

Community-Aware Search For Learning Object Repositories

Master Thesis
by
Lars Höppner

August 31, 2008

Submitted to the Faculty of Mathematics and Computer Science
Chair of Data Processing Technology
FernUniversität Hagen

Supervisors:
Dr. Peng Han (FernUniversität Hagen)
Prof. Gerd Kortemeyer, Ph.D. (Michigan State University)
Prof. Dr.-Ing. Bernd J. Krämer (FernUniversität Hagen)

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und nur mit den angegebenen Hilfsmitteln angefertigt habe. Wörtlich oder inhaltlich übernommene Literaturquellen wurden besonders gekennzeichnet.

Glinde, den 31. August 2008

Abstract

The search functionality in content management systems, especially in learning object repositories, is an important aspect for users of such systems who want to (re-)use documents or content. A common technique to support this search functionality is the usage of metadata, particularly of keywords, which are attached to a document to categorize and describe it. The comparison of a query of searched keywords with the keywords of the documents then allow locating possibly relevant documents. Since keywords are only one factor of many which could indicate the relevance of a document, a common problem of this is that the search results often do not match the user's expectations. In this thesis, two aspects of this problem are investigated. The first aspect is that a search query often yields so many search results that the effort to manually determine the value and relevance of each document is too high for a user. The second aspect is the contrary case where too few results are found which might not represent all relevant documents that exist. An often overlooked, but since the emergence of the so-called web 2.0 recognized source of information for the relevance of content, is the usage data of a system itself. Today a plethora of web applications exists which take advantage of social information to provide a better user-experience. This thesis investigates if and how usage data collected in learning object repositories could be used to derive communities of practice. Based on those communities it might be possible to enhance the search in the repository with functionalities that address the problems described above to allow the user to find the most relevant documents for him or her. A prototype search is implemented for a commonly used learning object repository developed at Michigan State University (LON-CAPA, see [Lon]) to allow an evaluation of different alternative algorithms that could help to achieve that goal. Furthermore the challenges for the implementation of the new functionalities are discussed and the results of the first evaluation presented.

Zusammenfassung

Die Suchfunktionalität in Content Management Systemen, insbesondere auch in Repositorien für wiederverwendbare Lernobjekte, ist ein zentraler Aspekt für Benutzer dieser Systeme, die Dokumente oder Inhalte (wieder) verwenden wollen. Eine übliche Technik zur Unterstützung einer solchen Suchfunktionalität ist die Verwendung von Metadaten, vor allem von Schlüsselwörtern, die jedem Dokument zugeordnet werden und dieses klassifizieren und beschreiben sollen. Ein Vergleich einer Suchanfrage mit gesuchten Schlüsselwörtern mit den zu einem Dokument gehörigen Schlüsselwörtern erlaubt dann das Auffinden möglicherweise relevanter Dokumente. Da Schlüsselwörter jedoch nur ein Faktor von vielen sind, die die Relevanz von Dokumenten bedeuten können, ist es in der Praxis häufig ein Problem, dass die Suchergebnisse nicht den Vorstellungen des Suchenden entsprechen. Für die vorliegende Arbeit bezieht sich diese Problematik vor allem auf den Fall, dass mit einer Suchanfrage so viele Dokumente gefunden werden, dass es einen hohen Aufwand verursachen würde, aus dieser großen Zahl die relevantesten Dokumente manuell herauszusuchen. Eine weitere Ausprägung der Problematik ist der gegenteilige Fall, dass nur eine sehr geringe Zahl von Suchergebnissen gefunden wird, die zwar relevant sind, aber möglicherweise nicht alle relevanten Dokumente repräsentieren. Eine in Vergangenheit wenig genutzte, aber spätestens seit dem Aufkommen des sogenannten Web 2.0 als nützlich erkannte Informationsquelle für die Relevanz von Inhalten sind die Nutzungsdaten von Systemen selbst. Es existiert heute eine Vielzahl von Webapplikationen, die die Nutzung von sozialen Informationen ermöglichen und auf diese Weise einen Mehrwert für die Benutzer bereitstellen. In dieser Arbeit wird daher untersucht, inwiefern Nutzungsdaten aus Lernobjekt-Repositorien genutzt werden können, um hieraus Communities of Practice abzuleiten. Aufbauend auf solche Communities ist es dann möglich, die Suche im Repository um Funktionalitäten zu erweitern, die an der oben beschriebenen Problematik ansetzen und dem Nutzer ergänzende Möglichkeiten an die Hand zu geben, die für ihn relevantesten Dokumente zu finden. Im Rahmen dieser Arbeit wird ein Prototyp implementiert, der für verschiedene mögliche Varianten dieser Funktionalitäten eine erste Evaluierung mit Benutzern eines weitverbreiteten an der Michigan State University entwickelten Lernobjektrepositorys (LON-CAPA, siehe [Lon]) erlaubt. Weiterhin werden mögliche Herausforderung bei der Umsetzung dieser Funktionalitäten besprochen sowie die Ergebnisse aus der ersten Auswertung dargestellt.

Acknowledgements

I would like to thank Professor Dr. Bernd Krämer and Dr. Peng Han for introducing me to the interesting research topic and providing valuable feedback during my work on this topic.

I also owe thanks to Professor Gerd Kortemeyer, who supported me in writing this thesis both with technical and organizational aspects, provided a lot of feedback and enabled my stay at Michigan State University and a visit to the annual LON-CAPA conference in Vancouver.

I'd like to thank Professor Wolfgang Bauer, Felicia Berryman and Stuart Raeburn for their ideas, suggestions and help with technical issues.

I'd also like to express my gratitude towards the evaluation participants, who helped to assess the practical use of the concepts developed in my thesis.

Finally, I want to thank my family for their support over the course of my studies.

Contents

1	Introduction	1
1.1	Learning	1
1.2	Reusable Learning Objects	2
1.3	The Purpose of This Thesis	3
1.4	Outline	3
2	Foundations	5
2.1	Reusable Learning Object Repositories	5
2.1.1	Reusable Learning Objects	5
2.1.2	Repositories	8
2.1.3	LON-CAPA	9
2.1.4	CampusContent	14
2.2	Community-Aware Search	15
2.2.1	Keyword-Based Search	15
2.2.2	Communities of Practice	16
2.2.3	Using Communities to Locate Resources	17
2.3	Related Work	18
3	Communities in LON-CAPA	20
3.1	Possible Approaches	20
3.2	Author Communities	22
3.2.1	Strong and Weak Author Communities	22
3.2.2	Taking the Distance Between Users Into Account	24
3.2.3	Choosing Appropriate Author Communities for Course Instructors	27
3.3	Instructor Communities	30
3.3.1	Relationship via the Used Authors	30
3.3.2	Relationship via the Use of Similar Resources	31
3.4	Parameters Influencing the Communities	32
3.5	Communities for New Users	33
3.5.1	Profiles	34
3.5.2	Start Users	35
4	Development of a Prototype	38
4.1	Requirements	38
4.1.1	Description and Target Group	38

4.1.2	Functional Requirements	40
4.1.3	Data Requirements	41
4.1.4	Technical Requirements	42
4.2	Design and Implementation	42
4.2.1	Approach	42
4.2.2	Structural Description	44
4.2.3	Functional Description	44
4.2.4	Data Model	49
4.2.5	Technologies Used	50
4.2.6	Integration and User Interface	50
4.2.7	Result	51
4.3	Additional Aspects	54
4.3.1	Access Rights and Security	54
4.3.2	Integrating Community Elicitation with Search	54
4.3.3	A User-Friendly Interface	55
4.3.4	Managing the Community-Data in a Distributed System	56
4.3.5	Other Possible Improvements	57
5	Evaluation	59
5.1	Objective	59
5.2	Setup and Execution	59
5.3	Results	60
5.4	Comparison	62
5.4.1	Filter Algorithms	62
5.4.2	Extension Algorithms	67
5.5	Discussion	68
6	Conclusion	70
6.1	Summary	70
6.2	Outlook	71
	Bibliography	I
	List of Figures	IV

1 Introduction

1.1 Learning

Learning has always been an important skill in an ever changing and evolving world and has probably gained in importance in the current information age. Learning might be described as the capability of adapting to changing circumstances, adopting new skills or acquiring better knowledge of important topics. It can happen at rather basic levels to ensure the survival of an individual or a species or at more sophisticated levels to further the understanding of complex systems or theories. While there seems to be a big difference in the cognitive effort necessary for both, they have much in common. A way to approach both situations could be to work out a theory (or a behavioral rule) and test it by acting or conducting an experiment. Afterwards one can (hopefully) draw conclusions from the results and change the theory or rule if necessary.¹

With the invention of computers and digital media, new ways of acquiring and managing information came up and also increased the amount of information we all have to deal with in our lives. Additionally to books, analog audio recordings and other traditional means of representing knowledge, now digital texts, images, audio recordings and video files could be used. Combined with networking technology this facilitated ways of distributing information that were much more extensive than ever before. Social software then provided the means to involve all users in a process of mutual teaching and learning.

The World Wide Web itself nowadays is continuously and increasingly being used as an instrument for learning by many users, both in a structured and organized manner (e-learning classes which were specifically designed for this) and simply as an access point to information.

¹Of course, this is a simplified and incomplete description of learning.

1.2 Reusable Learning Objects

Concept

Learning objects arose in the 1990s as a concept to formalize learning contents so that they can be shared and reused easily. Although quite some time has passed since, there is yet no widely accepted definition of what a learning object is. Instead, a variety of definitions exists, which emphasize different aspects of the requirements for didactic contents.

Section 2.1.1 will take a closer look at some of those concepts.

Since the concept of a search functionality is not different for learning objects or other content, the terms "learning object", "content", "document" or, especially in the context of the learning object repository LON-CAPA ([Lon]), "resource" are used interchangeably throughout this thesis.

Repositories

If there is a number of learning objects, the question naturally arises how to organize, manage and store those objects. A common way is to manage learning objects within a learning object repository. Examples of learning object repositories include LON-CAPA, MERLOT ([Mer]), and the repository under development at the DFG Research Center at the FernUniversität in Hagen, Campus Content [Cam].

Search algorithms

Since one important purpose of learning objects is to facilitate reuse, there needs to be a way to find appropriate learning objects for a specific task or situation. A standard method for search in digital libraries or repositories is using keywords to connect search queries with content.

With the emergence of the so-called Web 2.0, a new paradigm gained popularity and importance: search didn't only rely on keywords (possibly in combination with boolean expressions) anymore but instead was enriched with and guided by social aspects. Examples of this are "social bookmarking" or "social tagging". For many applications, taking those social aspects into consideration increased their usefulness for the user considerably.

1.3 The Purpose of This Thesis

This thesis intends to analyze the use of social aspects for the search functionality of reusable learning object repositories (which will subsequently simply be called "repositories" unless the meaning might be ambiguous), derive useful concepts and implement and evaluate them with the help of an active user base of a learning content management system currently in use. The search will be implemented for the LON-CAPA repository, as it has a large amount of learning objects and a big enough and active community to allow a useful evaluation. The evaluation should provide a background for the decision whether a social search functionality can be useful to locate resources in a repository and whether it should be integrated into CampusContent ([Cam]), a learning object repository in development at the FernUniversität Hagen. The experiences with the design and implementation of a prototype for the learning object repository LON-CAPA will also give insight on which challenges such a functionality might pose if it is to be implemented for a production system.

1.4 Outline

Chapter 2 explains the concepts on which the following chapters build.

In section 2.1, the concepts of reusable learning objects and repositories will be explained in more detail. A short overview over how a repository and selecting resources in it are associated will be given with a closer look at LON-CAPA in subsection 2.1.3, since the prototype and evaluation for this thesis will be based on this system.

In section 2.2, the concept of social search will be discussed with a short description of search in general, an introduction of "Communities of Practice" in subsection 2.2.2 and their use to improve the search functionality.

After section 2.3, which briefly discusses related work relevant to this thesis, several ways to identify communities in LON-CAPA are presented in chapter 3. This includes an overview of relationships in section 3.1 which exist in LON-CAPA and can help to derive communities, and the discussion of Author Communities (which were presented in [HKKvP08]) in section 3.2 as well as

other ways of establishing communities in section 3.3 and additional aspects which are relevant in the context of these communities.

The communities discussed in chapter 3 form the basis for the development of a prototype of a community-aware search in LON-CAPA, which is documented in chapter 4.

Section 4.1 outlines the requirements and in section 4.2 the search is constructed, designed and implemented.

Section 4.3 mentions additional aspects of using communities to improve search in the context of distributed repositories and other technical and organizational considerations which are not addressed with the concepts in this thesis and the prototype developed for complexity reasons.

Chapter 5 reports on efforts to evaluate the community-aware search and to determine which of the approaches are most useful. One of the goals of the evaluation of the community-aware search is to draw conclusions concerning possible further developments in both LON-CAPA and CampusContent.

This evaluation is set up to be based on experiences by current users of LON-CAPA with the new search features as well as a side-by-side comparison of the different algorithms, so that practical results can be expected.

Chapter 6 summarizes the progress and results and attempts to derive conclusions to guide the further use of social search functionality. It will also report on possible areas of future work and ideas which could not be implemented in this thesis.

2 Foundations

2.1 Reusable Learning Object Repositories

2.1.1 Reusable Learning Objects

This chapter discusses several concepts and definitions of "learning objects", with a focus on the one used for LON-CAPA.

Although quite some time has passed since the first mention of the concept, there is yet no widely accepted definition of what a learning object is. Instead, a variety of definitions exist that emphasize different aspects of the requirements for learning media.

The IEEE defines a learning object in their final draft standard "Learning Object Metadata" as "any entity - digital or non-digital - that may be used for learning, education or training" ([Lea02]). Of course, this definition is very vague - virtually anything can conform to this definition - and the definition can therefore not be considered sufficient and adequate for most situations.

In [Krä05], learning objects are introduced in the following way:

"The concept of learning objects arose in the late nineties driven by the motivation to reduce the development and maintenance cost of digital learning content by means of modularization and reusability. Learning objects promised to offer a new way to create and mediate educational content in terms of smaller units of learning that are self-contained, can be re-used in multiple contexts and pedagogic settings and can be grouped into coherent collections of digital learning content."

This description is more specific and therefore useful. It restricts learning objects to digital ones and emphasizes the aspect of reusability in different

contexts. A similar definition is provided by D.A. Wiley in [Wil02]. Learning objects are there defined as "any digital resource that can be reused to support learning". The possibility for instructors to create small components of educational content that can then be made use of many times certainly is one of the biggest appeals for learning objects. Other advantages compared to traditional instructional media are the possibility to collaborate to create or change learning objects and the ease of sharing those objects over a digital network.

Since the concepts in this thesis will be based on LON-CAPA, we also need to know what comprises a learning object there. In LON-CAPA, different levels of granularity exist:

- "fragment level resources", which could be images, movies, applets, and other file types, or "problems", which are the LON-CAPA representation of exercises that can be used as homework problems or, since they have many randomization features to provide individual problem-instances per user, even in exams
- a number of those fragment level resources assembled together are called a "page"
- a number of pages (or individual resources) linked together in a sequence are called "sequence" (also "lesson" or "chapter")
- several sequences linked together are called a "course" (a course can also contain single pages or resources, however)

Figure 2.1 (taken with permission from [KAB⁺03]) illustrates how the elements of different levels could be integrated into a course.

The motivation to create the concept of learning objects was to reduce the effort (and thereby the costs) of creating and maintaining learning resources by reusing existing ones. The idea has parallels to software development, where modularization and reuse have been rather successful, especially in the domain of object oriented software development.

Since the IEEE definition is very general, it doesn't give any directive on how one could take advantage of reuse.

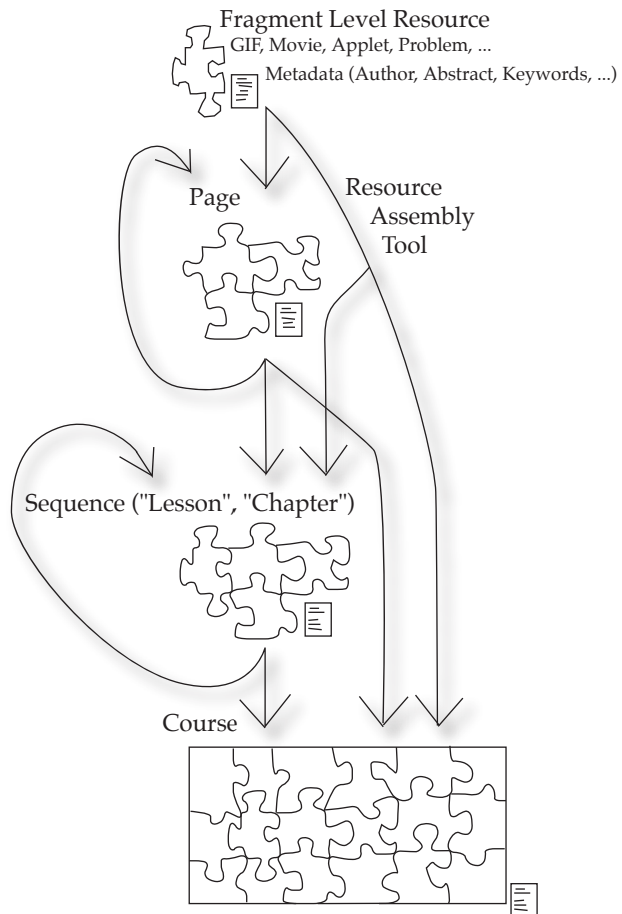


Figure 2.1: Resources (learning objects) and resource assembly in LON-CAPA

The description in [Krä05] specifically takes reuse into consideration as well as the one in [Wil02].

To provide a structured categorization of resources, learning objects are often annotated with metadata, that is, data about data. Examples for metadata include a title, keywords, information about the author, the creation date and many others. The metadata used for resources in LON-CAPA are described in section 2.1.3.

2.1.2 Repositories

To enable the reuse of learning objects, it is necessary to have access to them. There also needs to be a way of finding appropriate resources for specific learning objectives and courses.

Learning objects can be managed by repositories that provide basic functionalities to store, change and retrieve those learning objects and might also provide a versioning and archiving system, handle the distribution of objects across a network and more.

Depending on the availability, number and type of metadata provided with the learning objects, there are several ways to locate resources.

A simple way would be to access resources via an URL that is valid in the repository. For example, one resource author could simply tell another of a certain resource he created that might be of use to the other author. This is more or less the same as sharing which happened in the pre-networked/internet era and is a quite inefficient and impractical approach for large repositories, since it requires the participants to know each other and also know their respective fields of interest at any given time. Also, URLs are usually long and cumbersome to handle. However, the possibility to give recommendations is still a desirable feature as it is a natural way of information propagation. Since the resources are not really located with means of the repository, this cannot count as a property of the repository.

A way in which the repository could provide a functionality to locate resources would be a browsing functionality based on hierarchical structures, so that one could access data provided by a certain institution, group or author one trusts.

This approach has similar drawbacks, since one needs to know or guess which author might have the best material for a certain topic.

Those two methods don't use metadata and parallel search and browsing features in the World Wide Web which are primarily based on explicit links between documents. A third method makes use of metadata and has currently no directly corresponding counterpart in the World Wide Web, although efforts towards a "semantic web" are trying to change that.

As described in section 2.1.1, learning objects can have metadata to characterize them. Those metadata and what can be derived from them can be used to locate resources, e.g. via a keyword-based search functionality.

There could also be a variety of recommendation-based methods which are in use on many e-commerce websites today, or community-based methods to retrieve interesting resources.

This description of means to locate resources is certainly not complete, but provides a sufficient basis for further chapters of this thesis.

Overall, one would like to have as many means as possible of locating resources available in a repository while keeping the user interface as simple as possible.

2.1.3 LON-CAPA

History and Features

LON-CAPA (LearningOnline Network with a Computer Assisted Personalized Approach, [Lon]) is a web-based content authoring and management system designed for online learning and assessment.

It evolved out of two existing systems: LectureOnline, a learning content management system that was started in 1997 and CAPA (a Computer-Assisted Personalized Approach) that was started in 1992 at Michigan State University to provide homework for an introductory physics course. Since those homework problems can often be useful for different lecturers, different institutions and lecturers started exchanging problem libraries between each other. This was, however, not supported by a dedicated infrastructure, so that the users had to manually export and import those problems into their local installation. Therefore, increasing the ease of resource sharing and thereby supporting and

encouraging reuse of materials was one important incentive for the development of LON-CAPA ([KKBB08]).

Among the features currently offered by LON-CAPA are ([KKBB08]):

- cross-institutional load balancing
- localization for several languages
- creating randomized problems
- an automatic grading system
- immediate feedback to and performance statistics for students
- bulletin boards and user discussions
- printing out randomized exams and uploading grades from files

Architecture

The LearningOnline Network with CAPA works on top of a geographically distributed network of servers at the participating institutions (e.g. universities, colleges, schools). Every institution can run one or several servers in its own domain to allow for load balancing and scalability according to the size of their user base (users typically log into a server at their local institution to work with LON-CAPA - although they can log in into servers anywhere). Figure 2.2 illustrates the LON-CAPA network architecture. Two different types of servers are used: library servers (labeled A and E in the figure) which store personal records of users, handle the initial login and store resources (for authors within its domain it stores the current authoritative version as well as previous versions of the resources they published). Access servers (labeled C, D, I and J in the figure) handle the sessions for the users, which makes the system scalable so that it can handle more concurrent user sessions.

Any system with a web browser can serve as a client to log in to LON-CAPA.

Resources are stored locally in the domain of the author. Institutions can restrict the availability of their provided content to their domain or make it available over the complete network - every user in the network can then access the content via its unique URL in the system. To allow for faster access to the resources, the network also provides transparent resource replication.

When resources are accessed from sessions on an access server, the resource is replicated and stored as a local copy on the access server. These resources can then be accessed later without additional network traffic. The library server keeps a subscription list for each resource, so that when the resource is changed, the access servers are notified to update to the current version (the course instructors can still choose to keep using the older version though).

Resources and Resource Assembly

Resources in LON-CAPA can have the form of multimedia files like images, videos or sound files, web pages, applets, problems and a (possibly hierarchical) combination of all these. Problems can be used to provide randomized exercises. There are different types of problems for which templates are provided within the content assembly tool, e.g. simple multiple-choice problems, math problems that use a computer algebra system, click-on-image problems that require the user to correctly identify and click on certain areas of an image, or chemical problems that might require the user to input chemical formulas or draw a molecule. The problems are stored XML-formatted.

Usually those resources are not used independently, but in conjunction with other resources. There are different ways of assembling resources into an educational course. LON-CAPA offers a course management tool which facilitates the course composition. Usually, a course will consist of a syllabus, introductory contents, the main content, homework problems and possibly exams. The different resources can be structured hierarchically using folders. To add the contents, it is possible to import existing resources from the repository, or create a new resource from prepared templates like the ones mentioned above. It is also possible to upload documents which don't exist in the repository from an external source or a locally available documents (e.g. on the hard drive) for a course. The course contents can then be arranged in a preferred order. Apart from creating the content, a number of options can be used to manage how the course can then be used. For example, it is possible to enroll students in the course, manage the course users, set open and due dates for homework problems, influence whether discussions are allowed for certain problems and add similar organizational information that is useful to handle the contents.

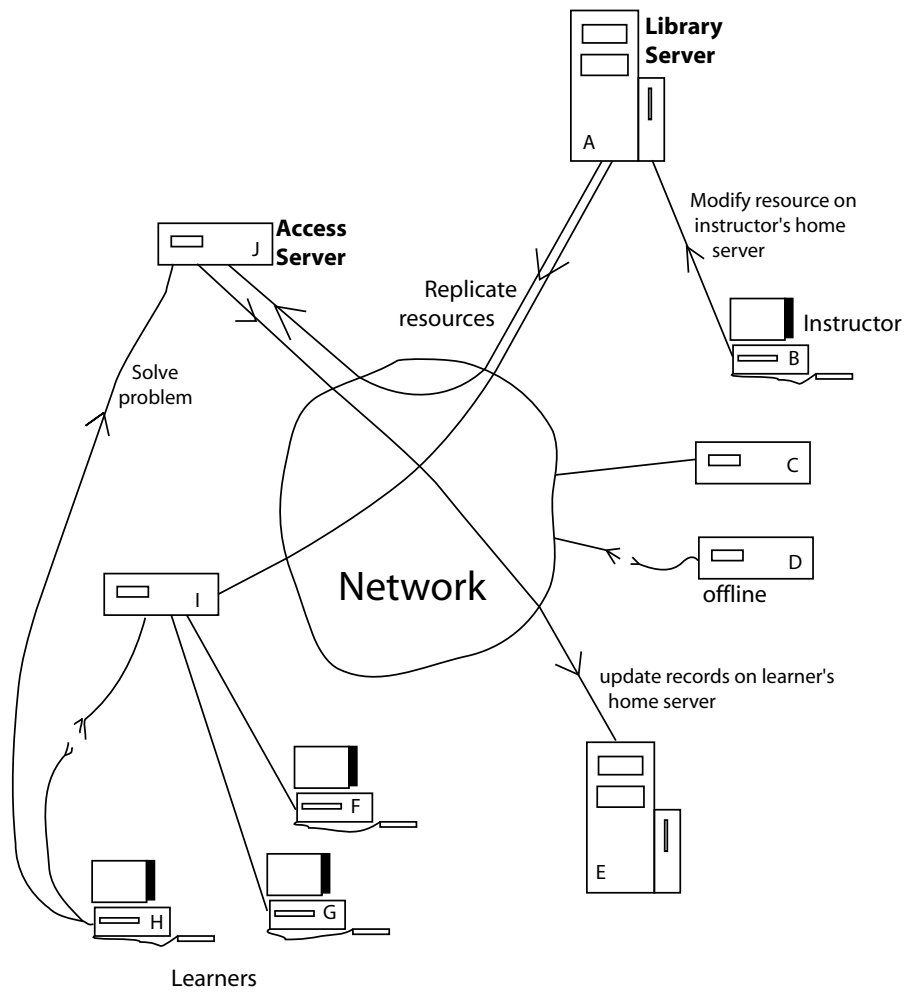


Figure 2.2: Network Architecture of LON-CAPA

Resource Metadata

Every resource is associated with a metadata file that contains static data (like title and author of the resource) as well as dynamically updated information about the usage of this resource within the system (e.g. how many times a resource has been accessed and in which courses a resource is being used).

Every time a resource is used in a course, this is recorded in the corresponding resource metadata file. This information is essential to establish communities as described in chapter 3.

The implicit relationships contained in such metadata might be used to improve the search functionality in learning object repositories, which will be investigated in this thesis for LON-CAPA. Since this data is generated dynamically, it has the advantage that it is a more reliably available source of information compared to metadata like keywords which usually have to be entered manually by the authors. Experience has shown that many users simply don't want to do that and skip entering keywords. Even if a user is willing to enter information like keywords, it is often difficult to find a good set of words that exactly represents the resources they are to characterize, especially if the user doesn't know and understand the concept and proper use of keywords.

Resource Selection

LON-CAPA currently offers two different methods to locate resources: keyword-based search and a browse-functionality that allows users to navigate through the hierarchy of domains (institutions) and their authors. Since LON-CAPA is a distributed system, a locally initiated search query is propagated to all domains which in turn search for results and report back to the server the user is currently logged in at. Resources can also be directly accessed by their URL.

Content Sharing

[KKBB08] reports on the current state of the LON-CAPA network and its contents: Today, over 120 institutions use LON-CAPA (most of which are universities and high schools). The repository provides over 300,000 resources,

including over 100,000 randomizing online problems, over 100,000 images and over 50,000 web pages.

Most resources were contributed by faculty members of the participating institutions, although many were also created in externally funded projects. The majority of resources created within LON-CAPA are published and available system-wide. Additionally, there are commercial publishing companies, e.g. textbook publishers who provide exercise problems accompanying their textbooks.

As already mentioned, data about the usage of each resource is recorded in its metadata. This includes how often users access a resource, if and where a resource is integrated into a composite learning object, and especially in which courses a resource is used.

In 2006, about 36 percent of all resources available were in active use (i.e. recently used in at least one course). About 44 percent of these were used across institutional boundaries, i.e. in at least one course at an institution different to the one that provided the resource. The percentage of active use and cross-institutional sharing is the highest for problem-type resources ([Kor06]).

2.1.4 CampusContent

CampusContent ([Cam]) is a learning object repository currently being developed at a competence center for e-learning at the FernUniversität Hagen funded by the German Research Foundation.

It aims to provide a learning content platform which supports sharing and reuse of learning materials combined with pedagogical scenarios. It consists of a distributed content repository and a portal that facilitates content creation, e.g. to design exercises.

Its distributed architecture enables participating institutions to host their content on their own server and to define rules for their accessibility within the network (for a more detailed description of the architecture see [BN06]).

By means of the underlying content management system it also uses social networking functionality ([KZ08]). The results and conclusions of this thesis

should serve as background on if and how social information could also be used to implement an efficient search functionality.

2.2 Community-Aware Search

2.2.1 Keyword-Based Search

The principle of keyword search is simple and widely in use: the user types in terms that describe the contents he or she is interested in and these keyword terms are then compared to either the complete content of a document or resource, or to metadata that describe the contents if the resources are annotated.

The current search approach in LON-CAPA is keyword-based. Because of the distributed architecture of LON-CAPA, the resources are distributed over a network, i.e. no server has information about all resources available. When a search term is entered and executed by the search module, it sends this query to every library server in the network. The servers then each perform a local search and report the results back to the server that issued the search.

In theory, keyword-based search should allow users to find exactly what they are looking for. Experience has shown though, that it often yields search results that are not satisfactory, e.g. a common case would be that a search yields a large number of results which vary in relevance to the user who now has to manually analyze and evaluate the results to find the best matches.

Reasons for this include:

- Resources have to be categorized, i.e. every resource has to be translated into a number of keywords that accurately describe the resource. Those keywords have to be general enough so that users will be able to find it, yet special enough so that the resource isn't found if it isn't relevant. This is a rather hard task even if done manually, and is often done automatically for text resources, e.g. by analyzing it to find words that appear often. This means the keywords are often of a rather low quality.
- A user trying to locate a resource has to translate his idea of the resource he is looking for into a set of keywords that represents the resource.

Most users are no information-retrieval experts, so the formulation of an effective query is difficult for those users.

- Since different people often use different terminology, they will expect different keywords for the same resource, so that existing relevant resources may be overlooked. Text-analyzing algorithms might annotate resources with keywords that happen to appear often in the resource, yet they may not be an accurate description of its content, which adds to the same problem.
- Non-text resources (e.g. images or videos) are hard to categorize automatically and thus often cannot be found if they aren't categorized manually.

To overcome those problems, other sources of information about relevant resources have to be considered additionally.

Social interaction has always been a valuable source of information when making decisions and learning new things. Since the world today is becoming more and more complex, there increasingly is the need to learn socially, i.e. by observation of the behavior of other people. The decisions of other people then give guidance as to which option to choose.

Since documents and resources are provided and utilized by (human) users, it is a logical step to apply the concept of social learning to the process of resource selection in a learning object repository.

Therefore, the following chapter will look at ways to use social information in the context of reusable learning object repositories to improve the search functionality.

2.2.2 Communities of Practice

The term "community of practice" (the shorter term "community" will often be used in the same sense from here on) was coined by Etienne Wenger and Jean Lave in the early 1990s. It describes the concept of "groups of people who share a concern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in this area by interacting on an ongoing basis" ([WMS02], p. 4).

A slightly more detailed definition of communities of practice can be found in [HNV01]: "Groups of people who come together to share and to learn from one another face-to-face and virtually. They are held together by a common interest in a body of knowledge and are driven by a desire and need to share problems, experiences, insights, templates, tools and best practices."

Those groups of people are, in our context, the course authors and lecturers who are interested in the same (or related) topics under the common theme of creating and using resources for teaching about those topics.

The goal of this thesis is to evaluate the use of information about communities to improve the search capability within reusable learning object repositories. To do so, the notion of a community has to be applied to such repositories. This will be discussed in the next chapter for the LON-CAPA system.

2.2.3 Using Communities to Locate Resources

Communities can help to improve search and browsing functionalities in various ways. There could be a recommendation system which notifies a user of interesting resources. This might be based on a comparison between the behavior of different users, e.g. by means of Collaborative Filtering ([BHK98]). Collaborative Filtering takes advantage of the fact that people often base their decisions on their knowledge about how others have decided in a situation. Using the same information, an algorithm can recommend resources another user, who is otherwise similar to you regarding the reuse behavior (i.e. who used the same or related resources), used.

Additionally, recommendations could be made during the course composition phase based on which resources are already present in the course and which resources have been used by other course instructors together with those.

Recommendations could also be derived from new resources which were published in a user's community.

It could be possible to browse a community, e.g. by their members or the keywords the community as a whole is most interested in. Basically, this would allow users to traverse the network spanned by the users and their resources as illustrated in section 3.1.

Another approach would be to use communities to filter and extend the search results of simple keyword queries, thus introducing an additional aspect to influence the effectiveness of search (this has been proposed in [HKKvP08]).

This thesis focuses on community-based filtering and extending of keyword search. The aspect of collaborative filtering will be taken into account as one way to derive communities, which will be discussed in chapter 3.

2.3 Related Work

The work in this thesis is based on the findings of Han et al. in [HKKvP08], where the authors established the concept of Author Communities that can be derived from the user network graph and represent groups of people who work in similar areas of interest. This will be discussed in more detail in subsection 3.2.

In [AA04], the authors present a ranking technique for search engines that takes interest-based communities into account. The communities are identified based on similarities between user sessions (which represent user interaction with an information retrieval service). Bayesian belief networks are used to model the relationships between search queries, objects in the search space, and communities. The communities provide a context for search queries so that a standard ranking technique can be combined with an interest ranking based on the relationships between communities and objects in the search space.

In [FS04], the authors present an architecture for a social search which keeps records of user interaction with the search functionality and uses this information to re-rank the search results compared to the standard ranking. The records are separated into individual communities so that the ranking can be tailored for a specific user issuing the search query.

In [FFB⁺07], the authors also discuss using communities to allow social browsing which combines the hierarchical content selection with information based on previous user behavior.

All of the approaches mentioned in the previous three paragraphs are related to the attempts in this thesis in that they use communities derived from records of user interactions to alter the behavior of a search functionality. In [AA04]

and [FS04], communities are only used to rank the results, i.e. change their order which should represent their relevance, while the goal in this thesis is to provide filtering and extension mechanisms to adapt the result set for a query, i.e. only to show the most relevant results or to find additional results. In a certain sense, filtering the result set also provides a ranking though; the resources which were filtered out could e.g. instead be kept in the result set with a lower ranking than the relevant resources. The approach in this thesis differs by an additional extension functionality, which allows users to extend a result set by other relevant results.

In [MGD06], the authors discussed integrating internet search and social networks to improve the content selection process. Based on a first evaluation, they concluded that communities might have the potential to improve internet search. Of course, internet search is different to the more special search for contents in learning object repositories. The authors also provided interesting background though on why and how communities help in the search process: since search terms often have different meaning depending on the search context, communities give an indication which meaning is relevant to a specific user. This is indirectly also used by the community-aware search in this thesis.

This is only an excerpt of many publications which shows that there is a high interest in the combination of social information and the search process in digital libraries or other content repositories.

3 Communities in LON-CAPA

3.1 Possible Approaches

Existing approaches which combine social network data with information retrieval often suffer from the problem that the users need to manually set up their communities by either entering a certain profile or explicitly choosing users they would like to collaborate with. Since the typical user wants to spend as few time as possible when using a system, it would be desirable to have a way of identifying communities without requiring the active participation of the users.

There are many implicit relationships between authors, resources, sequences, courses and course coordinators in LON-CAPA, which all might provide indicators which resources a particular course coordinator might be interested in.

Figure 3.1 might give an idea of how complex these relationships can become even for a small number of users and resources. For example, user A can here be linked with user E, since both users contributed resources (as part of sequences) which are then used by users C and G for a course. There is also a relationship between users C, G and F, since they all used sequence C for their courses.

One way of deriving communities from the above mentioned relationships is discussed in section 3.2. It is based on co-author relationship as presented in [HKKvP08].

This is illustrated by figure 3.2 in which a scenario with the relationship between two authors ("Isaac Newton" and "Archimedes") is shown, as well as the relationship between those authors and a course coordinator. Here, "Isaac Newton" and "Archimedes" constitute co-authors in the sense of the idea in

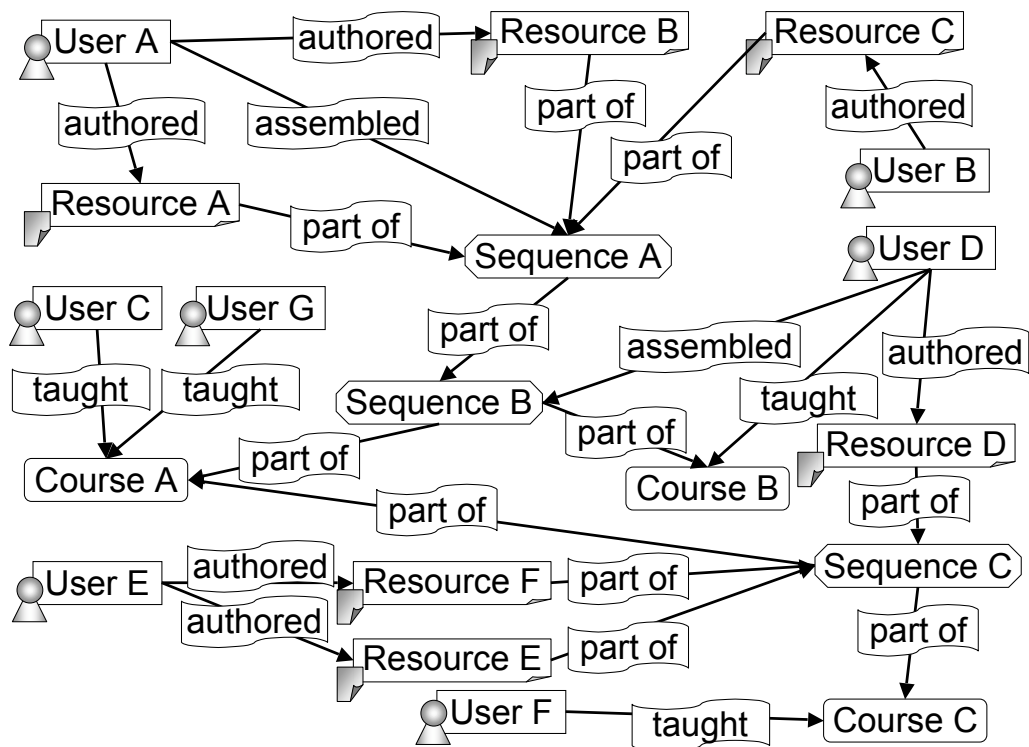


Figure 3.1: Relations in LON-CAPA

[HKKvP08], since both have provided resources which were then used together in a course context by user "Gottfried Wilhelm Leibniz".

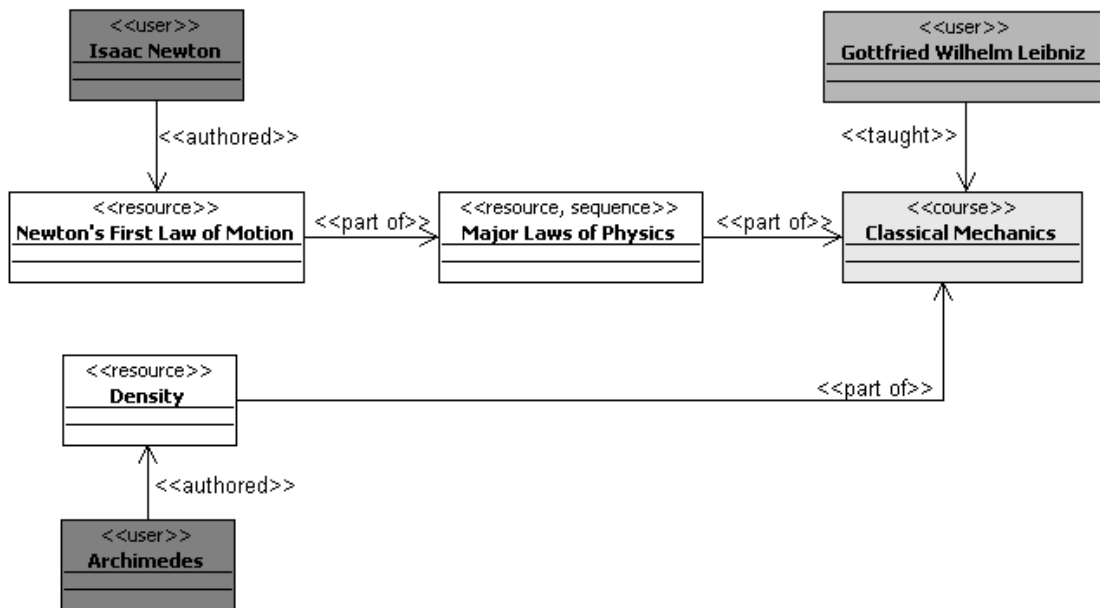


Figure 3.2: Identifying Strong Author Communities (SAC)

In social networks the idea of "degrees of separation" between users has been discussed by various publications. This concept is also applicable in our context. Here, the connection between two entities is not simply "to know the other" but "to have used a resource of another author" or "to have used similar resources as another course instructor". This idea will be discussed in section 3.3.

A look at figure 3.1 indicates that there are more possibilities to use the relations between users in their roles as authors or course instructors and the resources to establish communities.

3.2 Author Communities

3.2.1 Strong and Weak Author Communities

One way to use the information discussed in the previous section arises by using the co-author relationship as defined in [HKKvP08]. If two authors each

contributed a resource both of which are used together in a certain course, one might interpret this as a common interest of both authors in the topic taught with the course. Thus there is a certain likelihood that new resources created by those users might again fit into a similar context.

This concept has been presented in [HKKvP08]. The authors introduced so-called "Strong Author Communities" and "Weak Author Communities" based on user network graphs where the users are represented by nodes and their relationship is represented by an edge between two nodes which is labeled with a weight according to the number of times both users' resources have been used together in the same course. Figure 3.3 shows an example where nodes 1 to 4 form a Strong Author Community with a connectivity of (at least) 6 (since they are all directly connected to each other with a connectivity higher than 5) and nodes 1 to 9 form a weak author community of connectivity 6 (since they are all connected to each other, but some not directly). Node 10 is not in those two Author Communities because he contributed to courses together with User 1 not more than 5 times.

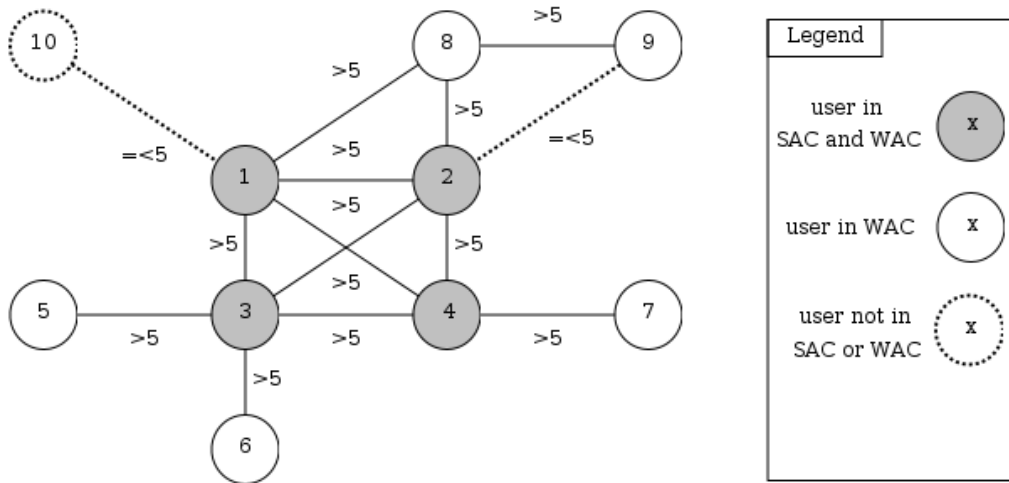


Figure 3.3: Author Communities in the User Network Graph

The relationship between those users is based on a "Co-Contribution Association" (CCA) which is based on the number of times both users' resources have been used together in the same course. More formally, the CCA between two users a_i and a_j is defined as:

$$CCA(a_i, a_j) = |\{C | \exists a, b \in C, a \in R_{a_i} \wedge b \in R_{a_j}\}|,$$

where C is a course (represented by the set of resources contained in the course) and R_{a_i} and R_{a_j} denote the sets of resources contributed to the system by author a_i and author a_j ([HKKvP08]).

A Strong Author Community (SAC) respectively a Weak Author Community (WAC) with a connectivity n is then defined in [HKKvP08] as:

$$SAC_n = \{A | \forall a_i, a_j \in A, CCA(a_i, a_j) \geq n\}$$

$$WAC_n = \{A | \forall a_i \in A, \exists a_j \in A, a_i \neq a_j \wedge CCA(a_i, a_j) \geq n\}.$$

3.2.2 Taking the Distance Between Users Into Account

A closer look at the user network spanned by LON-CAPA users revealed though, that the Weak Author Communities appear to provide an insufficient way to dissect the network into separate communities. In fact, for most connectivities, only one Weak Author Community can be found in the complete LON-CAPA user network.

This rather unusual result is probably due to the phenomenon that many LON-CAPA users manually establish something similar to an author community by creating an account that is in fact used by many users to provide their resources. This allows different authors to work collaboratively on resources and also forms a pool of resources in a certain area of interest (e.g., there is an account that is used to provide mainly physics related resources). Since resources provided by those collective users are then used in the courses taught by the individual users which are likely to also contain other resources authored by users with their individual user account, the collective users act as hubs connecting all users by the co-author relation who use this collective user to provide resources.

For that reason the concept of the Weak Author Community is modified in this thesis to take the distance between users (i.e. the number of edges between two user nodes in the network) into account. This is illustrated in figure 3.4 for a Weak Author Community found in LON-CAPA with a connectivity of 250. All edges that are unlabeled represent a connectivity higher than 250. To compute a distance, a reference point is needed, therefore these Weak Author Communities are centered around a specific user who is represented by the

node labeled 2673 in this case¹. The nodes 804, 828, 1754, 2302, 2673, 884 and 687 form a Weak Author Community if all related users are taken into consideration that have a distance to 2673 of one (this will subsequently also be called a Weak Author Community with a degree of 1 around user 2673). With a distance of two, the nodes 2180, 666, 2918, 278 and 2888 are now too a part of that Weak Author Community. The nodes 998, 1766 and 944 are not part of the community since they aren't connected to other nodes in the community with a high enough connectivity.

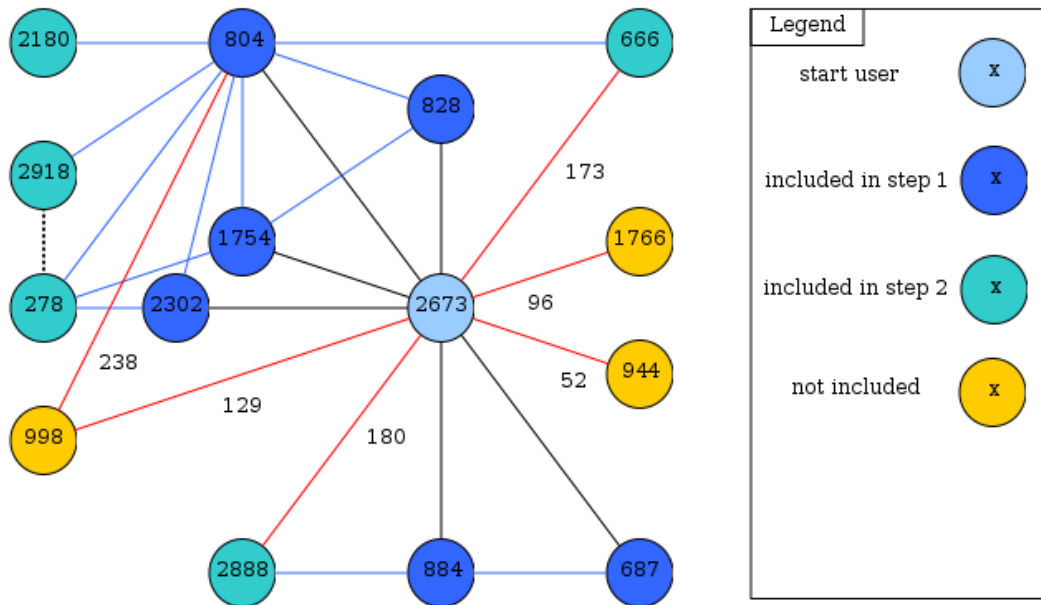


Figure 3.4: Development of a Weak Author Community taking distances between users into account

Table 3.1 shows how the number of communities found and the number of community members varies for different degrees and connectivities (where the degree is defined by the distance). The possible maximum number of members (last column) is determined by all users that could be reached with the community without taking the distance into account (i.e. this number represents the member count of the corresponding Weak Author Community as described in [HKKvP08]).

As expected, the number of communities found decreases with higher connec-

¹Of course, a Weak Author Community consisting of the exact same members might be computed for different users at the center

Degree	Connectivity	Number of Communities	Average Member Count ²	Possible Maximum Members
1	5	395	59.5	398
2	5	278	344.3	398
3	5	15	360.7	398
1	10	270	41.6	272
2	10	120	229.8	272
3	10	11	240.7	272
1	20	184	25.2	185
2	20	70	155.0	185
3	20	6	182.5	185
1	50	86	12.8	86
2	50	15	63.0	86
3	50	3	84.0	86
1	70	63	9.2	63
2	70	16	43.9	63
3	70	3	62.3	63
1	100	34	7.8	34
2	100	9	25.3	34
3	100	1	34	34

Table 3.1: Weak Author Community member count and average size depending on the distance (degree) and connectivity: the average member count of all found communities indicates that using a degree of 1 leads to the best distribution of members on different communities; a degree of 2 or 3 leads to a high amount of overlap between the communities, the average member count then approaches the maximum member count

tivities. More interestingly, the number of communities also decreases significantly with degrees above one.

Column 3 shows that for a degree of two, the average member count is already high compared to the possible maximum. With a degree of three, the average is very close to the maximum and there are only few distinct communities, which indicates that the user network graph defined by the CCA strongly connects many of its users. This means that nearly all users that could be reached are reached in only three steps from any user within that community. Apparently, these users are very well connected to each other. This, again, is probably caused by the collective users mentioned in the last section.

These conclusions are also illustrated in the diagrams of figure 3.5 which are based on the data of table 3.1.

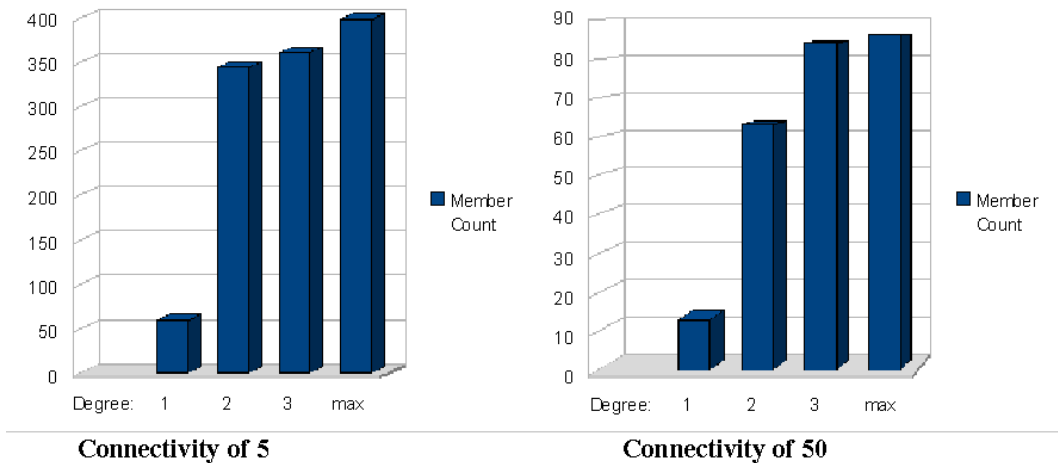


Figure 3.5: Weak Author Community member count and average size depending on the distance (degree) and connectivity: the diagrams illustrate that for a degree of 2 or 3, the average community size is close to the possible maximum size, which indicates a high amount of overlap between those communities and probably reduces their usefulness

With the concept of author communities we have created groups of users that create a material in a certain area of interest. Next, the question arises which of those groups are relevant to a course coordinator who searches for resources for his or her courses.

3.2.3 Choosing Appropriate Author Communities for Course Instructors

One approach to find appropriate communities for course instructors would be to create profiles for users and communities as described in [HKKvP08] and establish a proximity function to compute similar communities.

Based on countable sets $S = \{s_1, s_2, \dots, s_n\}$ of subjects and $K = \{k_1, k_2, \dots, k_m\}$ of keywords used in LON-CAPA, subject profiles $S(C_i)$ and a keyword profiles $K(C_i)$ are defined for a course C_i in the following way:

$$S(C_i) = \langle C_{is1}, C_{is2}, \dots, C_{isn} \rangle$$

$$K(C_i) = \langle C_{ik1}, C_{ik2}, \dots, C_{ikm} \rangle$$

where C_{is_j} denotes the frequency of subject term s_j and C_{ik_j} denotes the frequency of keyword term k_j in the metadata of resources in the course ([HKKvP08]). In a similar way, the frequencies of subjects and keywords in the metadata of resources authored by a specific user could be used to create profiles for a resource author and, by extension, to create profiles for communities. These profiles then describe around which subject area a certain community provides resources.

Users in their role as course instructors (who are usually those searching for interesting resources), could then provide an interest profile themselves, e.g. by selecting predefined subjects and keywords that they feel match their interests. This would enable the system to compute the closest communities and use this information for the search enhancement.

However, using subject and keyword profiles to assign course instructors to communities has the drawback that it relies on the quality of subject and keyword terms used, which is negatively influenced by the following factors:

- a closer look at the metadata for LON-CAPA resources shows that the subjects and keywords are often not consistently and adequately chosen; since there is no controlled vocabulary and hierarchy for subjects and keywords, users have to make up their own terms and hierarchy³, which leads to a variety of different terms for the same thing; users are also not always clear on what constitutes a keyword and what a subject, which leads to keyword terms which might be better suited as subject terms and vice versa (defining a taxonomy for subjects and keywords using existing lexical databases like WordNet⁴ might be an alternative approach which is discussed in subsection 3.5.1)
- currently, there is no way to distinguish composite terms from single keywords, e.g. the subject "Introductory Physics, Newton's Laws of Motion" might be a good description of the content of a certain sequence for a human reader, there is no information about which words are connected to which so that one could analyze this automatically (and if

³which is, on the other hand, desirable, since building and maintaining an externally controlled vocabulary and hierarchy would be either very time-consuming and controversial, or it wouldn't be able to adequately represent the necessary terminology desired by the different interest communities

⁴<http://wordnet.princeton.edu/>

that information existed, it could again increase the likelihood of a lot of different composite terms used for the same content)

- since the keywords and subjects are not directly relevant and of use to the author, creating them might be perceived as an unnecessary and onerous duty which might lead to many authors simply not providing these metadata (or doing so poorly) - indeed many resources in LON-CAPA don't have keywords or subjects at all

In conclusion, the use of subject and keyword profiles without a previous revision of subject and keyword terms regarding the mentioned problems (which would lead to a considerable amount of additional work and would enforce a taxonomy that not all users might be able to agree on) doesn't appear to be a practical solution at least in the context of this thesis.

For this reason, this thesis will rely on another approach that focuses on dynamic data, specifically the usage of resources by course instructors.

When an instructor uses a certain resource for his or her course(s), this clearly also gives insight into the areas of interest for that instructor. This leads to another way of assigning instructors to an appropriate community. The implicit assumption that authors who provided interesting resources have a higher than average likelihood of providing more of those interesting resources⁵, which has been used for the concept of author communities, also suggests the following strategy to assign a community for an instructor:

- determine how often resources which were provided by the community members were used by the instructor; let C be that number
- the community with the highest count C will be assigned as the instructor's community of interest

One problem with this approach is that new users who haven't used any resources yet cannot be assigned to communities. For those users, the possibility to create a profile based on a set of keywords that could be matched with community profiles as described in [HKKvP08] could be provided additionally.

⁵from the perspective of a certain user

As soon as the user reuses a resource, he or she can then be assigned to a number of candidate communities. Of course, these might not entirely represent relevant communities for this user. The more resources a user uses in his role as instructor, the better his communities will match his actual areas of interest.

The author communities as described in this section can be used to extend the number of search results for a given query by adding resources that are interesting to the searching user, but haven't been annotated well with meta-data.

It is also possible to filter a result list by author communities in several ways. This will be discussed in subsection 4.2.3.

3.3 Instructor Communities

3.3.1 Relationship via the Used Authors

This and the next subsection discuss another way of identifying relevant relationships between users and building communities on top of them.

The first one is based on information about which authors' resources were used by a course instructor. These authors then can be said to be of higher interest to the instructor compared to other authors. The used authors in turn have also used, in their roles as course instructor, certain other authors' resources. There might be a certain probability that those authors then also are of interest to the original instructor. Similar to the modification of the Weak Author Communities described in 3.2, this kind of relationship defines a community graph depending on a distance around a user.

Both approaches in this section rely on a user's recorded behavior to establish a community for the user.

The idea is illustrated in figure 3.6. User 0 here has used material authored by user 1 and 2. User 1 has used material authored by user 3 and 4, and so on. So, for a degree of one, a community based on this relationship would consist of the users 0, 1 and 2. For a degree of two, it would consist of 0, 1, 2, 3 and

4. For a degree of three, the community would also include user 5 and, for a degree of four, user 6.

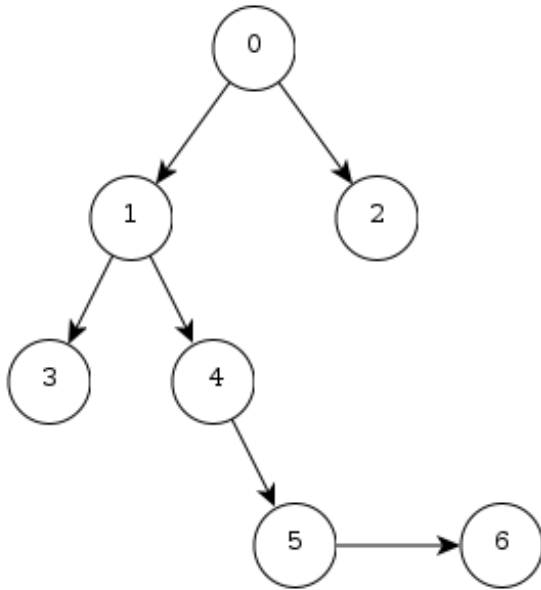


Figure 3.6: Degrees of separation in learning object repositories

3.3.2 Relationship via the Use of Similar Resources

Similar to the method discussed in the previous subsection, it is possible to establish communities based on the similarity of reuse-profiles. The reuse-profile of a user here is defined by the different resources an instructor has used in his courses. A similarity based on the ratio of how often two authors both used the same resource in their courses to the total number of resources used by both authors combined then could be computed according to the following formula:

$$Sim(a_i, a_j) = \frac{|\{r|r \in R_{a_i} \wedge r \in R_{a_j}\}|}{|R_{a_i}| + |R_{a_j}|}$$

where R_{a_i} is the set of resources used by user a_i for his or her courses and R_{a_j} the set of resources used by user a_j for his or her courses.

This similarity then could also be used as the relationship in the community graph as described in subsection 3.3.1 with illustration 3.6.

3.4 Parameters Influencing the Communities

The communities discussed in sections 3.2 and 3.3 have different parameters which affect who is included in a community and how many members form a community.

For Author Communities, the first determinant is the "connectivity", that is, how often were two authors' resources used together as part of a course. For the Weak Author Communities, there is the additional parameter "distance" (also called "degree of separation" in the user network graph).

The distance parameter is also effective for both communities described in section 3.3.

The second parameter for the communities which are based on the relationship of the used authors is the minimum number of reuse instances for a user to be considered, i.e. to be included into the community. If, for example, the instructor used resources from a user five times, but the minimum number of reuse instances is six, this user (and any other user possibly related to that user) is not included in the community.

The second parameter for the communities which are based on the relationship of a similar reuse-profile is the minimum similarity (as defined in subsection 3.3.2) for a user to be considered.

Since the size and composition of the communities established by the different relationships varies a lot depending on the choice of their respective parameters, the question arises which values to choose that will yield a useful dissection of the system users into communities with common interests.

Table 3.2 shows the values of those parameters that will also be used later in the evaluation. These values were determined with the intent to establish communities of a useful size (a community which includes all users of the system wouldn't be of much use, as well as communities which are too small) where the members have a varying degree of relationship with each other. Since there is no absolute way to fulfill both of these aspects, the values are based on observations and estimations with the help of experimental analysis.

Algorithm (parameter)	Option 1	Option 2	Option 3	Option 4
Used Authors (reuse instances)	1	10	20	50
Similar Resources (similarity)	5	20	40	-
Weak Author Communities (connectivity)	10	20	50	-
Strong Author Communities (connectivity)	15	20	-	-

Table 3.2: The parameter values shown cover a useful range regarding community size and the total number of communities in the user network. This means that values lower than the ones used as Option 1 lead to communities which probably have too many members to provide enough coherence in the members’ interests. Values higher than the ones used as Option 4, 3 or 2, respectively, would result in too few communities in the user network to be practical. These communities would have few members and a high coherence regarding the members’ interests though.

The discussion of the distance parameter in subsection 3.2.2 showed that a distance of 1 is in most cases probably suited best to derive meaningful communities. It will still be interesting to see whether higher degrees do in certain instances yield better results, so the degrees 2 and 3 will also be used later in the evaluation.

3.5 Communities for New Users

All of the alternative ways of assigning or building a community around a user presented in this chapter depend on the metadata which gives insight into their preferences regarding resources and authors.

Therefore, a community can only be established for users who are course instructors and have already used resources of other users. To enable the community-aware search for users who haven’t used any resources of other authors yet, there has to be another way to choose an appropriate community.

There are different ways to approach this. One possibility would be to use subject and keyword profiles as discussed in [HKKvP08].

3.5.1 Profiles

These profiles (which have already been mentioned in subsection 3.2.3) could be computed for every resource, author and community. Each user could then be asked to manually provide information regarding his field of interest. This information could then be used to map the user on a community that is closest to his interests.

For this approach to work, it would probably be necessary to create a taxonomy of subjects and keywords, so that a user can define his area of interest in a meaningful and standardized way without too much effort. Since people tend to use their own terms and expressions to describe the same thing (or use the same term to describe different things), the amount of different terms and similar names used in the metadata of resources might otherwise be hard to map on subjects and keywords which a new user might select for his interests. An additional challenge is the sparsity of subjects and keywords in the metadata for many resources, since many users don't bother entering the necessary information.

Some users are not clear on what subjects and keywords stand for and thus use both types interchangeably, resulting in keywords which represent a subject or category and subjects which are detailed concept descriptions which would better be represented as keywords.

A taxonomy might be established similar to the one indicated in figure 3.7 with categories which correspond to the LON-CAPA subjects and subcategories which might correspond to LON-CAPA keywords. Creating this taxonomy itself would pose a challenge, because it should be acceptable to all users of the system and the existing metadata would somehow have to conform to the new taxonomy. An approach to achieve this conformance could be to first compute the subject and keyword profiles for all communities as described in [HKKvP08], which reveals how many occurrences each term has, and then revising the keywords and subjects. Subject terms which are used often might be candidates for categories or subcategories. Subjects and keywords which are rarely used possibly can be connected to other existing ones which are more popular, or several similar terms could be subsumed under another integrative term.

Although tools like WordNet⁶ could be used to remove synonymous duplicates, a lot of this work still needs to be done manually, making this a rather time-consuming endeavor.

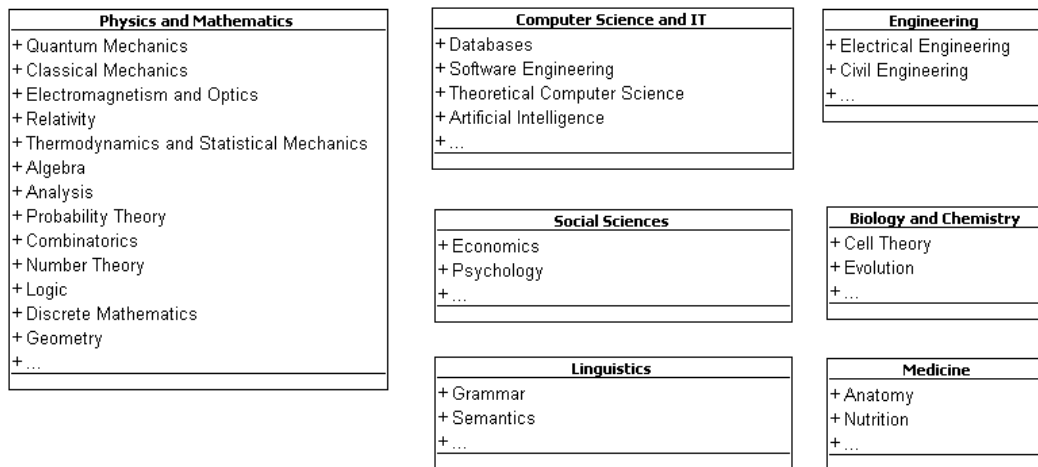


Figure 3.7: An (incomplete) draft for an example taxonomy to categorize and describe learning objects with main categories and subcategories

The process of assigning communities would then be similar to the one illustrated in figure 3.8.

The closest communities can be determined by computing a distance function between the chosen subjects (which probably should have more influence) and keywords of a user and their number of occurrences in the respective community.

3.5.2 Start Users

Another possibility is to explicitly name a start user who represents a user working in a related area. The community used would then be then one built around or assigned to this start user.

New users might not know anyone who works in their area, though. There are several possibilities how this could still work:

- A user could perform a standard keyword search and identify a useful resource. Now either the author of this resource or one of the users who

⁶<http://wordnet.princeton.edu/>

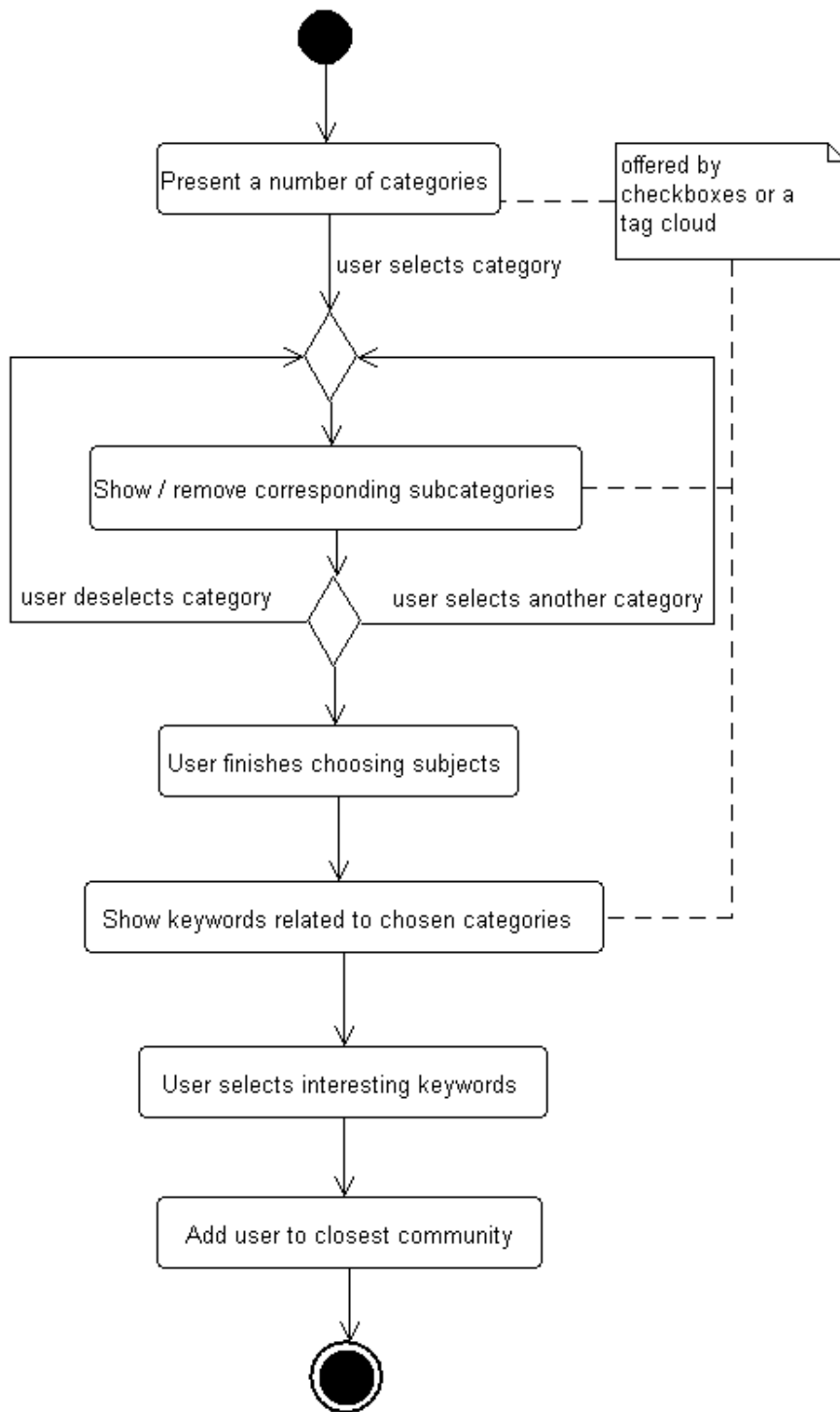


Figure 3.8: Assigning users to communities

has used this resource in his courses could be used as a start author. This way to explicitly choose a community would also be useful in general, not only for new users.

- Instead of using profiles to assign a user to a community, it would also be possible to directly compare the profiles of different users and then suggesting users (who are already related to a community) with similar profiles as start user.
- In cases, where courses are taught by different course instructors in different semesters, a new user who takes over a course can use the course instructors of the previous semesters as start users.

4 Development of a Prototype

4.1 Requirements

4.1.1 Description and Target Group

The current search approach in LON-CAPA is keyword-based. Because of the distributed architecture of LON-CAPA, the resources are distributed over a network, i.e. no server has information about all resources available. When a search term is entered and executed by the search module, it sends this query to every library server in the network. The servers then each perform a local search and report the results back to the server that issued the search.

The alternative would be to have all information available on each server in the system which would be quite resource intensive and require propagating every change on one server to all other servers within the system and thus wouldn't be very efficient.

The distributed nature of LON-CAPA also is a challenge for introducing the concept of dynamically computed communities, since they directly depend on the dynamic metadata collected for each resource. This is an issue too complex for a prototype, so the implementation will not take this into consideration. Some considerations for dealing with the distribution of the community data will be discussed in section 4.3.4.

Interviews with LON-CAPA users and a review of their feedback given through means like the bug-tracking system have revealed that the current approach of searching for resources in LON-CAPA isn't really satisfactory. Some users reported that they were unable to locate resources which existence they knew of, even though the keywords entered into the search should match the resource. The reason for this often is the sparse metadata, which cannot sufficiently be supplied by using automated text analysis of the resources. LON-CAPA also

has a browse-functionality, which allows to browse the system hierarchically by institutions, their users and courses for resources in a tree-like structure, but users were often overwhelmed by the amount of choices. On the top level, there are already more than 120 institutions to choose from ([KKBB08]). If a user doesn't already know where to look, it is usually hard to find interesting resources by browsing.

In summary, the current ways to locate resources leave room for improvement. A problem of keyword-based search is that it often retrieves too many results, so that finding the most relevant result(s) still needs a lot of manual work (reviewing every resource found), or that it retrieves too few results. By introducing social aspects to the search, it should be possible to alleviate both of these problems:

- limiting the amount of search results by filtering them according to communities and by related authors may reduce the amount of results with low or no relevance to the user and only leave the most relevant resources
- extending the amount of search results by showing resources that are related to the ones that are already found may allow to find additional interesting resources which might otherwise be hard to find

The community-aware search should be a superset of the standard search, that is, it should allow for the same functionality and have additional functionality to harness community information to achieve (possibly) better results.

The necessary social information has to be available to the search before a user issues a query to allow for an efficient search, that is, the search algorithm shouldn't have to derive the information during the search as this would be too resource intensive and take too much time to be user friendly.

This means that the community data should be retrieved from the dynamic metadata and be available at search time. This could be done at intervals on every server, although these processes should be synchronized with each other, so that the community data is the same on every server in the network. For a prototype, this aspect will be disregarded and only discussed in section 4.3.4. The relevant metadata for the prototype is available in a static file that has been created by retrieving all metadata at a given moment and then assembling this as a simple text file.

The software to be built for evaluating the community-aware search in this thesis therefore can be split into two parts with the following tasks:

- parse and analyze the static file containing the dynamic metadata of all LON-CAPA resources, then store the resource and community data into a database for fast access
- use the community data available in the database to improve the approach to locate resources in LON-CAPA by enhancing the keyword-based search with the possibility to filter and extend search results based on the available community data and the relationships between resources and users

4.1.2 Functional Requirements

- F10: The data file containing the dynamic metadata has to be parsed and interpreted line by line to extract the necessary metadata for each resource. The relevant metadata for each resource has to be stored in a database to use it when computing the communities.
- F20: There has to be a way to derive the communities described in chapter 3 with the varying options and store those communities in a database.
- F30: The search functionality should allow searching and refining search results as illustrated in figure 4.1. This includes a simple keyword-search which is already available in the LON-CAPA system.
- F40: After a keyword-search, a user should be able to extend the result set by resources which are related to the ones already found with respect to the following criteria:
 - resources which were used directly before or after (in a sequence of resources) the originally found resources
 - resources which were used in the same course as the originally found ones and were authored by a member of the user's currently assigned community

- F50: After a keyword-search, a user should be able to filter the result set so that only resources are shown which have a higher relevance according to the user's currently assigned community.

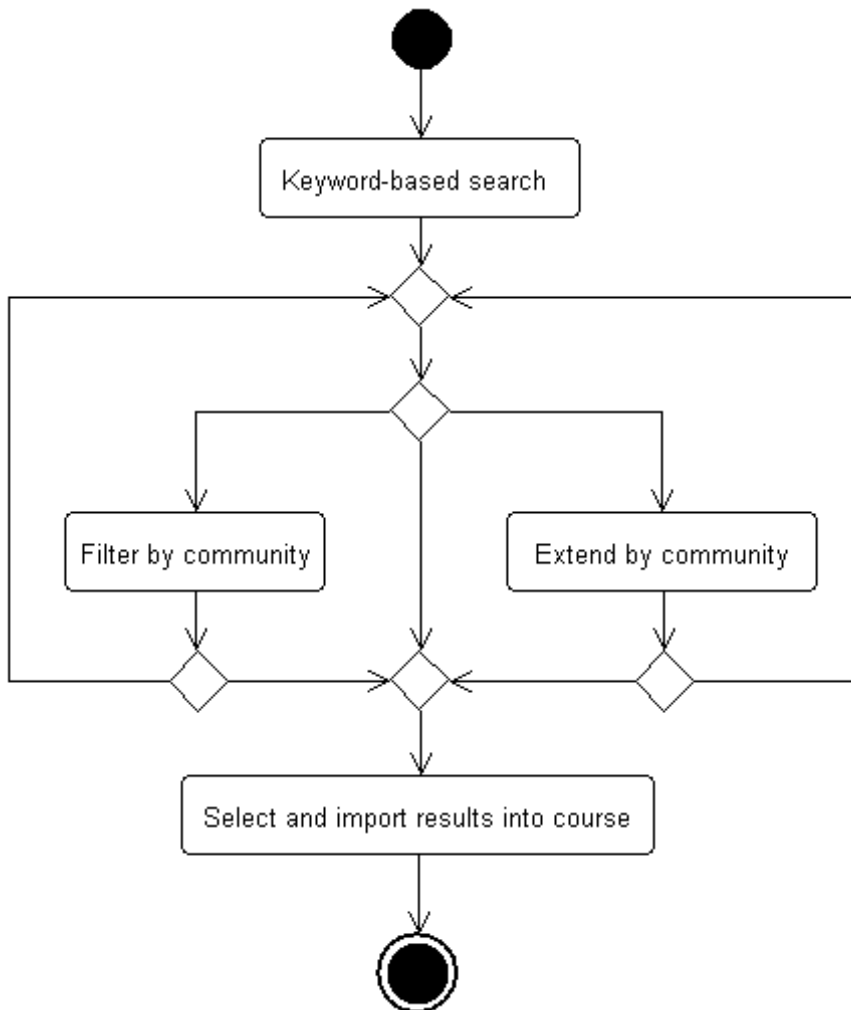


Figure 4.1: Searching and refining the search results

4.1.3 Data Requirements

The following data should be stored in a database:

- D10: The resources and their metadata.
- D20: Data about the courses and their instructors.

- D30: Data about users, especially which resources they have used and how often.
- D40: Data about communities.

4.1.4 Technical Requirements

There are also a few non-functional requirements to be met:

- T10: The front-end for the search has to be implemented using Perl 5 since this is the language LON-CAPA is based on.
- T20: The community-data should be persisted using the database used on LON-CAPA servers (MySQL).
- T30: The search should be fast enough to allow a realistic evaluation by the users.
- T40: The user-interface should be user-friendly and easy to use, since users want to spend as little time and energy as possible to locate resources.

4.2 Design and Implementation

4.2.1 Approach

This chapter describes the design and implementation of a prototype that aims to meet the elicited requirements.

The requirements are split up and dealt with by two different parts of software which limits the complexity of the tasks:

- The metadata for resources in LON-CAPA are spread across the distributed system. Building and maintaining the communities relies on access to these metadata for all resources. There are different ways to approach this challenge, however, since the focus of this thesis is on using communities to improve search, the data collection and distribution aspect will be ignored for the prototype. The metadata for all resources was compiled once into one file, so the first part of the software analyzes

this data file, extracts the necessary information and supports establishing the various types of communities. The community data is then stored to a database so it can be retrieved by the search functionality.

- The second part is the implementation and integration of the community-aware search capabilities into the existing LON-CAPA system. It enables filtering and extending the standard keyword search result list by the communities established beforehand.

The tool which creates the community-data will be implemented using Java and takes care of the functional requirements F10 and F20.

The search extension in LON-CAPA will be implemented using Perl 5 and takes care of the remaining functional requirements F30, F40 and F50. The implementation deviates a little from the requirements in that the filter functionality is accessible directly from the front search page instead of from the search result page. Since the prototype features many alternative choices on the used algorithm and parameters, this would otherwise take up a lot of space on the result page which might be too confusing to use. This deviation from the requirements doesn't affect the possibility to evaluate the usefulness of the filter functionality though.

The data and technical requirements are respected by both implementations. The functional description focuses on the alternative options of the search extension. Since extracting the data is a trivial task and building the communities has already been described in chapter 3 so that it can directly be implemented, this part of the implementation is not elaborated on in this chapter.

The aspect discussed in section 3.4 concerning users who haven't used other users' resources yet is factored in by enabling the user to choose a start user whose community is then used. Since this prototype is built to facilitate an evaluation, it is likely that the users willing to provide evaluation feedback already have an idea about which other users are relevant to them. Choosing the start author could later be supported by providing a selection of automatically detected interesting users instead of requiring the user to type the start user in.

Profiles to assign communities aren't implemented in the prototype since this is not technically necessary for a first evaluation of the efficiency of a community-aware search and would require efforts which exceed the limits of this thesis. It would however be desirable to additionally use profiles in a final version of a community-aware search.

4.2.2 Structural Description

The tool to create the community-data consists of several Java classes. One of those classes parses the data file containing the metadata for resources as well as the data file containing the courses and their course instructors and stores the necessary information into the database (see section 4.2.4). Two other classes establish the Strong Author Communities and Weak Author Communities and assign users to communities according to chapter 3 using the stored data. Another class computes additional information necessary for the algorithms which use the "Used Authors"-relationship and the "Similar Resources"-relationship to establish communities. This data and the Author Communities are also stored in the database.

The search extension itself consists of two Perl modules, one handling the interface and exchange of parameters between browser and server, the other one performing the actual search tasks according to the description in subsection 4.2.3.

4.2.3 Functional Description

Filter Algorithms

Algorithm 1: Filtering by a Community Based on the "Used Authors"-Relationship

This algorithm relies on the "Used Authors"-relationship described in section 3.3.

The parameters which influence this algorithm are:

- The number of reuse instances, i.e. how often one user has to have used resources provided by another user so that they are considered to be related; the prototype allows to use four predefined values which early experiments indicated to cover the practically useful range: 1, 10, 20 or 50 reuse instances.
- The degree up to which distance (in the user network graph) another user is considered to be related; since the LON-CAPA user network is tightly connected, a degree higher than 3 doesn't enlarge the communities any more, so the prototype features degrees between 1 and 3.
- The start user as optional parameter around whom the community is built (the default start user is the user currently logged in); this parameter allows users who could otherwise not use the community-aware search to link into existing communities and allows more experienced users to easily change their currently used community temporarily. The start user has to be typed in manually in the prototype, although this would be different in a final version; instead the start user could be selected from a number of recommendations based on the ideas in subsection 3.5.2 and subsection 4.3.3.

The search results found by a simple keyword search are then filtered so that only those resources which have been used by one of the community members are kept in the result set.

Figure 4.2 illustrates a community based on the "used author"-relationship. All users who can be reached with this relation are represented as a node where the edge connecting it to the rest of the graph is labeled with the number of times the user's resources were used by the user represented by the node he is connected to. Depending on the chosen connectivity, i.e. the minimum number of reuse instances, the size of the community for user 1 varies (irrespective of the degree):

- minimum reuse instances = 5: all users remain in the community
- minimum reuse instances = 10: the users 5, 2, 9, 12 and 7 are no longer in the community for user 1; user 2 is no longer in the community because he is cut off since user 5 isn't included any more

- minimum reuse instances = 50: only the two users 8 and 11 remain; all other users are no longer part of the community for user 1

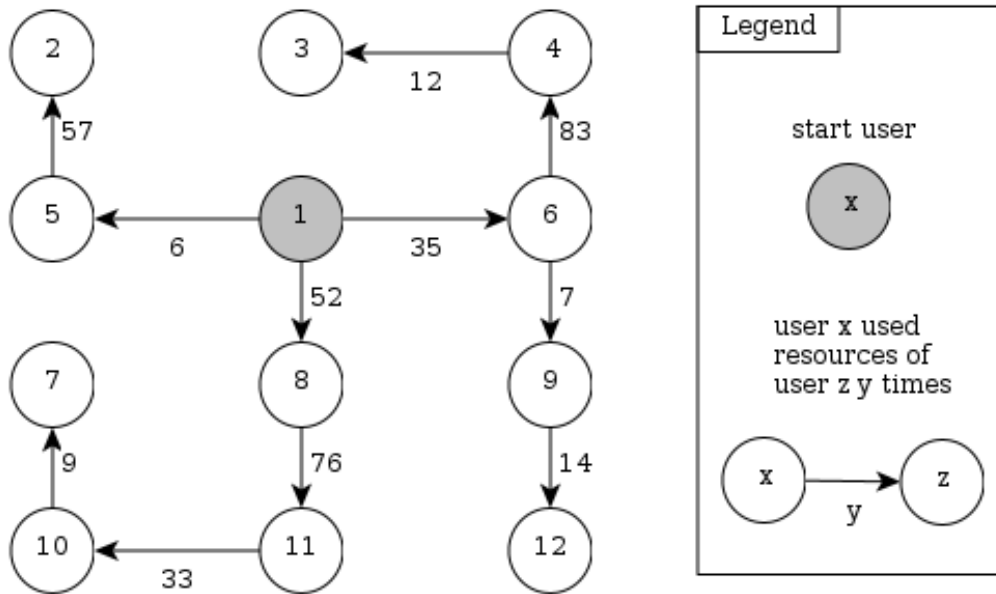


Figure 4.2: Community built by the "used author"-relation

Algorithm 2: Filtering by a Community Based on the "Similar Resources"-Relationship

This algorithm relies on the "Similar Resources"-relation described in section 3.3.

The parameters which influence this algorithm are:

- The minimum similarity (as defined in subsection 3.3.2) between users; the prototype allows to use three predefined values which early experiments indicated to cover the practically useful range: 5, 20 or 40 percent.
- The degree as described in the previous paragraph for algorithm 1.
- The start user described in the previous paragraph for algorithm 1.

The search results found by a simple keyword search are then filtered so that only those resources which have been used by one of the community members are kept in the result set.

Algorithm 3: Filtering by a Strong Author Community

This algorithm relies on the Strong Author Communities as described in section 3.2. It takes the start user and the connectivity (i.e. the number of common courses two authors have provided resources for) as parameters. Since the computation of Strong Author Communities is resource-intensive (the underlying clique-problem is NP-hard), the lowest connectivity chosen for the evaluation is 15; the other option uses a connectivity of 20. Experiments conducted for this thesis indicated that for most authors a connectivity higher than 20 doesn't lead to a useful community, which suggested the upper limit of 20 for the connectivity value. The search results found by a simple keyword search are then filtered so that only those resources which have been provided by one of the community members are kept in the result set.

Algorithm 4: Filtering by a Weak Author Community

This algorithm works in the same way as algorithm 3 while using the Weak Author Communities which were modified with the degree parameter as described in section 3.2.

Extension Algorithms

Algorithm 5: Extending by Directly Related Resources

This algorithm takes one resource as parameter and extends the result list by resources which were used directly before or after (e.g. in a navigable sequence of resources) the chosen resource. This information is available in the metadata and is updated continuously in LON-CAPA.

The resources which are found this way are directly related to the original one. It is possible to selectively follow a "path" of resources which are closely related in this way by using the direct extension on one of the new resources.

Algorithm 6a: Extending by Resources in the Same Courses

This algorithm takes one resource as a parameter and extends the result list by resources which satisfy the following conditions:

- they are used in the same course the original resource is used in at least once

- they have been authored by one of the currently assigned Strong Author Community members
- they share at least one keyword term of the initial search query with the original resource (this requirement was established because early experiments revealed that the number of additional resources would otherwise be too high and the relevance of additional resources would often not be high enough)

Algorithm 6b: Extending by Resources in the Same Courses

This algorithm works the same way as algorithm 6a, but uses the assigned Weak Author Community instead of the Strong Author Community.

Ranking and Sorting

Some of the communities allow to rank the search results by their popularity:

- When using Author Communities: The resources which were authored by a member of a Strong Author Community or a Weak Author Community could be ranked by how interested the searching user is in each individual author in the community. This could be defined by how often the searching user has used resources of each author, so that the resources provided by the author whose resources the searching user used the most (i.e. when assembling courses or compound learning objects) would be ranked higher than those provided by an author whose resources the searching user used less. Since the Weak Author Communities used in this thesis are centered around one (or a few alternative) user(s), these authors' resources could also be ranked higher.
- When using communities established by similarity in the reuse-profile as described in section 3.3: These communities are centered around users by establishing relations to users with a similar reuse-behavior. Resources can be ranked based on:
 - how big the distance of another user is to the searching user in the respective community graph
 - how high the similarity in the reuse-behaviour computed according to the formula in section 3.3 is

- When using communities established by the "used author"-relationship as described in section 3.3: Resources could be ranked in the same way as when using the reuse-similarity based communities; instead of using the similarity, the second ranking criterion would be the number of reuse instances, i.e. how often an author's resources have been used by the searching user

Other factors for ranking the results like the access count or explicitly named friends or relationships could also be taken into account in combination with the methods mentioned above.

4.2.4 Data Model

The reuse-data and the derived community-data will be stored using the object relational mapping shown in figure 4.3.

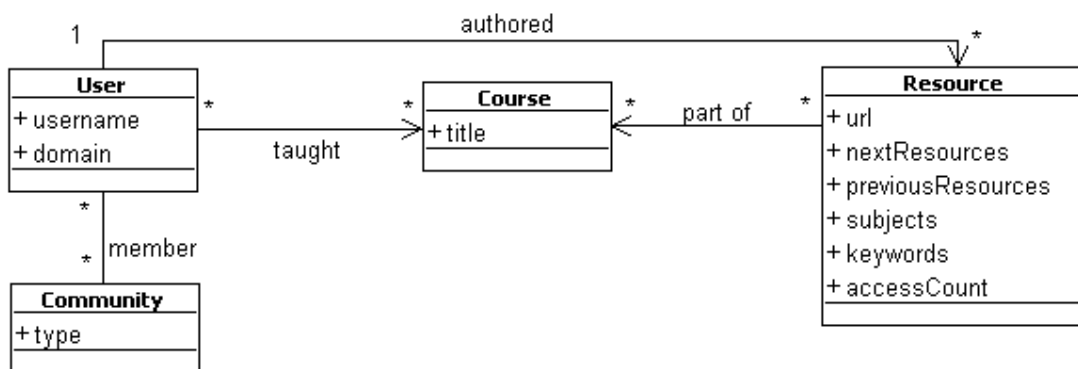


Figure 4.3: Data model

The main entity is the user which represents a learning object repository user, in this case a LON-CAPA user with a unique identifier. In LON-CAPA, this identifier consists of a user name concatenated with a colon and the users domain (e.g. "newton:tcc", where tcc could stand for the Trinity College in the University of Cambridge). A user can have a number of authored resources, where a resource represents the atomic element of educational content (learning object) in the repository. A user can also teach a number of courses which

in turn consist of a number of resources. Courses can be structured using resources which have been assembled into sequences and chapters (or folders).¹ The relationship between a course and the resources which are part of it is used to determine which resources a user has used when establishing the communities. The resource entity has some basic metadata as properties. The property named "nextResources" is a list of resources which were used directly after this resource (e.g. in a sequence of resources which forms a lesson for a certain topic). The property named "previousResources" is a list of resources which were used directly before this resource, accordingly. Representing this in the data model is useful, because the extend functionality can then be performed at different levels: depending on the needs of a user, potentially relevant resources might be suggested from a whole course containing a resource which has previously been identified as interesting, or only from those used directly before or after the interesting resource in a sequence.

4.2.5 Technologies Used

As already mentioned, the implementation used both Java and Perl as programming languages.

The Strong Author Communities described in section 3.2 are represented in the user network graph as cliques. The library JGraphT ([JGr]), which implements a version of the Bron-Kerbosch algorithm ([BK73]), was used to compute the Strong Author Communities in the user network.

As already stated in section 4.2.1, the tool to create the community-data was implemented using Java, while the search itself used Perl according to the technical requirement T10. The database used to store the community-data is MySQL, since this database is already present on and used by LON-CAPA servers.

4.2.6 Integration and User Interface

The standard LON-CAPA search is realized as a Perl module which handles both the interface and the search itself. This module is replaced by an adapted

¹see section 2.1.3 for a more detailed description of learning objects and their use in LON-CAPA

version which mainly takes care of the interface, supported by another Perl module which now handles the standard and community-aware search functionality.

Since this evaluation prototype features several different filter algorithms, each with its own parameters, this functionality is not integrated into the search results page. Instead, filtered searches can be issued directly.

A final version of this community-aware search would probably only feature the most useful of those algorithms (and use practical default parameters), so that it could easily fit on the search page.

The extension algorithms are included for each search result. They are based on the same community that was chosen before for the filter algorithm.

While this interface is not optimal, it allows to compare the usefulness of the different algorithms.

4.2.7 Result

The resulting user interface for the prototype is shown in figure 4.4.

The screenshot shows a search interface with the following elements:

- Search Input:** A text box containing "energy capacitor heat" and "kortemey.msu". Below it, instructions read "enter keywords (no boolean expressions)" and "start user (optional), format **user:domain**".
- Algorithm Selection:** A section titled "Algorithm: (choose one of the five)" with five radio button options:
 - (1) Used Authors
 - (2) Related Coordinators
 - (3) Weak Author Communities
 - (4) Strong Author Communities
 - (5) Only Keywords
- Parameters:** A section titled "Parameters (more results <----> less results): (colors indicate which parameter is effective for which algorithm)".
 - reuse instances:** Radio buttons for 1, 10 (checked), 20, 50.
 - similarity in %:** Radio buttons for 5, 20 (checked), 40.
 - connectivity:** Radio buttons for 10, 20 (checked), 50.
 - connectivity:** Radio buttons for 15, 20.
 - degree:** Radio buttons for 3, 2, 1 (checked).
- Buttons and Controls:** "Search", "CLOSE", "Summary View" (dropdown), and "20" (dropdown) "Records per Page".

Figure 4.4: User interface for the search prototype

The test user can enter the keywords he is interested in, possibly a test user (see section 3.5) and choose between the different filter algorithms and a standard keyword search, so that the effect of a filter can be compared with the result set of keyword search. To influence which community is used, the test user can also choose between several parameter options, which change the selectivity of the filter algorithms and have been described in subsection 4.2.3.

Once a search has been issued and computed, the result page (see figure 4.5) will be shown. The section directly above the search results is only used for the evaluation. The test users can rate the quality of the currently shown search results on a scale from 1 to 5. Additionally, they can enter a comment.

There are several possibilities to sort the resources in the result set (see subsection 4.2.3):

- by the popularity of the resources within the currently used community and the interest of the searching user in the individual community members (when the currently used community is an Author Community)
- by the relevance of the resources to the searching user (when the currently used community is one of the other community types)
- by the access count of the resources
- by the title of the resources
- by the author names of the resources

The test user then can further modify the result list by extending as described in subsection 4.2.3 (Extension Algorithms), which will show additional results as seen in figure 4.6 (the additional resources are labeled as "extension" in red font).

This implementation allows a first evaluation of the different algorithms. However, to fully meet the requirements in section 4.1 it will be necessary to integrate the filtering option(s) into the result view. This will bring its own challenges since too many options might confuse the average user unnecessarily. Once an appropriate filter algorithm has been identified as useful with certain parameters, it should be possible to add the filter functionality directly in the result view without adding too much complexity to the interface.

For the evaluation, the search prototype is installed on a test server which is connected to the LON-CAPA network, so that users can also access and evaluate the resources they find with the search.

Course, Portfolio and Catalog Search

There are 7 matches to your query. [Revise search](#)

Search: energy capacitor heat

How well do the results meet your expectations?

(bad) 1 2 3 4 5 (good) ; your comment: (max. 20

[IMPORT](#)

Sort by [relevance \(only for algorithms 1 and 2\)](#) [Descending](#)

[Prev](#) [Reload](#) [Next](#)

Results 1 to 7 out of 7

[?](#) 6 [Switching on a Circuit](#)

[/res/msu/physicslib/msuphysicslib/60_CircuitsTransient_RC/msuprob01.problem](#)

[... Extend by direct relation](#)

[... Extend by course and SAC](#)

[... Extend by course and WAC](#)

> physicslib.msu

[<](#) 5 [Theoretical Background for Energy Stored in a Capacitor Activity](#)

[/res/csm/csmphysicslib/P200_Materials/StudioActivities/Block2-Circuits/EnergyStoredInCapacitor/background.html](#)

[... Extend by direct relation](#)

[... Extend by course and SAC](#)

[... Extend by course and WAC](#)

Figure 4.5: Display of the search results

[IMPORT](#)

Sort by [Network-wide number of accesses \(hits\)](#) [Descending](#)

[Prev](#) [Reload](#) [Next](#)

Results 1 to 20 out of 59

[?](#) 129784 [Loop of Wire in a Field](#)

[/res/msu/physicslib/msuphysicslib/64_EMInduction1/msuprob24.problem](#)

[... Extend by direct relation](#)

[... Extend by course and SAC](#)

[... Extend by course and WAC](#)

> physicslib.msu

[extension](#)

[?](#) 101730 [Charge and Voltage](#)

[/res/msu/physicslib/msuphysicslib/56_Capacitance/msuprob04.problem](#)

[... Extend by direct relation](#)

[... Extend by course and SAC](#)

Figure 4.6: Additional result after use of an extend algorithm

4.3 Additional Aspects

4.3.1 Access Rights and Security

Users in a learning object repository usually want to be able to influence how the resources they provide can be accessed and used. They want to control who can locate their resources, who can retrieve them, who can use them and who can edit them. Lecturers especially don't want their students to access exam problems before an exam or before they are supposed to be accessed. In LON-CAPA, they also don't want their students to be able to gain access to the underlying mechanisms (i.e. the source code) which randomize problems, since this might enable them to solve the problems without doing the actual work. This aspect has been ignored in the prototype implementation since it doesn't directly affect the question whether resources are relevant in a search situation or not. The question of data security is important for the implementation of a community-aware search which is to be used system-wide on servers, though.

4.3.2 Integrating Community Elicitation with Search

The implementation of a prototype was split up in two separate parts due to performance and complexity issues. It might appear appealing to integrate both tasks, so that the communities are built whenever a search is issued, so that the most recent changes in the communities are immediately considered. However, computing the communities is in most cases a rather complex task which might add too much delay to search queries. Since a fast response is important to most users, this doesn't seem practical. Searches are usually issued at a higher frequency than changes occur in the reuse-behavior of users, which is another factor that discourages handling both tasks at the same time. Since LON-CAPA is a distributed system, computing the communities so frequently would also lead to a lot of network traffic (see subsection 4.3.4). A better approach would probably be to compute the communities at regular intervals which balances out the cost in computation time with the benefit of taking recent changes in communities into account.

4.3.3 A User-Friendly Interface

Since several ways to create communities were proposed in this thesis, the user interface of the prototype developed to assess and evaluate the ideas of this thesis leaves room for improvement. It has too many options, which tend to confuse the average user. The main goal for the evaluation is to learn more about how useful and effective the different approaches are.

However, for an implementation of these search-extensions to be accepted and of value in everyday use, the interface should be more intuitive and easy to use.

This especially pertains to the following issues:

- In the prototype, filtered searches can only be issued from the start page, due to the amount of options available. Both the filtering and extension options should ideally be accessible from the search result page.
- The currently used community can also be used to provide additional filter options: e.g. one option would be to show a tag cloud of the subjects and keywords used by a community, where the size (i.e. the importance) of each term would be determined by the number of occurrences of the term in the community's used and/or authored resources.
- Only the most useful methods to filter and extend should be accessible by default, so that the result page doesn't get cluttered and confusing.
- The start user can only be typed in manually by users in the prototype. An improvement for this would be to provide the user with a list of candidates who might be interesting to him:
 - based on resources which are already used in the course context he is currently working on
 - for every resource found with a query, each result could feature the users who used the respective resource in a course (e.g. ordered by the number of reuse instances), so that a user could click on one of those users and either receive more results, or limit the results to those which are related to this user (e.g. regarding which other resources were used by this user)

4.3.4 Managing the Community-Data in a Distributed System

In the previous chapters, the community-aware search was designed for a monolithic system where only one server exists that harbours all resources and users. This approach is sufficient as a proof of concept and allows to evaluate the community-features. To provide them across a distributed system like LONCAPA, it is necessary to have the community-data available on every server in the network that should be able to use the search. Since this data has to be recomputed periodically based on the dynamic reuse data of the resources, the question arises what would be a realistic and efficient way of doing so with regards to the following issues:

- the computation of the communities is resource-intensive
- the metadata necessary to compute the communities is distributed across the network
- the community-data should be consistent across the network, i.e. it shouldn't matter which server a user logs into - he should have the same communities on every server
- regularly exchanging data about communities might be too communication-intensive

There are basically two options:

- collecting the necessary metadata from all servers and computing the communities on one or a few central servers and then distributing the data across the network
- distributing the necessary metadata to each server in the network and computing the communities on each server separately

Both face the problem that they do in a way break the idea of a distributed system, the first option because it reinstates a centralized structure where the network depends on a few certain servers. The second option requires to have all data to be available on each server in the network.

This is a challenge which probably needs a closer analysis regarding how much of an impact those drawbacks actually present.

4.3.5 Other Possible Improvements

Other improvements that could be of use and implemented for a community-aware search at a later stage might include:

- excluding resources from the result list which were authored by a certain author or members of a certain community
- explicitly allowing to select an interest profile using some sort of (maybe partially) pre-defined taxonomy and then matching this interest profile with communities
- the possibility for a user to manipulate his community, e.g. by removing certain other users or adding new ones
- the possibility to maintain several communities which relate to different areas of interest per user
- ranking the results by the reuse-behaviour in previous searches by other users as described e.g. in [FS04]
- using tag clouds (as mentioned in subsection 4.3.3) to allow filtering and extending based on keywords that users in a community used; the tags could be sized according to their importance (i.e. the number of occurrences in the metadata of resources authored by all members) in the community
- showing the users who used a resource when displaying the results (as mentioned in subsection 4.3.3) ; this might help new users to find interesting communities once they have found one interesting resource, thus enabling them to use the filter algorithms without having to explicitly provide an interest profile
- an explicit tagging system, which allows users to tag resources they like
- an explicit friends-concept which would include designating other users as friends; this could then also be used to influence the search
- explicitly announcing new resources that were created within a user's community (or close to the community)

- extending search patterns based on the history of searches (i.e. suggestions to make it easier for the user to form appropriate search queries), e.g. similar to the functionality offered by Google Suggest²
- providing recommendations of resources based on which resources were clicked on in similar search scenarios by other users
- showing the members of the community currently used, allowing to change to a different community one of the members belongs to
- filtering and extending the search results according to the context the search was issued in, i.e. presenting resources that have a similarity to the resources currently present in a course when searching for resources to complete this course
- popularizing new and underused resources; the methods discussed in this thesis mainly rely on the fact that resources are already used in course; new resources could be "advertised" additionally
- integrating search and browse functionalities in one interface

As indicated by this rather long list, the amount of possibilities is large and although many of those possibilities have the potential to be useful, only a selection of them can usually be implemented and evaluated regarding their user acceptance and usefulness at a time.

²<http://labs.google.com/>

5 Evaluation

5.1 Objective

The evaluation should determine if and which of the different filter- and extend-algorithms are perceived to be useful by the users. It should also identify the best parameters to use for the different alternatives.

Since the use of both ways of influencing the search results will probably be different in different scenarios, evaluation participants were asked to simulate assembling a course within their field of interest using the prototype search.

5.2 Setup and Execution

Every test user had access to a search prototype (see subsection 4.2.7) that featured both the standard keyword search as well as the new community-aware search functionality, i.e. the different filter- and extend-mechanisms presented in subsection 4.2.3.

The test users were encouraged to compare the different algorithms and parameter options using their own search queries. In the result view the user could then rate the results according to his or her expectations on a range from 1 (bad quality of results) to 5 (very good quality of results). Additionally, a free-text comment field could be used to give individual feedback.

The search queries and the corresponding evaluations were recorded in a database.

5.3 Results

A total number of 13 users experimented with the community-aware search of which 8 users participated in the evaluation by rating and commenting on the results of their search queries. The other users showed interest by issuing searches, but didn't provide any feedback on their results. A total of 109 ratings and 34 comments were entered by the participants on the different search result sets.

According to their comments, the users' satisfaction with the community-aware search prototype was mixed. A little over one third of the comments were positive (i.e. they expressed satisfaction with the search results and the possibility of filtering or extending results in the offered manner), a little less than one third of the comments expressed dissatisfaction with the search results (although in some cases the criticism was not specific to the community-aware functionality). The remaining comments were either neutral regarding the quality of the search results, or they were questions on how the search works regarding certain aspects or they referred to the user interface.

The ratings offer a similar picture: the average rating for the results obtained with the different filter algorithms was 3.11 on the scale from 1 to 5 while the average rating for the simple keyword search was 2.1 on the same scale. This indicates that the community-aware search may be superior to a simple keyword search. These ratings were averaged over a rather small data set though (79 evaluations for the different filter algorithms and 10 for the keyword search), so these numbers have to be interpreted with caution.

The evaluation of the extension-mechanism also yielded mixed results. Users sometimes felt that the additional resources were too far away from the area they were actually interested in. At the same time, some test users remarked that they found resources they would otherwise not have found. Since there is a trade-off between increasing the number of results and keeping the additional results relevant, this evaluation result was probably to be expected. The community-aware features are meant to be used additionally to the conventional search method, so finding results which could not or hardly be discovered using those conventional methods should be considered as an indicator that this functionality is beneficial to the search process. The average rating for

Algorithm	Filter	Extend
Keyword Search	2.1 (based on 10 evaluations)	-
Used Authors	3.47 (based on 30 evaluations)	-
Similar Resources	2.48 (based on 21 evaluations)	-
Weak Author Communities	3.03 (based on 12 evaluations)	3.02 (based on 5 evaluations)
Strong Author Communities	2.89 (based on 16 evaluations)	3.03 (based on 12 evaluations)
Extend by direct relation	-	3.67 (based on 3 evaluations)

Table 5.1: Average ratings (rounded on hundredths) on a scale from 1 to 5 for the different algorithms (regardless of the used options)

the result set obtained by using the extension algorithms was 3.15 on the scale from 1 to 5.

Table 5.1 shows the average ratings for the different algorithms.

For the filter-functionality the "Used Authors"-algorithm achieved the highest average rating, which is significantly higher than that for the "Similar Resources"-algorithm. The algorithms based on Author Communities both have similar ratings which are higher than those for the "Similar Resources"-algorithm but lower than those for the "Used Authors"-algorithm. Therefore it is not entirely clear whether one of the Author Community types is better suited to determine relevant authors. The evaluation results for using the Author Communities to extend the search result set are nearly the same. The extension based on directly related resources leads to better ratings for the search result sets, which confirms that resources which were used together in a sequence have a high similarity between each other.

The overall result for the filter- and extend-functionality appears to be positive and encouraging; however, the goal to differentiate between the different algorithms and options could only partially be achieved by this evaluation due to an insufficient amount of evaluation data. While the different algorithms might be differentiated by the results, there were not enough evaluations for the different options so that these could be compared in a meaningful manner.

The average test user only issued a few search queries and most of them simply used the default settings. This was probably caused by the evaluation design which was relatively open (the test users were asked to define their own search queries and use the prototype to compose a course) so that the results would be collected in search situations comparable to everyday use.

An alternative might have been to pre-determine search queries, possibly with even more specific search scenarios. Since different users work in different areas of interest, they may not be qualified to evaluate the search results of a query that is designed to find resources in an area which they have little or no knowledge of though.

In conclusion, the evaluation user response gave insight into how the different algorithms affect the quality of search results found but was not high enough to allow a realistic deduction concerning the different parameter options. Evaluating search results is a hard task that requires a lot of effort, because the quality of results is a rather vague metric and the needs of the searching user vary in different scenarios. The users also had to come up with their own search queries and form an idea of what kind of results they would want for that query which increased the complexity of the task.

The existing evaluation data provided by the test users who did participate in the evaluation does suggest that a community-aware search can be superior to a simple keyword search. Further investigations would be desirable to confirm the findings in this evaluation.

It might be possible to gather additional insight by focusing on very specific scenarios, this would need a close interaction with users though, since those scenarios would have to be designed for their needs.

To provide some insight into the differences between the various algorithms, the next section offers a comparison for a typical search scenario.

5.4 Comparison

5.4.1 Filter Algorithms

This section shows a comparison of an example search query (using three search terms) issued by an example user with the different filter algorithms. While this doesn't qualify as a realistic evaluation, it provides some insight in the usage behavior of the developed algorithms.

Table 5.2 shows the number of results that are found with the different algorithms and parameters. The first column denotes the algorithm used, the

Algorithm	Degree	Option 1	Option 2	Option 3	Option 4
Keyword (standard)	-	49	-	-	-
Used Authors	1	8	6	4	4
	2	13	9	7	6
	3	15	12	8	6
Similar Resources	1	11	2	0	-
	2	15	3	0	-
	3	16	7	0	-
Weak Author Communities	1	6	0	0	-
	2	6	0	0	-
	3	6	0	0	-
Strong Author Communities	-	6	6	-	-

Table 5.2: Comparison of the number of results with the different filter algorithms for an example user and query

second the degree as described in chapter 3. Options 1 to 4 refer to the parameter specific to each algorithm, that is for the "Used Authors"-algorithm the number of reuse instances, for the "Similar Resources"-algorithm the similarity between the reuse profile of the coordinators, for the "Weak Author Community"- and "Strong Author Community"-algorithms the connectivity in their respective graph (refer to chapter 3 and subsection 4.2.3 for background on and a closer description of those parameters).

Table 5.3 shows the values of those options which were used for the evaluation.¹ The example query leads to 49 results that match the chosen keywords. The query used three keywords and thus was already rather specific; as shown in [HKKvP08], most users use less terms for their queries which would lead to more results in general. The relations between the different algorithms for other search queries with less terms are in most cases similar to this example though. The number of results is decreased significantly by every filter algorithm, which indicates that many of the original search results might have been of little relevance to the searching user.

Resources that were filtered out include those with a low access count, although this is not the primary criterion. On the other hand, the most popular resources regarding the access count are kept by most algorithms as expected, since a correlation between reuse and access count is likely.

The reason that the results for the WAC algorithm don't increase in numbers

¹the keyword search works without additional options

Algorithm (parameter)	Option 1	Option 2	Option 3	Option 4
Used Authors (reuse instances)	1	10	20	50
Similar Resources (similarity)	5	20	40	-
Weak Author Communities (connectivity)	10	20	50	-
Strong Author Communities (connectivity)	15	20	-	-

Table 5.3: The parameter values for the different filter algorithms as established in section 3.4

with a higher degree is that in many cases with a degree of one already all (or at least most) related users are included in the corresponding community (this was also discussed in section 3.2). The fact that the SAC algorithm with option two yields more results than the WAC algorithm which uses the same connectivity indicates that the Strong Author Communities are not always a subset of the Weak Author Communities. Although all SAC are in general a subset of a larger WAC, the SAC assigned for a user might not be a subset of the WAC assigned for this user. This is the case when many of the interesting authors for a user are in the assigned WAC, but not in the SAC which is the subset of said WAC.

To give an idea about the quality of the results, the most accessed resources in the result sets could be compared. The resources titled "LC and Oscillator" and the untitled resource named "problem04.problem" (see figure 5.1) were kept by all of the algorithms. Since the different filter algorithms have different criteria to define the relevance of resources, this should be an indication that this resource is relevant. However, this doesn't help in differentiating the algorithms.

Table 5.4 shows how the choice of the options influences the selectivity of the different algorithms for a resource called "Chemical Principles for BCH521" (not shown in figure 5.1). All algorithms designate this resource as relevant at least once using one of the several options though.

Another resource in the list called "Essential concepts" (not shown in figure 5.1) however is only included by the "Used Authors"-algorithm as illustrated in table 5.5, albeit only with some of the parameter options.

Similar differences exist for other resources, so it is clear that the alternative communities have different effects and don't all represent the same information, although they are all based on similar data.

? 22576 [LC and Oscillator](#)

[/res/msu/physicslib/msuphysicslib/69_CircuitsAC_1_LR_LC/msuprob05.problem](#)

[... Extend by direct relation](#)

[... Extend by course and SAC](#)

[... Extend by course and WAC](#)

> physicslib:msu

? 4141 [untitled](#)

[/res/msu/westfal3/exam1/problem04.problem](#)

[... Extend by direct relation](#)

[... Extend by course and SAC](#)

[... Extend by course and WAC](#)

> westfal3:msu

? 2465 [The energy in an oscillating LC circuit containing an inductor](#)

[/res/sc/gblanpied/courses/usclib/hrw7/chapter31/hrw31p7.problem](#)

[... Extend by direct relation](#)

[... Extend by course and SAC](#)

[... Extend by course and WAC](#)

Figure 5.1: The first three results found by filtering based on the assigned Strong Author Community

Algorithm	Degree	Option 1	Option 2	Option 3	Option 4
Used Authors	1	yes	yes	yes	yes
	2	yes	yes	yes	yes
	3	yes	yes	yes	yes
Similar Resources	1	yes	no	no	-
	2	yes	no	no	-
	3	yes	yes	no	-
Weak Author Communities	1	yes	no	no	-
	2	yes	no	no	-
	3	yes	no	no	-
Strong Author Communities	-	yes	yes	-	-

Table 5.4: Filter queries which keep the resource called "Chemical Principles for BCH521" are marked "yes", others "no"

Algorithm	Degree	Option 1	Option 2	Option 3	Option 4
Used Authors	1	yes	no	no	no
	2	yes	no	no	no
	3	yes	yes	no	no
Similar Resources	1	no	no	no	-
	2	no	no	no	-
	3	no	no	no	-
Weak Author Communities	1	no	no	no	-
	2	no	no	no	-
	3	no	no	no	-
Strong Author Communities	-	no	no	-	-

Table 5.5: Filter queries which keep the resource called "Essential concepts" are marked "yes", others "no"

Algorithm	1st resource	2nd resource	3rd resource	4th resource
Algorithm 5 (direct relation)	16	10	4	4
Algorithm 6a (course and WAC)	686	284	180	140
Algorithm 6b (course and SAC)	953	494	354	315

Table 5.6: Comparison of the number of results with the different extension algorithms for an example user issued on four different resources: the number of additional results is higher for resources with higher a access count; algorithm 5 leads to rather few additional results with a high relevance as indicated in chapter 5, while algorithms 6a and 6b lead to many more results with varying relevance to the user

5.4.2 Extension Algorithms

Table 5.6 shows a comparison for the different extension algorithms. They were issued on the four results found with the "Used Authors"-algorithm with option 3 from table 5.2.

The four results differ in how often they were accessed within the system, with the first resource having the highest access count (which indicates a rather high popularity) and the fourth resource having the lowest access count. It is likely that a higher access count also means that the resource is embedded in more courses (since this will usually increase the number of people who will work with it and by that, the number of accesses). Resources with a higher access count are thus expected to yield more additional results with the extension algorithms.

The "course and WAC"- and "course and SAC"-algorithms generally lead to a rather large number of additional resources. This shows that the originating resources was used together in courses with many other resources which had at least one of the keywords in common with the originating resource. The direct relation in comparison leads to less additional results, which is also expected, since the resources which were used directly before or after a resource usually form a subset of the resources which were used in the same courses as the original resource.

As expected, the extension algorithms lead to more additional results for resources which have a high access count. The amount of additional results is probably too high to be of practical use, though. This indicates that the rela-

tionships used for extending search results are probably not specialized enough to provide only the most relevant resources.

5.5 Discussion

The results on the usefulness of a community-aware search functionality for LON-CAPA indicated that the community-features can help users to locate relevant resources. Regarding the user satisfaction, the "Used Authors"-relationship appeared to be superior to the others on which the communities can be based when filtering the search results. The direct relationship defined by the use of resources in sequences yielded the best results when extending the search results.

The evaluation results have to be interpreted with caution, since only a small number of users was willing to participate in the evaluation. The evaluation participants also rarely provided enough valuable insight on the perceived quality of results for the varying parameter options.

The limited support for the evaluation appears to conflict the data of a previous survey conducted within the context of [HKKvP08], which indicated that people would be willing to invest some time into community-features. However, the questions in this survey mainly related to providing an interest-profile and evaluating individual resources a user is familiar with. Evaluating the quality of search result sets is a more complex task, since usually at least several resources have to be examined to determine how relevant they are to the search task. Since the goal of the evaluation was to compare several different algorithms, this may have exceeded how much effort most users were willing to spend when indicating their support in the survey. A few users might already have been deterred by the amount of options which were provided by the prototype user interface.

Since the evaluation setup used for this thesis had limited success in attracting evaluation participants, it is desirable to find another way to evaluate the community-aware search, so that the evaluation results in this thesis can be confirmed.

A different approach for the evaluation might be to separate the following aspects for an evaluation:

- the efficiency of different search filter algorithms, i.e. a simple keyword search algorithm and one or more community-aware algorithms, in separating results of high relevance from those with lesser relevance concerning a certain search task
- the efficiency of different search extension algorithms which recommend additional resources to a user
- the user acceptance and satisfaction with filtering and extending search results using a certain interface

Evaluating these aspects individually would reduce the complexity of an evaluation and the efforts the test users would have to spend.

To evaluate filtering algorithms, users could be asked for individual search queries which correspond to their areas of interest. The results of different search algorithms could then be compiled for each of those queries and users, so that the search results found by several different algorithms could be compared on a single web page. This comparison, however, would still be rather complex if done properly.

To further reduce the efforts for the users, they could be asked to rank the search results of the keyword search according to their relevance. This ideal ranking could then be compared to the results found by the filter algorithms, e.g. regarding whether the most important resources are still in the list.

Evaluating extension algorithms could follow a similar approach. The users could be asked to rank a list of search results which was compiled by adding relevant resources recommended by different extension algorithms. This would allow to derive which algorithm recommended the most interesting resources.

The evaluation of a practically useful user interface would then be the last step before implementing a final version of the new search features.

6 Conclusion

6.1 Summary

After the introduction of learning objects and repositories, this thesis discussed how keyword search in learning object repositories can be improved by applying the concept of communities, using this to determine the relevance of resources and filtering or extending the search results according to this additional information. Section 2.2 gave a background on what constitutes a community and discussed several ways in which communities could be used to improve search. The chapter finished with a short presentation of related work.

Chapter 3 discussed several ways to derive communities in LON-CAPA based on dynamic usage data. These communities then provided the basis for the design of filter and extension algorithms which may help users to locate relevant resources. These algorithms were implemented in a prototype search which was documented in chapter 4.

Chapter 5 then reported on the evaluation of the community-aware search prototype and presented a comparison of the effects of the different algorithms. It discussed, why the evaluation process was only partially able to provide the information necessary to judge the effectiveness of the new functionality and proposed a different approach for the evaluation which could potentially lead to more evaluation participants by decreasing the complexity of the evaluation task and the efforts the users have to spend.

The evaluation data suggested that one of the algorithms to filter less relevant resources out of a result set (and by this, the way to establish communities) is superior to the others and leads to significantly better results than the simple keyword search. The algorithms to extend a result set were a little bit harder to differentiate concerning their ability to identify relevant resources, although the data suggests that all of them are able to identify potentially relevant

resources. The algorithm relying on direct relationships between resources which were used together in sequences appeared to provide the best additional results of the evaluated alternatives.

6.2 Outlook

In conclusion, the data gathered for this thesis supports the assumption that a community-aware search might be beneficial in the search process and help in identifying relevant resources. The use of social information for the search functionality in other learning object repositories like CampusContent is likely to be of value for its users.

One major challenge in the implementation of a community-aware search is to establish practically useful and meaningful communities without requiring a lot of effort from the users.

Since the evaluation results of the community-aware search are based on a rather small data set, it would be desirable to conduct another evaluation to confirm the results (and to derive conclusions regarding the different parameter options) with a different setup taking the conclusions from section 5.5 into account.

Section 4.3 already mentioned other areas which may need a closer examination.

Extending the community elicitation and management process on a distributed architecture will probably necessitate some experimentation on how to keep the communication costs acceptable.

Since a practical and simple user interface is also important for users and since there are many ways to design a user interface for a community-aware search, this should also be a primary concern for the further development.

Bibliography

- [AA04] R. Almeida and V. Almeida. A community-aware search engine. In *Proceedings of the 13th International Conference on World Wide Web*, pages 413–421. ACM Press, 2004.
- [BHK98] John S. Breese, David Heckerman, and Karl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52. Morgan Kaufmann, 1998.
- [BK73] Coen Bron and Joep Kerbosch. Finding All Cliques of an Undirected Graph. *Communications of the ACM*, Vol. 16 (No. 9):575–579, ACM, 1973.
- [BN06] Sascha Bobrowski and Olaf Nowaczyk. Architektur eines verteilten Lernobjektrepitoriums (German). *Forschungsberichte des Fachbereichs Elektrotechnik & Informationstechnik, FernUniversität Hagen*, Vol. 2, 2006.
- [Cam] <http://www.campuscontent.de>, last accessed 2008-07-17.
- [FFB⁺07] Jill Freyne, Rosta Farzan, Peter Brusilovsky, Barry Smyth, and Maurice Coyle. Collecting Community Wisdom: Integrating Social Search & Social Navigation. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 52 – 61. ACM, 2007.
- [FS04] Jill Freyne and Barry Smyth. An Experiment in Social Search. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 3137/2004 of *Lecture Notes in Computer Science*, pages 95–103. Springer Berlin / Heidelberg, 2004.

- [HKKvP08] Peng Han, Gerd Kortemeyer, Bernd J. Krämer, and Christine von Prümmer. Exposure and Support of Latent Social Networks Among Learning Object Repository Users. *Journal of Universal Computer Science (J.UCS)*, Vol. 14, 2008.
- [HNV01] C. Hubert, B. Newhouse, and W. Vestal. *Building and Sustaining Communities of Practice*. American Productivity & Quality Center, Houston, 2001.
- [JGr] <http://www.jgrapht.org/>, last accessed 2008-07-17.
- [KAB⁺03] G. Kortemeyer, G. Albertelli, W. Bauer, F. Berryman, J. Bowers, M. Hall, E. Kashy, D. Kashy, H. Keefe, B. Minaei-Bidgoli, W. Punch, A. Sakharuk, and C. Speier. The LearningOnline Network with Computer-Assisted Personalized Approach. *Computer Based Learning in Science Conference, Cyprus*, 2003.
- [KKBB08] Gerd Kortemeyer, Edwin Kashy, Walter Benenson, and Wolfgang Bauer. Experiences using the open-source learning content management and assessment system LON-CAPA in introductory physics courses. *American Journal of Physics*, Vol. 76 (Issue 4):438–444, American Association of Physics Teachers, 2008.
- [Kor06] Gerd Kortemeyer. Re-Usable Learning Objects in Practical Use. Technical report, Michigan State University, 2006.
- [Krä05] Bernd J. Krämer. Reusable learning objects - let's give it another trial. *Forschungsberichte des Fachbereichs Elektrotechnik & Informationstechnik, FernUniversität Hagen*, Vol. 4, 2005.
- [KZ08] Bernd Krämer and Annett Zobel. Rollout of CampusContent (retrieved from <http://elead.campussource.de/archive/4/1417/> - last accessed 2008-08-31). *e-learning and education*, FernUniversität Hagen, 2008.
- [Lea02] Learning Technology Standards Committee. Draft Standard for Learning Object Metadata. (retrieved from <http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html> - last accessed 2008-08-31), 2002.
- [Lon] <http://www.lon-capa.org/>, last accessed 2008-07-17.

- [Mer] <http://www.merlot.org>, last accessed 2008-07-17.
- [MGD06] Alan Mislove, Krishna P. Gummadi, and Peter Druschel. Exploiting social networks for internet search. In *Proceedings of the 5th Workshop on Hot Topics in Networks (HotNets'06)*, November 2006.
- [Wil02] David A. Wiley. *Connecting Learning Objects to Instructional Design Theory: A Definition, a Metaphor, and a Taxonomy*. (retrieved from <http://reusability.org/read/chapters/wiley.doc> - last accessed 2008-08-31). Agency for Instructional Technology, 2002.
- [WMS02] Etienne Wenger, Richard McDermott, and William M. Snyder. *Cultivating Communities of Practice. A Guide to Managing Knowledge*. McGraw-Hill Professional, 2002.

List of Figures

2.1	Resources (learning objects) and resource assembly in LON-CAPA	7
2.2	Network Architecture of LON-CAPA	12
3.1	Relations in LON-CAPA	21
3.2	Identifying Strong Author Communities (SAC)	22
3.3	Author Communities in the User Network Graph	23
3.4	Development of a Weak Author Community taking distances between users into account	25
3.5	Weak Author Community member count and average size de- pending on the distance (degree) and connectivity: the diagrams illustrate that for a degree of 2 or 3, the average community size is close to the possible maximum size, which indicates a high amount of overlap between those communities and probably re- duces their usefulness	27
3.6	Degrees of separation in learning object repositories	31
3.7	An (incomplete) draft for an example taxonomy to categorize and describe learning objects with main categories and subcat- egories	35
3.8	Assigning users to communities	36
4.1	Searching and refining the search results	41
4.2	Community built by the "used author"-relation	46
4.3	Data model	49
4.4	User interface for the search prototype	51
4.5	Display of the search results	53
4.6	Additional result after use of an extend algorithm	53
5.1	The first three results found by filtering based on the assigned Strong Author Community	65