, following the extensive research that has been done in schema mapping and ontology alignment. Furthermore, society evolves to new needs and therererefore new ontologies are required to adapt to its needs. The competence model presented in this document is only an example, but many others will follow in order to address the community/social perspective that is currently emerging.

Finally, ranking is one of the most active research areas in the information retrieval field. Due to the existing amount of data and the information growth rate, new algorithms need to be created. Research in this area includes both the development of new and more advanced algorithms (e.g., personalized ranking) as well as the improvement in the performance and results of existing ones (e.g., parallel computations). However, ranking among distributed heterogeneous sources without links among them have not been extensively exploited yet. Most of the situations in which this scenario occurs use collaborative filtering and recommendation algorithms as ranking measurement.

Chapter 11

Conclusions and Open Issues (Español)

Este documento proporciona una revisión del estado del arte en el campo de interoperabilidad en sistemas distribuidos parcialmente enfocado a la comunidad de aprendizaje. Detalla no sólo la motivación para el trabajo en interoperabilidad y su necesidad desde el punto de vista del consumidor y proveedor de información sino también los diferentes componentes que se requieren para garantizarla. Además, los resultados presentados en este documento contribuyen a mejorar la perspectiva de interoperabilidad en sistemas de gestión de aprendizaje y en repositorios de objetos de aprendizaje online así como cada uno de los pasos a seguir para conseguir dicho objetivo. El trabajo presentado en esta tesis perfecciona o soluciona los principales desafíos en interoperabilidad para mejorar soluciones existentes e incrementar su eficiencia y efectividad desde el punto de vista de consumidores y proveedores. De esta manera, la cantidad de información disponible a usuarios se incrementa drásticamente mientras que los costes de los proveedores se reducen.

Las principales contribuciones de este documento son:

Identificación de requerimientos para la interoperabilidad de sistemas.

Los aspectos en los que dos o más sistemas tienen que llegar a un acuerdo para poder intercambiar y usar información incluyen un protocolo de comunicación común y su respectiva interfaz, un lenguaje de búsqueda común, un vocabulario global (y/o técnicas de integración semántica) así como algoritmos avanzados de ranking. Todos ellos son cruciales para garantizar la interoperabilidad.

Especificación y estandarización de una interfaz de búsqueda simple que sea adoptada por cualquier sistema con intención de ser interoperable.
Se ha presentado la Simple Query Interface, una especificación estándar que proporciona una solución sencilla, en cuanto al número de métodos que la componen y a los costes de implementación, así como flexible que permite su adopción bajo distintos escenarios de interoperabilidad, incluyendo por ejemplo, comunicaciones síncronas, asíncronas, con estado o sin estado. SQI es un CEN/ISSS Workshop Agreement (acuerdo oficial del grupo de trabajo CEN/ISSS) desde octubre del 2005. Además, es también uno de los protocolos incluidos en un documento oficial publicado por el IMS en servicios de búsqueda [75] y ha sido adoptado por un gran número de repositorios haciendo posible redes de repositorios que no existían antes de su creación.

Desarrollo de componentes de código libre (basados en SQI) para su adopción por proveedores de información.
Estos componentes se han desarrollado para reducir el esfuerzo y coste de los proveedores de información usando bases de datos relacionales o repositorios de RDF. Estos componentes de distribuyen como código libre para maximizar el número de proveedores que los usan y la cantidad de sistemas que los utilizan actualmente demuestra su éxito.

Especificación de una arquitectura basada en proxies para abrir entornos (típicamente) cerrados a otros consumidores y proveedores.
Algunos entornos requieren la implementación de interfaces específicas para poder compartir e intercambiar información. Un ejemplo de este tipo de entornos son las redes P2P. Este documento presenta una arquitectura basada en SQI para solucionar esta limitación y acceder y compartir información con sistemas fuera del entorno “cerrado”. Esta arquitectura ha sido usada por varios sistemas para proporcionar interoperabilidad a redes heterogéneas de repositorios.

Creación de ontologías y modelos de datos para la anotación de material de aprendizaje y la representación de competencias complejas.
Para poder entenderse dos sistemas necesitan usar el mismo vocabulario. Cuánto más rico es este vocabulario más avanzada puede ser la comunicación. Las ontologías proporcionan interoperabilidad a los recursos que describen. Este documento presenta una ontología para recursos de aprendizaje para

su clasificación y la descripción de metadatos que pueden ser posteriormente usados para la búsqueda de material relevante. Sin embargo, es difícil encontrar un único vocabulario universal y por tanto, este documento sugiere una división en dos niveles: un vocabulario sencillo para la compartición básica de información y uno más complejo para servicios avanzados. Además, esta tesis presenta un modelo de datos para la representación interoperable de competencias, de manera que búsquedas basadas en competencias, requerimientos y objetivos sea posible.

Presentación de mappings y su aplicación por medio de técnicas de reescritura de búsquedas que permiten proporcionar componentes para la integración semántica de manera efectiva y a bajo coste.
La imposibilidad de tener un único vocabulario universal a todos los niveles es un hecho conocido y aceptado. Sin embargo, los vocabularios pueden tener conceptos similares y en muchos casos se pueden traducir entre ellos (incluso quizás con conceptos implícitos). Los mappings presentados en este documento permiten la conexión de repositorios a nivel de vocabulario, requiriendo únicamente la especificación de un conjunto de reglas de mapeo. Una versión de código libre de un mediador usando estas técnicas ha sido implementado y es usado actualmente por muchos repositorios, de esta manera demostrando su utilidad y éxito.

Descripción de nuevos algoritmos de ranking, un primero proporcionando resultados personalizados a cada usuario (asumiendo la existencia de relaciones entre recursos) y un segundo adecuado a entornos en los cuales no existen relaciones en el conjunto de recursos.
Dada una búsqueda, un número muy grande de resultados puede ser considerado relevante. Para reducir el tiempo que usuarios necesitan para filtrar este conjunto de resultados se necesitan algoritmos de ranking. Este documento presenta dos algoritmos: uno para el ranking personalizado en conjunto de recursos con relaciones (p.e. enlaces) y otro para el ranking (opcionalmente personalizado) de conjuntos de recursos sin relaciones. El primero recibe información del usuario para extraer sus preferencias e intereses y así generar un ranking personalizado. El segundo proporciona un mecanismo ponderado para el ranking de resultados de proveedores basado exclusivamente en metadatos, ontologías y perfiles de usuario.

Integración de todos los elementos anteriores en un sistema como muestra y prueba de concepto de la interoperabilidad obtenida.

Todas las contribuciones anteriores ayudan a mejorar cada uno de los pasos necesarios para proporcionar interoperabilidad. Esta tesis también proporciona un ejemplo en el que la mayoría de estas contribuciones han sido integradas en un único sistema con un servicio de búsqueda en red como muestra de su utilidad y éxito para la búsqueda en repositorios de aprendizaje heterogéneos.

Demostración de los logros obtenidos en interoperabilidad a través de varias redes de proveedores de recursos de aprendizaje y proyectos alrededor del mundo.
El trabajo desarrollado ha conseguido proporcionar interoperabilidad a varias redes internacionales de repositorios demostrando de esta manera no sólo su viabilidad sino también su éxito. El incremento del número de repositorios uniéndose a estas redes de proveedores de recursos de aprendizaje demuestra la mejora en la interoperabilidad respecto a la misma situación años atrás y por tanto demuestra también el éxito de las contribuciones de esta tesis. Además, este trabajo no se limita a lo descrito en esta tesis sino que está siendo extendido y mejorado como parte de otros proyectos.

Aunque este documento describe varias mejoras al estado del arte en interoperabilidad, éstas representan sólo pequeños avances. Existen todavía muchos desafíos que tiene que ser investigados en detalle. Por ejemplo, este documento presenta un estándar para una interfaz de búsqueda. Sin embargo, aún no existen estándares para otros servicios básicos para la interoperabilidad como por ejemplo publicación de recursos, gestión de repositorios o gestión de usuarios (y por tanto sus perfiles). Existe actualmente una colaboración entre ARIADNE y TENCompetence para establecer interfaces estándar para estos servicios [37].

Además, este documento usa QEL, un potente lenguaje de búsqueda, como lenguaje de búsqueda común. Sin embargo, el rendimiento de algunas operaciones, por ejemplo, búsquedas en el texto completo (y no sólo en atributos) debe ser mejorado. Incluso algunos repositorios pueden no requerir búsqueda semántica con un lenguaje tan avanzado y sería más adecuado usar uno más simple (p.e. palabras clave). El proyecto PROLEARN está investigando el “Lenguaje de Búsqueda PROLEARN” (PLQL, siglas del inglés) [117], un lenguaje de búsqueda especializado para repositorios de aprendizaje con cinco niveles de expresividad.

Trabajo en vocabularios seguirá existiendo y adaptándose a las necesidades de mercado. Desde mi punto de vista, no es posible tener un único vocabulario universal (ni siquiera uno por dominio de aplicación) y por tanto se necesita continuar el trabajo en componentes que permitan la especificación (semi-)automática de mappings entre dos vocabularios. Además, la sociedad evoluciona hacia nuevos requerimientos y nuevas ontologías son necesarias. El modelo de competencias presentado en este documento es sólo un ejemplo pero se esperan otros nuevos para representar la perspectiva social que está emergiendo actualmente.

Finalmente, ranking es una de las areas de investigación más activas en el campo de la recuperación de información. Debido a la cantidad existente de información y su ratio de crecimiento, nuevos algoritmos son necesarios. Investigación en este area incluye no sólo el desarrollo de algoritmos nuevos y más avanzados (p.e. ranking personalizado) sino también la mejora en el rendimiento y los resultados de algoritmos existentes. Sin embargo, ranking entre proveedores de información homogéneos, entre cuyos recursos no existen relaciones no han sido explotados todavía. En muchas de las situaciones en las cuales este escenario ocurre se utilizan técnicas de filtrado colaborativo y algoritmos de recomendación como medida de ranking.

Acknowledgments

I would like to thank my supervisors Pablo Castells and Wolfgang Nejdl for their help during the development of my research and their lead during the preparation of my thesis. I also thank Bernd Simon and Stefaan Ternier for their useful comments and discussions as well as their priceless help on many of the issues dealt with in this document. Special thanks go also to many of the people that contributed to and interacted with me in the discussions, specifications or developments of any of the contributions described in this document like[1] Stefan Brantner, Paul Chirita, Peter Dolog, Erik Duval, Tomas Klobucar, Christoph Lofi and Stefan Sobernig. Thanks also to my colleagues Jörg Diederich, Wolf Siberski and Avaré Stewart for their useful comments and corrections to this written thesis. Finally, the research presented here has been partially founded by the projects Elena [48], PROLEARN [118], REWERSE [125] and TENCompetence [144].

[1] In alphabetical order

Bibliography

[1] Karl Aberer, Philippe Cudré-Mauroux, and Manfred Hauswirth. The chatty web: emergent semantics through gossiping. In *International World Wide Web Conferences*, Budapest, Hungary, may 2003.

[2] ACKNOWLEDGE: Accessible & open knowledge infrastructure for flanders. `http://projects.ibbt.be/acknowledge`.

[3] ACM classification. `http://www.acm.org/class/1998/overview.html`.

[4] Alcts/ccs/committee on cataloging: Description and access. task force on metadata: Final report. `http://www.libraries.psu.edu/tas/jca/ccda/tf-meta6.html`, June 2000.

[5] Altavista search engine. `http://www.altavista.com/`.

[6] Amazon.com: Online shopping. `http://www.amazon.com/`.

[7] ARIADNE: Foundation for the european knowledge pool. `http://www.ariadne-eu.org/`.

[8] Frans Van Assche, Erik Duval, David Massart, Daniel Olmedilla, Bernd Simon, Stefan Sobernig, Stefaan Ternier, and Fridolin Wild. Spinning interoperable applications for teaching & learning using the simple query interface. *Educational Technology & Society. Special Issue (April 2006) on Interoperability of Educational Systems*, 9(2):51–67, 2006.

[9] Piero A. Bonatti and Daniel Olmedilla. Driving and monitoring provisional trust negotiation with metapolicies. In *6th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2005)*, pages 14–23, Stockholm, Sweden, June 2005. IEEE Computer Society.

[10] Stefan Brantner, editor. *Smart Learning Space Description.* EU IST ELENA Deliverable, May 2005.

[11] BrightPlanet. `http://www.brightplanet.com/`.

[12] Sergey Brin, Rajeev Motwani, Lawrence Page, and Terry Winograd. What can you do with a web in your pocket? *Data Engineering Bulletin*, 21(2):37–47, 1998.

[13] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[14] Andrei Broder. A taxonomy of web search. Technical report, IBM Research, 2002.

[15] Ingo Brunkhorst and Daniel Olmedilla. Interoperability for peer-to-peer networks: Opening p2p to the rest of the world. In *Innovative Approaches for Learning and Knowledge Sharing, First European Conference on Technology Enhanced Learning (EC-TEL)*, volume 4227 of *Lecture Notes in Computer Science*, pages 45–60, Heraklion, Greece, Oct 2006. Springer.

[16] Daniel Burgos, Eelco Herder, and Daniel Olmedilla. Tencompetence: Construyendo la red europea para el desarrollo continuo de competencias. *Revista Iberoamericana de Inteligencia Artificial (IberoAmerican Journal of Artificial Intelligence)*, 11(33):79–84, 2007.

[17] Daniel Burgos, Martin Memmel, Daniel Olmedilla, Eric Ras, Stephan Weibelzahl, and Martin Wolpers, editors. *Joint International Workshop on Professional Learning, Competence Development and Knowledge Management*, Heraklion, Greece, October 2006.

[18] J. Carriere and R. Kazman. Webquery: Searching and visualizing the web through connectivity. In *Proceedings of the International WWW Conference*, 1997.

[19] CGIAR learning resources center. `http://learning.cgiar.org/`.

[20] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic resource list compilation by analyzing hyperlink structure and associated text. In *Proceedings of the 7th International World Wide Web Conference*, 1998.

[21] D. Chen and G. Doumeingts. European initiatives to develop interoperability of enterprise applications-basic concepts, framework and roadmap. *Annual Reviews in Control*, 27(2):153–162, 2003.

[22] Paul-Alexandru Chirita, Daniel Olmedilla, and Wolfgang Nejdl. Finding related pages using the link structure of the www. Technical report, L3S and University of Hannover. http://www.l3s.de/ chirita/pros/pros.html.

[23] Paul-Alexandru Chirita, Daniel Olmedilla, and Wolfgang Nejdl. Finding related hubs and authorities. In *1st Latin American Web Congress (LA-WEB 2003), Empowering Our Web*, pages 214–215, Santiago, Chile, November 2003. IEEE Computer Society.

[24] Paul-Alexandru Chirita, Daniel Olmedilla, and Wolfgang Nejdl. Finding related pages using the link structure of the WWW. In *2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004)*, pages 632–635, Beijing, China, September 2004. IEEE Computer Society.

[25] Paul-Alexandru Chirita, Daniel Olmedilla, and Wolfgang Nejdl. Pros: A personalized ranking platform for web search. In *3rd International Conference Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2004)*, volume 3137 of *Lecture Notes in Computer Science*, pages 34–43, Eindhoven, The Netherlands, August 2004. Springer.

[26] imc advanced learning solutions. `http://www.clix.de/`.

[27] David Cohn and Huan Chang. Learning to probabilistically identify authoritative documents. In *Proc. 17th International Conf. on Machine Learning*, pages 167–174. Morgan Kaufmann, San Francisco, CA, 2000.

[28] Juri L. De Coi, Eelco Herder, Arne Koesling, Christoph Lofi, Daniel Olmedilla, Odysseas Papapetrou, and Wolf Siberski. A model for competence gap analysis. In *WEBIST 2007, Proceedings of the Third International Conference on Web Information Systems and Technologies: Internet Technology / Web Interface and Applications*, Barcelona, Spain, Mar 2007. INSTICC Press.

[29] Simona Colucci, Tommaso Di Noia, Eugenio Di Sciascio, Francesco M. Donini, Marina Mongiello, and Marco Mottola. A formal approach to ontology-based semantic match of skills descriptions. *J. UCS*, 9(12):1437–1454, 2003.

[30] Concise oxford dictionary. `www.askoxford.com/worldofwords/wordfrom/concise11/`.

[31] Common query language (CQL). http://www.loc.gov/standards/sru/cql/.

[32] A gentle introduction to CQL. http://zing.z3950.org/cql/intro.html.

[33] Dublin core metadata innitiative (DCMI). http://dublincore.org/.

[34] Jos de Bruijn and Axel Polleres. Towards an ontology mapping specification language for the semantic web. DERI Technical Report, jun 2004.

[35] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[36] Anthony J. Delmonte. Enterprise application integration in an e-commerce environment. Find Articles. http://www.findarticles.com/p/articles/mi_qa3766/is_200110/ai_n8992044, December 2001.

[37] Elena Demidova, Stefaan Ternier, Daniel Olmedilla, Erik Duval, Michele Dicerto, Krassen Stefanov, and Naiara Sacristán. Integration of heterogeneous information sources into a knowledge resource management system for lifelong learning. In *The 2nd TenCompetence Workshop: Service Oriented Approaches and Lifelong Competence Development Infrastructures*, Manchester, United Kingdom, January 2007.

[38] DMReview: Information is your business. http://www.dmreview.com/rg/resources/glossary.cfm.

[39] Peter Dolog, Barbara Kieslinger, Zoltán Miklós, Daniel Olmedilla, and Bernd Simon. Creating smart spaces for learning. *Journal of Technology Challenges for Digital Culture*, 7, April 2004.

[40] Peter Dolog, Wolfgang Nejdl, and Daniel Olmedilla, editors. *Artefacts and Service Network v3 (D2.3)*. EU Elena Project, June 2004. http://www.elena-project.org/images/other/D2_3_final.PDF.

[41] Peter Dolog, Wolfgang Nejdl, and Daniel Olmedilla, editors. *Schema Distribution and Evaluation Report (D2.7)*. EU Elena Project, May 2005. http://www.elena-project.org/images/other/D2_7.PDF.

[42] Dutch basic classification codes. http://www.kb.nl/vak/basis/bc98-en.html.

[43] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW '01: Proceedings of the tenth international conference on World Wide Web*, pages 613–622, New York, NY, USA, 2001. ACM Press.

[44] EDNA: Australia's free online network for educators. http://www.edna.edu.au/.

[45] The EducaNext portal for learning resources. http://www.educanext.org/.

[46] eduSource canada: Canadian network of learning object repositories. http://www.edusource.ca/.

[47] EDUTELLA: a p2p networking infrastructure based on rdf. http://edutella.jxta.org/.

[48] ELENA: smart space for learning. http://www.elena-project.org.

[49] Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D. Sivakumar, and Erik Vee. Comparing and aggregating rankings with ties. In *PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 47–58, New York, NY, USA, 2004. ACM Press.

[50] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 301–312, New York, NY, USA, 2003. ACM Press.

[51] Susan Feldman. The high cost of not finding information. KMWorld Magazine. http://www.kmworld.com/Articles/ReadArticle.aspx?ArticleID=9534, March 2004.

[52] Susan Feldman and Chris Sherman. The high cost of not finding information. IDC White Paper, June 2003.

[53] FIRE/LRE: The EUN learning resource exchange. http://fire.eun.org/.

[54] G. Cheetam G. and G. Chivers. *Professions, Competence and Informal Learning.* Edgard Elgar Publishing Limited, 2005.

[55] Rita Gavriloaie, Wolfgang Nejdl, Daniel Olmedilla, Kent E. Seamons, and Marianne Winslett. No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web. In *1st European Semantic Web Symposium (ESWS 2004)*, volume 3053 of *Lecture Notes in Computer Science*, pages 342–356, Heraklion, Crete, Greece, May 2004. Springer.

[56] The global learning objects brokered exchange (GLOBE). `http://globe-info.org/`.

[57] Mario Gomez and Chema Abasolo. A general framework for meta-search based on query weighting and numerical aggregation operators. *Intelligent Systems for Information Processing: From Representation to Applications*, pages 129–140, 2003.

[58] Advanced google search operators. `http://www.google.com/intl/en/help/operators.html`.

[59] Google help: Cheat sheet. `http://www.google.com/intl/en/help/cheatsheet.html`.

[60] Google search engine. `http://www.google.com/`.

[61] Google soap serch api. `http://www.google.com/apis/`.

[62] Sigrún Gunnarsdottir, editor. *Smart Learning Space - Version 1 Trial Report.* EU IST ELENA Deliverable, December 2003.

[63] Sigrún Gunnarsdottir, editor. *User Trials - Evaluation Report.* EU IST ELENA Deliverable, May 2005.

[64] A. Halevy, Z. Ives, D. Suciu, and I. Tatarinov. Schema mediation in peer data management systems. In *Proc. of ICDE*, 2003.

[65] Alon Y. Halevy. Answering queries using views: A survey. *VLDB Journal: Very Large Data Bases*, 10(4):270–294, 2001.

[66] T. Haveliwala. Topic-sensitive pagerank. In *In Proceedings of the Eleventh International World Wide Web Conference, Honolulu, Hawaii*, May 2002.

[67] Human capital development suite (HCD-Online). http://www.hcd-online.com/.

[68] Eelco Herder, Arne Koesling, Daniel Olmedilla, Hans Hummel, Judith Schoonenboom, Ayman Moghnieh, and Luk Vervenne. European lifelong competence development: Requirements and technologies for its realisation. In *Workshop on Learning Networks for Lifelong Competence Development*, Sofia, Bulgaria, March 2006.

[69] Jun Hirai, Sriram Raghavan, Hector García-Molina, and Andreas Paepcke. WebBase: A repository of web pages. In *Proceedings of the Ninth World-Wide Web Conference*, 2000.

[70] HR-XML Measurable Competencies. http://www.hr-xml.org, aug 2004.

[71] Interoperability clearinghouse (ICH). ICH architecture resource center. www.ichnet.org/glossary.htm.

[72] IEEE (institute of electrical and electronics engineers): Standard computer dictionary- a compilation of IEEE standard computer glossaries, 1990.

[73] IMS digital repositories interoperability. http://www.imsproject.org/digitalrepositories/index.html, January 2003.

[74] IMS learning design (IMS-LD). http://www.imsglobal.org/learningdesign/.

[75] IMS query services white paper. http://www.imsglobal.org/query/imsQueryServices.html, June 2005.

[76] IMS Reusable Definition of Competency or Educational Objective (RD-CEO). http://www.imsglobal.org/competencies/, oct 2002.

[77] International survey of e-commerce. World Information Technology and Services Alliance (WITSA). http://www.witsa.org/papers/EComSurv.pdf, 2000.

[78] G. Jeh and J. Widom. Scaling personalized web search. Technical report, Stanford University, 2002.

[79] G. Jeh and J. Widom. Simrank: A measure of structural-context similarity, 2002.

[80] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543, New York, NY, USA, 2002. ACM Press.

[81] Gregory Karvounarakis, Sofia Alexaki, Vassilis Christophides, Dimitris Plexousakis, and Michel Scholl. Rql: a declarative query language for rdf. In *11th International World Wide Web Conference (WWW)*, pages 592–603, Honolulu, Hawaii, USA, May 2002. ACM.

[82] Mohammad Kaykobad, Q. N. U. Ahmed, A. T. M. Shafiqul Khalid, and Rezwan al Bakhtiar. A new algorithm for ranking players of a round-robin tournament. *Computers & OR*, 22(2):221–226, 1995.

[83] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[84] KMR: knowledge management research group. `http://kmr.nada.kth.se`.

[85] knowledgebay: Das medienportal. `http://www.knowledgebay.de/`.

[86] Ka Wai Lam and Chi Ho Leung. Rank aggregation for meta-search engines. In *13th international conference on World Wide Web - Alternate Track Papers & Posters, WWW 2004*, pages 384–385, 2004.

[87] LASON: Latin american studies online. `http://www.lateinamerika-studien.at/en/index.htm`.

[88] Thorsten Lau and York Sure. Introducing ontology-based skills management at a large insurance company. In *Modellierung 2002, Modellierung in der Praxis - Modellierung für die Praxis*, pages 123–134, Tutzing, Deutschland, mar 2002.

[89] Bernard Lefebvre, Gilles Gauthier, Serge Tadié, Tran Huu Duc, and Hicham Achaba. Competence ontology for domain knowledge dissemination and retrieval. *Applied Artificial Intelligence*, 19(9-10):845–859, 2005.

[90] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks (Amsterdam, Netherlands: 1999)*, 33(1–6):387–401, 2000.

[91] Maurizio Lenzerini. Data integration: A theoretical perspective. In *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 233–246, Wisconsin, USA, jun 2002.

[92] The lionshare project. `http://lionshare.its.psu.edu/`.

[93] 1484.12.1 IEEE standard for learning object metadata. `http://ltsc.ieee.org/wg12`, June 2002.

[94] LORNET: Portals and services for knowledge management and learning on the semantic web. `http://www.lornet.org/`.

[95] MACE: Metadata for architectural contents in europe. `http://mace-project.eu/`.

[96] MELT: Learning resources for schools. `http://info.melt-project.eu/`.

[97] MERLOT: Multimedia educational resource for learning and online teaching. `http://www.merlot.org/`.

[98] Paul Miller. Interoperability. what is it and why should i want it? *Ariadne*, 24, June 2000.

[99] Money for growth. the european technology investment report 2005. PricewaterhouseCoopers Report. `http://www.pwc.com/extweb/pwcpublications.nsf/docid/3358B35D1B54C053852%57038006FCFB6`, Jun 2005.

[100] Jehad Najjar, Erik Duval, Stefaan Ternier, and Filip Neven. Towards interoperable learning object repositories: The ariadne experience. In *IADIS International Conference WWW/Internet*, Algarve, Portugal, nov 2003.

[101] Wolfgang Nejdl, Boris Wolf, Changtao Qu, Stefan Decker, Michael Sintek, Ambjörn Naeve, Mikael Nilsson, Matthias Palmer, and Tore Risch. Edutella: A P2P networking infrastructure based on RDF. In *11th International World Wide Web Conference (WWW'02)*, Hawaii, USA, jun 2002.

[102] Andrew Y. Ng, Alice X. Zheng, and Michael I. Jordan. Link analysis, eigenvectors and stability. In *IJCAI*, pages 903–910, 2001.

[103] Andrew Y. Ng, Alice X. Zheng, and Michael I. Jordan. Stable algorithms for link analysis. In *Proc. 24th Annual Intl. ACM SIGIR Conference*. ACM, 2001.

[104] M. Nilsson and W. Siberski. RDF query exchange language (QEL) - concepts, semantics and RDF syntax. `http://edutella.jxta.org/spec/qel.html`, 2003.

[105] NIMEglad: Gateway to learning for ability development. `http://nime-glad.nime.ac.jp/en/`.

[106] NISO (national information standards organization). understanding metadata. NISO Press. `http://www.niso.org/standards/resources/UnderstandingMetadata.pdf`, 2004.

[107] Open knowledge initiative. `http://www.okiproject.org/`.

[108] Daniel Olmedilla. Working with edutella. technical report. `http://www.l3s.de/~olmedilla/projects/edutella/edutella.pdf`.

[109] Daniel Olmedilla. Finding hubs for personalized web search. different ranks to different users. Tribunal de Estudios Avanzados (TEA). Universidad Autónoma de Madrid, September 2003.

[110] Daniel Olmedilla and Matthias Palmér. Interoperability for peer-to-peer networks: Opening p2p to the rest of the world. In *WWW Workshop on Interoperability of Web-Based Educational Systems*, volume 143 of *CEUR Workshop Proceedings*, Chiba, Japan, May 2005. Technical University of Aachen (RWTH).

[111] Daniel Olmedilla, Nobuo Saito, and Bernd Simon, editors. *Proceedings of the WWW'05 Workshop on Interoperability of Web-Based Educational Systems*, volume 143 of *CEUR Workshop Proceedings*, Chiba, Japan, May 2005. CEUR-WS.org.

[112] Daniel Olmedilla, Nobuo Saito, and Bernd Simon, editors. *Educational Technology & Society. Special Issue (April 2006) on Interoperability of Educational Systems*, volume 9, 2006.

[113] Open directory project. `http://dmoz.org/`.

[114] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[115] Christos H. Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 1-3, 1998, Seattle, Washington*, pages 159–168. ACM Press, 1998.

[116] PLANET digital repository. `http://planet.urv.es/planetdr/`.

[117] PLQL: PROLEARN query language definition. `http://ariadne.cs.kuleuven.be/lomi/index.php/QueryLanguages_stable`.

[118] PROLEARN: professional learning. `http://www.prolearn-project.org/`.

[119] PROS project home page. `http://www.l3s.de/~chirita/pros/pros.html`.

[120] PUBELO project. `http://ariadne.cs.kuleuven.be/pubelo/`.

[121] C. Qu and W. Nejdl. Interacting edutella/JXTA peer-to-peer network with web services. In *2004 International Symposium on Applications and the Internet (SAINT 2004)*, Tokyo, Japan, jan 2004. IEEE Computer Society Press.

[122] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *VLDB J.*, 10(4):334–350, 2001.

[123] IEEE 1484.20.1/draft - draft standard for Reusable Competency Definitions (RCD). `http://ieeeltsc.org/wg20Comp/Public/IEEE_1484.20.1.D3.pdf`, nov 2005.

[124] RDQL: A query language for RDF. `http://www.w3.org/Submission/RDQL/`.

[125] REWERSE: Reasoning on the web with rules and semantics. `http://rewerse.net/`.

[126] Dirk Riehle. Composite design patterns. In *ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages & Applications*, pages 218–228, 1997.

[127] Shareable content object reference model (SCORM). `http://www.adlnet.gov/scorm/index.cfm`.

[128] Seminar-Shop. `http://www.seminar-shop.com/`.

[129] Sesame. http://www.openrdf.org/.

[130] B. Simon, D. Massart, F. van Assche, S. Ternier, E. Duval, S. Brantner, D. Olmedilla, and Z. Miklós. A simple query interface for interoperable learning repositories. In *WWW Workshop on Interoperability of Web-Based Educational Systems*, volume 143 of *CEUR Workshop Proceedings*, Chiba, Japan, May 2005. Technical University of Aachen (RWTH).

[131] Bernd Simon, Peter Dolog, Zoltán Miklós, Daniel Olmedilla, and Michael Sintek. Conceptualising smart spaces for learning. *Journal of Interactive Media in Education. Special Issue on the Educational Semantic Web*, 1, 2004.

[132] Bernd Simon, Stefan Sobernig, Fridolin Wild, Sandra Aguirre, Stefan Brantner, Peter Dolog, Gustaf Neumann, Gernot Huber, Tomaz Klobucar, Sascha Markus, Zoltán Miklós, Wolfgang Nejdl, Daniel Olmedilla, Joaquín Salvachúa, Michael Sintek, and Thomas Zillinger. Building blocks for a smart space for learning$^{\text{tm}}$. In *6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006)*, pages 309–313, Kerkrade, The Netherlands, July 2006. IEEE Computer Society.

[133] Simple query interface. http://www.prolearn-project.org/lori/.

[134] Structured query language (SQL). http://en.wikipedia.org/wiki/SQL.

[135] Simple Reusable Competency Map proposal (SRCM). http://www.ostyn.com/resources.htm, feb 2006.

[136] Suryanarayana M. Sripada. Information management challenges from the aerospace industry. In *Proceedings of 28th International Conference on Very Large Data Bases (VLDB)*, pages 1006–1007, Hong Kong, China, August 2002.

[137] SRU: (search/retrieve via URL). http://www.loc.gov/standards/sru/.

[138] SRW: Search/retrieve web service. http://www.loc.gov/standards/sru/srw/index.html.

[139] Steffen Staab, Bharat K. Bhargava, Leszek Lilien, Arnon Rosenthal, Marianne Winslett, Morris Sloman, Tharam S. Dillon, Elizabeth

Chang, Farookh Khadeer Hussain, Wolfgang Nejdl, Daniel Olmedilla, and Vipul Kashyap. The pudding of trust. *IEEE Intelligent Systems*, 19(5):74–88, 2004.

[140] Danny Sullivan. The invisible web gets deeper. *The Search Engine Report*, aug 2000.

[141] York Sure, Alexander Maedche, and Steffen Staab. Leveraging corporate skill knowledge - from proper to ontoproper. In D. Mahling and U. Reimer, editors, *3rd International Conference on Practical Aspects of Knowledge Management*, Basel, Switzerland, oct 2000.

[142] Survey: Integration costs still hamper agility. Computerworld Today. `http://www.itworld.com/AppDev/641/060206integration/`, February 2006.

[143] Igor Tatarinov and Alon Halevy. Efficient query reformulation in peer data management systems. In *SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 539–550, New York, NY, USA, 2004. ACM Press.

[144] TENCompetence: building the european network for lifelong competence development. `http://www.tencompetence.org/`.

[145] Stefaan Ternier, Daniel Olmedilla, and Erik Duval. Peer-to-peer versus federated search: towards more interoperable learning object repositories. In *2005 World Conference on Education, Multimedia, Hypermedia & Telecommunications (ED-MEDIA)*, Montreal, Canada, July 2005. Association for the Advancement of Computing in Education (AACE).

[146] ULI: Partially virtual computer science curriculum. `http://www.uli-campus.de/index\_en.html`.

[147] Jeffrey D. Ullman. *Principles of Database and Knowledge-Base Systems, Volume I.* Computer Science Press, 1988.

[148] Jeffrey D. Ullman. Information integration using logical views. *Theoretical Computer Science*, 239(2):189–210, 2000.

[149] UR: swedish educational radio and television. `http://www.ur.se`.

[150] Representing vCard objects in RDF/XML. `http://www.w3.org/TR/vcard-rdf`.

[151] Visa europe. `http://www.visaeurope.com/glossary/`.

[152] Executive academy der wirtschaftsuniversität wien. http://www.executiveacademy.at/.

[153] Stanford webbase project.
http://www-diglib.stanford.edu/ testbed/doc2/webbase/.

[154] Webster online dictionary. http://www.webster.com/.

[155] Whatis?com: The leading it encyclopedia and learning center. http://www.whatis.com/.

[156] Wikipedia: The free encyclopedia. http://en.wikipedia.org/.

[157] Xquery 1.0: An xml query language. http://www.w3.org/TR/xquery/, November 2006.

[158] Yahoo! search engine. http://www.yahoo.com.

[159] Information retrieval (Z39.50): Application service definition and protocol specification. ANSI/NISO Z39.50-2003, NISO Press, November 2002.

Appendices

Appendix A

QEL Example Queries

A.1 General Queries

Query for all resources with Dublin Core title, description, language, subject and rights:

$$
\begin{array}{l}
@prefix\ qel : < http : //www.edutella.org/qel\# > .\\
@prefix\ dc : < http : //purl.org/dc/elements/1.1/ > .\\
?-\ qel : s(Resource, dc : title, Title),\\
\quad qel : s(Resource, dc : description, Description),\\
\quad qel : s(Resource, dc : language, Language),\\
\quad qel : s(Resource, dc : subject, Subject),\\
\quad qel : s(Resource, dc : rights, Rights).
\end{array}
$$

Query for all resources with Dublin Core language encoded using RFC1766:

$$
\begin{array}{l}
@prefix\ qel : < http : //www.edutella.org/qel\# > .\\
@prefix\ dc : < http : //purl.org/dc/elements/1.1/ > .\\
@prefix\ dcq : < http : //dublincore.org/2000/03/13/dcq\# > .\\
?-\ qel : s(Resource, dc : language, Language),\\
\quad qel : s(Language, dcq : RFC1766, LanguageRFC).
\end{array}
$$

Query for all resources with dc:creator encoded using vcard full name:

$$
\begin{array}{l}
@prefix\ qel : < http : //www.edutella.org/qel\# > .\\
@prefix\ dc : < http : //purl.org/dc/elements/1.1/ > .\\
@prefix\ vcard : < http : //www.w3.org/2001/vcard\text{-}rdf/3.0\# > .\\
?-\ qel : s(Resource, dc : creator, Creator),\\
\quad qel : s(Creator, vcard : FN, Name).
\end{array}
$$

Query for all resources with LOM typical learning time, catalog entry, location and version:

```
@prefix qel : < http : //www.edutella.org/qel# > .
@prefix lom_edu :
    < http : //www.imsproject.org/rdf/imsmd_educationalv1p2# > .
@prefix lom_gen :
    < http : //www.imsproject.org/rdf/imsmd_generalv1p2# > .
@prefix lom_tech :
    < http : //www.imsproject.org/rdf/imsmd_technicalv1p2# > .
@prefix lom_life :
    < http : //www.imsproject.org/rdf/imsmd_lifecyclev1p2# > .
?− qel : s(Resource, lom_edu : typicallearningtime, LearningTime),
    qel : s(Resource, lom_gen : catalogentry, CatalogEntry),
    qel : s(Resource, lom_tech : location, Location),
    qel : s(Resource, lom_life : version, Version).
```

A.2 Queries with conjunctive restrictions on keywords

Query for all resources with Dublin Core title, description, creator, language, subject and rights which creator is “Daniel Olmedilla” and title contains both “policy” and “negotiation”:

```
@prefix qel : < http : //www.edutella.org/qel# > .
@prefix dc : < http : //purl.org/dc/elements/1.1/ > .
?− qel : s(Resource, dc : title, Title),
    qel : s(Resource, dc : description, Description),
    qel : s(Resource, dc : creator, Creator),
    qel : s(Creator, vcard : FN, Name),
    qel : s(Resource, dc : language, Language),
    qel : s(Resource, dc : subject, Subject),
    qel : s(Resource, dc : rights, Rights),
    qel : like(Title,′ %policy%′),
    qel : like(Title,′ %negotiation%′),
    qel : equals(Name,′ DanielOlmedilla′).
```

A.3 Queries with disjunctive restrictions on keywords

Query for all resources with Dublin Core title, description, creator, language, subject and rights which creator is “Daniel Olmedilla” and the keywords "policy" and "negotiation" appear on title, description or subject (but not necessarily both in the same field):

$$
\begin{array}{l}
@prefix\ qel : < http : //www.edutella.org/qel\# > .\\
@prefix\ dc : < http : //purl.org/dc/elements/1.1/ > .\\
searchInFields(X, Y) : -qel : s(X, dc : title, Y).\\
searchInFields(X, Y) : -qel : s(X, dc : description, Y).\\
searchInFields(X, Y) : -qel : s(X, dc : subject, Y).\\
?-\ \ qel : s(Resource, dc : creator, Creator),\\
\qquad qel : s(Creator, vcard : FN, Name),\\
\qquad qel : s(Resource, dc : language, Language),\\
\qquad qel : s(Resource, dc : rights, Rights),\\
\qquad searchInFields(Resource, Keyword),\\
\qquad qel : like(Keyword,' \%policy\%'),\\
\qquad qel : like(Keyword,' \%negotiation\%'),\\
\qquad qel : equals(Name,' DanielOlmedilla').
\end{array}
$$

Appendix B

Default Value Formalization

Let T be the set of all triples of the form (R, P, O) such that R, P and O are resource, predicate and object respectively. In addition, let D be the set of default values such that $d = (P, V)$ where P is a property and V a literal.

Then, for all queries Q, define $Q \xrightarrow{D}_1 P$ iff

- $Q = (T_1, \ldots, T_{i-1}, T_i, T_{i+1}, \ldots, T_n)$
- $T_i = (R, p, O)$
- $U = \begin{cases} T_i & \text{if } \exists p, d | T_i = (R, p, O), d \in D, d = (p, V), \\ \emptyset & \text{otherwise.} \end{cases}$
- $P = Q \setminus U$

Finally, we denote with $\xrightarrow{D}$ the reflexive transitive closure of $\xrightarrow{D}_1$ and the unique result of rewriting of the query with default values by $Q_2 = removeDV(Q, D)$.

After this process, the resulting query Q_2 of this process is sent to the repository and a result set S is received as an answer to the query.

Then, let U be the set of default values applied in the previous process and for all rows R in S, define $R \xrightarrow{U}_1 W$ iff

- $R = (V_1, \ldots, V_n)$
- $V_{n+1} | d = (P, V_{n+1}), d \in U$
- $W = R \cup V_{n+1}$

and finally we denote with $\xrightarrow{U}$ the reflexive transitive closure of $\xrightarrow{U}_1$ and the unique result of this operation as by $S_2 = addDV(S, U)$ where S_2 is the final result set returned to the query.

Appendix C

Mappings for the ULI course

This appendix presents the mappings used for the ULI [146] provider.

C.1 Property Mappings

$$\begin{aligned}&(X, http://purl.org/dc/elements/1.1/identifier, Y) \leftarrow\\&\qquad\qquad (X, http://www.l3s.de/\ olmedilla/uri, Y).\end{aligned}$$

$$\begin{aligned}&(X, http://purl.org/dc/elements/1.1/contributor, Y),\\&\qquad (Y, http://www.w3.org/2001/vcard\text{-}rdf/3.0\#fn, Z) \leftarrow\\&\qquad\qquad (X, http://purl.org/dc/elements/1.1/creator, Z).\end{aligned}$$

$$\begin{aligned}&(X, http://www.open\text{-}q.de/supplier\_name, Y),\\&\qquad (Y, http://www.w3.org/2001/vcard\text{-}rdf/3.0\#orgname, Z) \leftarrow\\&\qquad\qquad (X, http://www.open\text{-}q.de/supplier\_name, Z).\end{aligned}$$

$$\begin{aligned}&(X, http://www.open\text{-}q.de/organizer\_contact, Y),\\&\qquad (Y, http://www.w3.org/2001/vcard\text{-}rdf/3.0\#fn, Z) \leftarrow\\&\qquad\qquad (X, http://www.open\text{-}q.de/organizer\_contact, Z).\end{aligned}$$

$$\begin{aligned}&(X, http://www.hcd\text{-}online.com/nsv1/add\_information, Y) \leftarrow\\&\qquad\qquad (X, http://www.l3s.de/\ olmedilla/uri, Y).\end{aligned}$$

C.2 Default Values

$$http://purl.org/dc/elements/1.1/language \leftarrow' de'.$$

$$http://purl.org/dc/elements/1.1/creator \leftarrow' JanBrase'.$$

$http : //ltsc.ieee.org/2002/09/lom\text{-}rights\#cost \leftarrow' No'.$

$http : //ltsc.ieee.org/2002/09/lom\text{-}rights\#$
$copyright\_and\_other\_restrictions \leftarrow' No'.$

$http : //ltsc.ieee.org/2002/09/lom\text{-}edu\#$
$learning\_resource\_type \leftarrow' Unknown'.$

$http : //www.open\text{-}q.de/supplier\_name \leftarrow' null'.$

$http : //www.open\text{-}q.de/organizer\_contact \leftarrow' null'.$

$http : //www.open\text{-}q.de/goal \leftarrow' null'.$

$http : //www.open\text{-}q.de/priceamount \leftarrow' 0'.$

$http : //www.open\text{-}q.de/pricecurrency \leftarrow' EUR'.$

$http : //www.hcd\text{-}online.com/nsv1/learning\_resource\_category \leftarrow' LM'.$

Appendix D

HubRank Experimental Results

In this chapter I will present some results of the HubRank algorithm. For these examples we have executed the HubRank algorithm giving more importance to hubs using a dumping factor of 0.25 in order to increase a little bit the difference against the PageRank algorithm.

D.1 Small Graph

To demonstrate the main effects of the algorithm, discussion is here restricted only to a miniature Web graph. Figure D.1 shows a small graph and summarizes the results of applying HubRank, PageRank, HITS and Randomized HITS on it. Then numbers represents the position of the node in the overall ranking with each algorithm. The table D.1 is a comparison between all the results returned for each rank algorithm and the HubRank with the same graph (for the HITS, SALSA and Randomized HITS algorithms the hub scores were taken because the results of the HubRank are biased on hubs). A reader is referred to the technical report [109, 22] for more details.

A good example of the utility of HubRank is the *Project List*. As it is the best hub and a poor authority, HITS and Randomized HITS rank it in a top position, whereas PageRank does not give much importance to it (although it might be very useful). Finally, HubRank, considering both aspects, ranks it on place 4. Similarly, it also gives a slightly higher rank to *Researcher C* (place 8 out of 11 in HubRank, while it is the last in PageRank).

High authorities with no hub value are likely to have a decreased score. *Project B* is a very good authority and therefore still the first in HubRank, but its score is about 10% lower than the one computed with PageRank and this would result in a small rank decrease with bigger graphs. *Project C* has no hub value at all, which is materialized in HubRank by a rank decrease

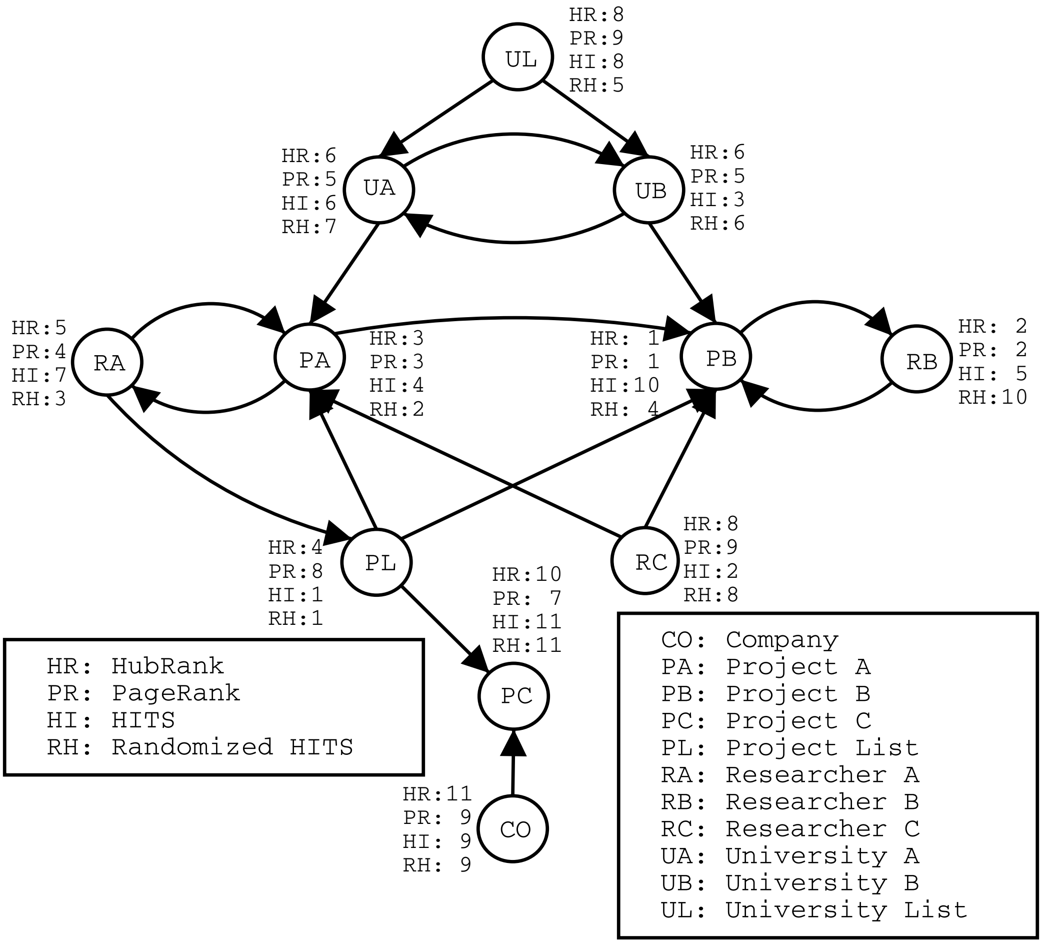


Figure D.1: Small graph ranking comparison

Node	**PageRank**	**HITS**	**SALSA**	**R.HITS**	**HubRank**
Company	0.15	0.09	0.06	0.58	0.15
Project A	0.65	0.31	0.11	1.01	1.13
Project B	3.13	0.00	0.06	1.00	2.83
Project C	0.37	0.00	0.00	0.15	0.30
Project List	0.33	0.58	0.17	1.31	0.73
Researcher A	0.42	0.25	0.11	1.00	0.73
Researcher B	2.81	0.27	0.06	0.52	2.28
Researcher C	0.15	0.49	0.11	0.88	0.31
University A	0.37	0.27	0.11	0.92	0.67
University B	0.37	0.33	0.11	0.92	0.67
University List	0.15	0.11	0.11	0.95	0.31

Table D.1: Score comparison among algorithms.

from 7 to 10 (out of 11) - which we intended.

Generally, we can summarize the outcome of HubRank as follows. A strong authority will always have a good rank, but it will be top ranked only if it is at least an average hub (otherwise it might drop some ranks, compared to PageRank). An average authority will have a rank very dependent on its hub value. Finally, a poor authority will have a low rank if it has a low or average hub value, but it may get a significant raise when it has high hub value (e.g., if one creates a new but very good hub, we try to promote her slightly faster than in PageRank).

HITS and Randomized HITS separate hub values and authority values, which is not useful for us. For example, HITS ranks as second the node *Researcher C* that has no authority at all. On the other hand, PageRank is not considering hub importance at all, which could generate rank drops for important pages, and therefore loss of information. HubRank biases the best authorities of the Web graph according to their value as hubs, so it is more accurate than the other algorithms presented on this example.

D.2 Big crawl

The WebBase crawler [153, 69] was extended in order to store the link structure of the Web and not only the content. As our web crawler is a focused web crawler a root set of domain sites to crawl had to be chosen. Three different crawls were performed[1]: a first one using a full list of universities from Spain and Germany (in order to focus on the educational domain), a second one incrementing it with different domains extracted from the Open Directory Project [113] and the final one on a total of 4,106 different domain sites[2].

In this final link structure 729,384 pages from 4,106 different domains were crawled and 3,587,842 different links were found. The results of the different ranking algorithm over these link structure is here presented.

It is important to mention that these results were computed with a big link structure with 4,106 domains what is good enough to test our algorithms but what is of course not enough to be compared with current search engines. The reader may probably miss pages such as "www.yahoo.com" or "www.netscape.com" among the highest ranked URLs but this situation appears because these domains were not crawled. In addition, a reader can also think that probably the site "www.macromedia.com" should be ranked higher but in our set of domains crawled there were not so many sites pointing to it so that is why it is not in the top authorities. In order to avoid this kind of misunderstandings (and also to compare the effectiveness of HubRank with state of the art algorithms) it was decided to compute the ranks that existing algorithms would compute for this link structure. Results from PageRank point of view are provided in table D.2[3].

In this example, it can be seen how HubRank behaves well in comparison with other algorithms. The best authorities found by PageRank are mostly all well ranked by HubRank (what is not the case by HITS, SALSA or Randomized HITS). Moreover, the top hubs found by HITS in our link structure are also well ranked by HubRank. Therefore, these results show how HubRank acts as a filter to the PageRank ranking by biasing towards hubs.

Table D.3 presents the results from the point of view of the HubRank algorithm.

It can be observed in this new list that all the URLs which are ranked top by HubRank are also top pages from the point of view of the PageRank algorithm, that is, they are good authorities. Moreover, these pages have also good quality as hubs, as it can be seen how these pages are mostly all well ranked by HITS, SALSA and Randomized HITS. Of course, in these experiments more importance is given to the authority value than to the hub

[1] A fourth one was later performed in order to retrieve around 3 million web pages which was used for evaluating the PROS framework.

[2] In the fourth crawl we included 4,887 new sites that our previous crawlers discovered but did not crawl.

[3] Pages marked with an asterisk (*) were ranked outside of the 50000 top pages returned.

URL	PageR	HITS	SALSA	RHITS	HubR
www.trendmicro.com/en/home/...	1	*	*	*	26
www.trendmicro.com/en/about/...	2	*	*	*	27
edis.ifas.ufl.edu	3	*	14895	974	618
www.w3.org	4	33716	8982	1665	36
www.planum.net/lato.htm	5	*	*	8144	2106
www.amoebasoft.com	6	11204	*	*	1
...					
www.oribtz.com	37	*	23406	269	2522
www.acm.org	38	11081	34352	6128	87
www.morrisnathansondesign.com	39	*	*	*	2661
www.planning.org	40	*	3472	16064	3
www.aardvarktravel.net	41	*	*	*	174
www.umass.edu	42	*	*	10916	273
www.voxengo.com/phorum/...	43	*	27077	19252	91
www.macromedia.com	44	*	*	9319	49
...					
www.gardenvisit.com	99	*	13213	1613	2113
www.steinberg.net	100	*	42036	6938	5344
www.kaspersky.com	101	*	35894	26285	86
...					
www.ecoiq.com/syndicated...	1000	*	*	*	2080
www.bangaloreit.com/...	1001	*	33467	2175	7179
www.wired.com/animation/...	1002	44055	*	*	3735

Table D.2: Scores from the ranking algorithms over the whole graph

URL	HubR	HITS	SALSA	RHITS	PageR
www.amoebasoft.com	1	11204	*	*	6
www.amoebasoft.com/index.asp	2	11205	*	*	7
www.planning.org	3	11081	16064	16064	40
...					
www.trendmicro.com/en/about...	27	*	*	*	2
www.visualbuilder.com	28	2292	4938	8804	618
www.java.sun.com	29	32365	*	*	32
...					
www.voxengo.com/phorum.index...	35	*	*	*	31
www.w3.org	36	33716	8982	1665	4
www.maps.com	37	27708	24402	5285	105
...					
stats.bls.gov/iif/home.htm	10000	*	26794	5835	9572
www.ecoiq.com/landuse/magazine...	10001	39124	8642	3974	12069
www.scottlondon.com/bio/index...	10002	*	*	*	4497
...					
www.jspin.com/home/apps/filemani	49999	49509	*	7239	43276
www.stuffit.com/compression/file...	50000	*	*	*	39121

Table D.3: Scores from the ranking algorithms over the whole graph

value (the dumping factor used is 0.25) and that is why some top pages of these algorithms are not top in HubRank as well. HubRank focuses on ranking higher a good combination of both good authorities and good hubs. A good example of its success is that the site "www.java.sun.com" is mainly ranked equally by PageRank and HubRank because it is a good authority and also a good hub but, on the other hand, "www.planning.org" turns out to be a really good hub for the architecture community but it is ranked badly by PageRank.

Appendix E

HubFinder Experimental Results

HubFinder was tested against two extensions of HITS, in which a starting set of pages is extended several times using the Kleinberg extension and in the end a trimming step is performed.

First, the proposed solution in [22] for the trimming formula was used. The number of pages explored by HubFinder is far less than the number of pages explored using the HITS extensions. This also makes HubFinder much faster than the other algorithms tested, while all algorithms used similar amounts of main memory. The percentage of explored pages decreases faster at the initial steps and then slower, depending more on the set size as we explore further. In the end, an additional trimming step was performed, in order to adjust the size of the output set to the desired size. Finally, the resulting sets were a bit different in size, but with very similar content.

HubFinder allows personalization by means of different filtering criteria "plugged into" it, depending on the pages one wants to obtain as result. One could for example use HubRank or PageRank to obtain globally appreciated pages. HubRank is also the best criterion when computing input sets for the Personalized PageRank algorithm [25].

In order to choose the pages we will keep after each iteration we will use different algorithms over the whole graph and keep the best ranked ones. That is why we have executed the algorithm once with each of them.

The pages from the bookmark set shown in the table E.1 were used as starting pages. HubFinder was then applied to this starting pages with different algorithms in the *Select* function so several algorithms were used to decide which pages were kept and which ones discarded. Table E.2 shows the final results.

The *Total* shows the results to apply HubFinder with each algorithm.

www.domus3d.com/default.asp
www.gardenvisit.com
www.urban-advantage.com/index.html
www.metropolismag.com/html/designmart/index.html
phillipsplanning.com/pastprojects1.html
www.museumofarchitecture.com/main.html
deck-porch-gazebo-plan.vadeck.com
www.wrdanielsdesign.com/gallery.html
www.architekturmuseum.ch/htmls/indexe.htm
www.bauhaus.de/english/museum/index.htm
www.bauhaus.de/english/bauhaus1919/architektur/index.htm
chi-athenaeum.org/gdesign/gdesin0.htm
www.archiradar.com/eng/index.htm
gdlalliance.com
www.aztechsoft.com/gpscadrv.htm
www.fatcad.com/home.asp
www.cadfx.com/2004.phtml
www.command-digital.com/panorama1.htm
www.contractcaddgroup.com/3d.htm
www.e-magine.ws/products.htm
www.atlanticarchitects.com/about.htm
www.architecture.com/go/Architecture/Reference/Library_898.html

Table E.1: Bookmarks set

	PageRank	**HITS**	**SALSA**	**Randomized HITS**	**HubRank**
Total	632	334	334	676	650

Table E.2: HubFinder comparison with different algorithms

Every page in each round has to fulfill some requirements or it will be discarded. In this experiments, pages having a good rank and a minimum out-degree were kept.

As it can be seen Randomized HITS and HubRank are the best ones. While HITS and SALSA provide good hubs but not so important (low authority), PageRank provide only good authorities. Therefore, none of them are desirable for our purpose. Taking a look into the Randomized HITS and HubRank results, depending on the preferences on running the algorithms it would be possible to select any of both. Randomized HITS seems to give more hubs while HubRank give a good amount of hubs (less than Randomized HITS) but it assures that all of them will be good hubs and also good authorities.

Regarding computation time and performance HubFinder is faster than the original algorithms. The reason is simple, as the graph is filtered at each step, a bit more time is taken to select which pages are kept and which ones are discarded but more time is saved in next iterations because of having a smaller set. HubFinder performed up to 33% faster than other algorithms in most of the situations of our experiments.

Index