TILBURG UNIVERSITY

Department of Information Management

The Link Management Interface

Date	12th October 2005
Document ID	MACS-2004-01
Version	4.1
Status	Final
Authors(s)	Jeroen Hoppenbrouwers
Keywords	Link Database, System Architecture, LMI
Abstract	This document describes the fundamental ideas underlying the architecture and design of the Link Management Interface (LMI), part of the MACS infrastructure. From a user point of view, the basic concepts are explained and illustrated with some examples. Subsequently the technical design and implementation of the LMI is discussed in more detail.
Distribution	Public

Document Change Log

Parts of the document not yet completed are marked with a • symbol in the text and the right margin. Changes to the previous version are marked with vertical change bars in the left margin.

2005-02-15 Initial release for discussion purposes.

2005-03-38 Completed up to actual state of the database.

2005-07-21 Updated database part.

2005-10-04 Final distribution release for MACS partners.

2005-10-12 Final public release.

Contents

Chapter 1

Introduction

MACS aims to provide multilingual subject access to library catalogues. MACS enables users to simultaneously search the catalogues of the project's partner libraries in the language of their choice (English, French, German). The partners are: the Swiss National Library (SNL, project leader), the Bibliothèque nationale de France (BNF), The British Library (BL), and Die Deutsche Bibliothek (DDB). The project is running under the auspices of the Conference of European National Librarians (CENL).

Multilingual search is made possible thanks to the equivalence links created between the three indexing languages used in these libraries: SWD (for German), RAMEAU (for French), and LCSH (for English). Topics (headings) from the three lists are analysed to determine whether they are exact or partial matches, of a simple or complex nature. The end result is neither a translation nor a new thesaurus but a mapping of existing and widely used indexing languages.

On the basis of this approach, a prototype and later a production system has been developed by Tilburg University (Netherlands) which contains a significant subset of data from the indexing languages and the libraries' databases so that link creation and management and subsequent searching can be explored, tested, and used in a production environment.

1.1 Project Target

The four partner libraries each maintain an *authority list of subject headings*. These Subject Heading Lists (SHLs) are used to index library databases and therefore must be fairly static to be of practical use. Their different historical development and underlying natural language (English, French, German) make these SHLs unsuitable for cross-databases searches. It is not straightforward to access a library database that has been indexed with subject headings from one SHLwhile using subject headings from another SHL.

The aim of the project is to provide a way of linking these SHLs so that cross-SHL/cross-database searches become possible, and to develop both a system and a management approach to facilitate the maintenance of the links between the 'living' SHLs that are, and will stay, in continuous development.

Because the individual SHLs are crucial in the local (in our context, mostly national) database indexing and need to follow the demands of the local librarians and users, it is mandatory that each authority maintains *full independent control* over their SHL. There

can not be any attempt to centralize SHL (thesaurus) development. The management approach presented in this report acknowledges this given fact.

Lastly, the chosen management solution and database structure must be scalable to international (European) size, embracing hundreds of thousands of terms and a multitude of inter-SHL links with up to a hundred individual authorities contributing to the link management.

1.2 The Link Management System

The MACS link management system (LMI) is based upon a federated management approach, where each contributing authority maintains both their own SHL (using their own tools they have been using before) and a limited part of the federated link database. This link database is technically centralized, but accessible via the Web. It is therefore possible to manage it in a decentralized fashion. The basic assumption of our approach is:

> The links between subject headings in the Link Database are provided on a voluntary basis by the authorities, and should under no circumstances be considered permanent or final.

Therefore, the Link Database can be used by each authority to find links between SHLs, and those links are reasonably up to date and at least somebody found them useful. However, the Link Database is never guaranteed to offer a 'hard' mapping between SHLs. This is because of several environmental reasons:

1. All SHLs are in continuous development – they change.
2. Not all subject headings have a good mapping to any subject heading in another SHL.
3. There is no editorial staff that checks and acknowledges all links between subject headings which are provided by the individual authorities.

However, there is an analogy to a *peer review process* of the Link Database that makes it possible for links to be improved in time. Chapter 2 expands on this process.

1.3 Documentation Structure

The link management process is described in detail in Chapter 2, including the organization of people and authorities in a federation for mutual benefit.

Chapter 3 describes the negotiation process that takes place between the authorities to establish mutually acceptable links, and the way the LMI provides software support for these processes.

Chapter 4 presents the Link Management System (LMI) that supports the federated approach outlined in Chapter 2.

Chapter 5 contains a technical description, from different points of view, of the actual Link Database and the design of important parts of it.

Chapter 2

Link Management

The main purpose of the MACS project is to prove that there is a way of linking up individual SHLs in such a way that:

1. Each SHL maintains its independence.
2. All SHLs can be searched/browsed as a single, big, virtual SHL.
3. The management of the system is scalable and distributed over the participating authorities.
4. The system and the process can be used in a production environment.

This chapter discusses the basic ideas behind the Link Management procedure and the Link Database. It presents the choices we made while designing the infrastructure on which the LMI and MACS have been built.

2.1 Linking Philosophy

For the system to work well, it is mandatory that the individual authorities retain full responsibility over their own SHL. This prohibits any type of unified or merged database in which all SHLs are stored together. Such an integrated system would be easier and more consistent to use, because it can be tightly controlled by a central editorial authority, but it is unfeasible for the individual authorities to give up their freedom in changing and managing their own SHL. Previous experiences in database integration have tried to maintain the individual aspects of the contributing databases, but never came to a satisfying solution.

The complete system (see Figure 2.1) has several distinct databases that are interlinked. At the periphery of the system are the various SHL databases, which are completely maintained by their respective owners. The LMI will not have access to these databases, which therefore must be partially copied into the central system. However, it must be clear from the start that the final authority for each SHL cannot and will not be the LMI. It is the responsibility of the authorities to keep the (partial) copy of their SHL in the LMI up to date.

At the core of the system is the MACS Link Database, which contains additional data that can be used, on a strictly voluntary basis, to suggest semantic links between

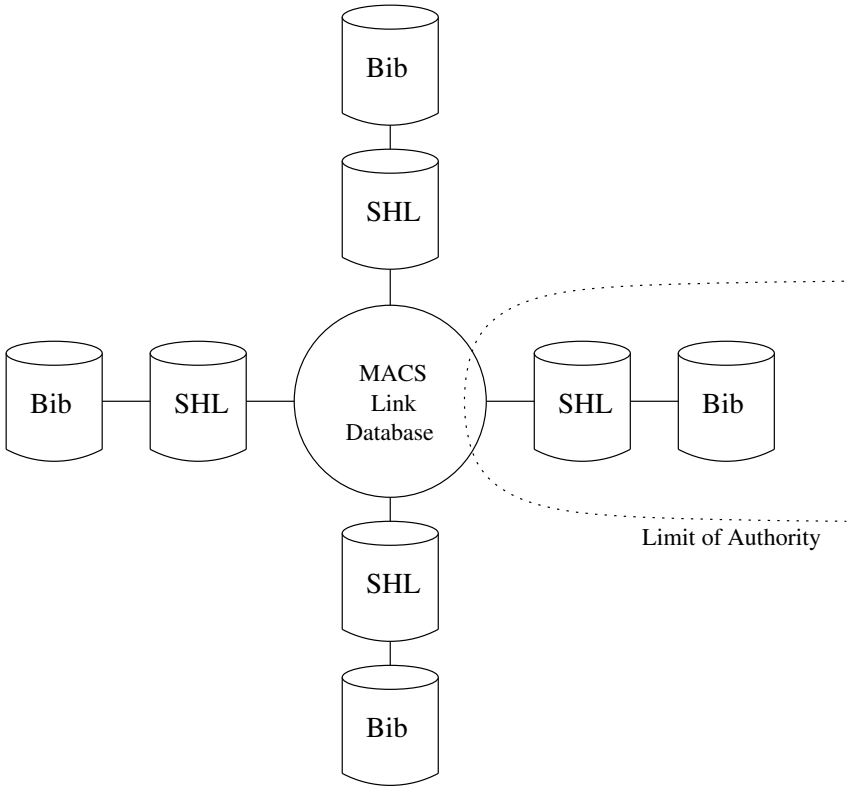

Figure 2.1: Architecture of the MACS System

various subject headings of the SHL databases. An authority's area of influence encompasses all of its own databases (SHL and the bibliographical databases indexed through the SHL), plus a small part of the centralized Link Database. Importantly, the Link Database *does not contain information that is already available in the individual* SHL*s*, although for performance reasons some redundancy might be introduced. However, the SHL stays the only authoritative source of information. Browsing the Link Database is no true replacement for browsing the individual SHLs.

The Link Database contains *Links*. A Link is a cluster of *Expressions* (queries, in a sense) in several subject heading languages (Figure 2.2). These Expressions can theoretically be any utterance that would be meaningful in the respective SHL. By placing some Expressions together in a cluster, the system is given the knowledge that these Expressions can be substituted for each other without a *significant* change in meaning of the query. The Expressions, from a user point of view, are interchangeable. Of course each Expression is intended to be used with a specific SHL and underlying database.

Each Expression contains *Terms* (subject headings) from the SHL and operators to connect them, typically AND. The great majority of Expressions will consist of a single Term.

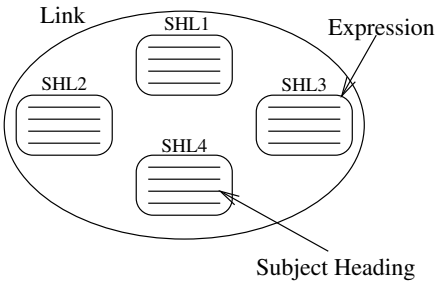

Figure 2.2: A Link between four Expressions

2.2 Link Semantics

The LMI only supports one semantical link type, that of *approximate equality*. We restrict the database to this type of link because its only purpose is to provide a 'bridge' between several SHLs. These SHLs may contain more complex semantical relationships between their subject headings,[1] and it would be a violation of the 'no redundant data'-rule to add this information to the Link Database. In a sense, the Link Database should contain *no semantics at all*, just blind references to subject heading language expressions in external SHLs.

Scientific applications of thesaurus technology such as (Euro)WordNet[2] also consider the synonymy relationship the most important one, and only add more refined relationships later. Their main semantical building block is the *synset*, a set of semantically (almost) equivalent terms.

In the context of the MACS Project, we say that two SHL expressions are 'approximately equal' when:

> The expressions are SHL queries that return the most equivalent results possible, given the information stored in both the databases and the SHLs.

This means that the expressions are not necessarily equal in (linguistic) meaning. We do not consider this a problem. The Link Database is intended to provide a bridge between potentially non-matching SHLs, and under such circumstances any reasonable mapping is better than no mapping at all. Subject heading languages are *not natural languages* and we should emphasize this difference.

2.2.1 Single-expression Links

It is a matter of policy whether Links with only one expression are allowed in the Link Database. A Link with one expression does not link anything, so logically it should be disallowed. However, many SHLs contain subjects that do not at all map into other SHLs, and the availability of single-expression links in the Link Database gives a strong clue about the white spots on the map. Filling in the white spots, either by locating appropriate subject headings in other SHLs or by extending those SHLs, is an important management activity, which is much better supported when single-expression links are allowed.

[1] Semantic relationships such as narrower-term, broader-term, part-of, etc.
[2] `http://www.illc.uva.nl/EuroWordNet/`

The current production system allows single-expression links, and there is a mechanism available to list out all links in the system with no expression of the own SHL of the user.

2.3 Expression Semantics

Between simple expressions there is no need for expression semantics, since they consist of a single subject heading. However, more complex expressions contain multiple subject headings connected by operators. This section discusses the implications of compound expressions for the organization of the system.

2.3.1 OR Semantics

As an example, the following table shows typical one-subject-heading expressions, with each row containing a Link with three Expressions. Note that the equivalence is not *exact*, but only *approximate*. Linguists would likely not accept the equivalence.

LCSH	RAMEAU	SWD
Decathletes	Décathloniens	Zehnkampfer
Decathlon	Décathlon	Zehnkampf
Discus throwing	Lancer du disque	Diskuswurf
Divers	Plongeurs	Kunstspringer
Diving	Plongeon	Wasserspringen
Hammer throwing	Lancer du marteau	Hammerwurf
Hurdle racing	Course de haies	Hürdenlauf

The previous table contains simple subject headings only, where relationships are straightforward. In the next example, a more complex situation is sketched. There is a significant difference in scope of the used subject headings, leading not to one but to two expression clusters (links) which partially overlap.

LCSH	RAMEAU	SWD
Jumping	Saut en hauteur	Hochsprung
Jumping	Sauts (atlétisme)	Sprung

Here the English (LCSH) subject heading *Jumping* has a broader scope than the French (RAMEAU) and German (SWD) terms *Saut en hauteur/Hochsprung* and *Sauts (atlétisme)/Sprung*. Therefore, when looking for search equivalents of *Jumping*, the system needs to provide two French and two German terms that must be combined by OR in order to cover the whole subject.

While searching with *Hochsprung*, the French term is 'better' (narrower) than the English one, but in the LCSH there simply is no better choice.

It is also possible to explicitly combine the RAMEAU and SWD terms into a native SHL expression for the equivalent of *Jumping*. The advantage of this method is that we now have three concepts with clearly separate meanings, instead of deducing a virtual third concept like in the previous example:

LCSH	RAMEAU	SWD
Jumping	Saut en hauteur	Hochsprung
Jumping	Sauts (atlétisme)	Sprung
Jumping	Saut en hauteur OR Sauts (athlétisme)	Hochsprung OR Sprung

Such a link, with more than one subject heading in one of the expressions, is said to be a *compound link* with at least one *compound expression*. Note that leaving the first two lines out and reasoning that the third line contains all information anyway is wrong: the link between *Saut en hauteur* and *Hochsprung* would be lost, and replaced by a more generic link to *Hochsprung OR Sprung*.

Implicit OR versus explicit OR

We had to make a decision whether to use the implicit or the explicit OR. Advantage of the implicit approach (two links for *Jumping*) is the space saving in the database and the cleanest knowledge representation. Disadvantage is that the system must itself combine the RAMEAU and SWD headings into compound expressions when a user enters the database with *Jumping*.

The explicit approach costs a bit more disk space, and more work to enter. However, it provides the users with more fine-grained tuning capabilities and also makes it clear that there are, actually, *three* concepts involved: the generic jumping, high jumping, and both together. On top, the LCSH also recognizes *broad jumping*. This brings us close to the next step, which would be to also add the knowledge that broad jumping and high jumping are both special cases of jumping. However this would be a bridge too far. We do not want semantical information in the Link Database. In this light, the explicit approach might already be too explicit in terms of semantics.

A possible solution could have been to use the expression qualities to prefer some links over others, given the entry point. In this case, the entry points could be sorted by quality.

LCSH	**RAMEAU**	**SWD**
Jumping (0.70)	Saut en hauteur (1.00)	Hochsprung (1.00)
Jumping (0.70)	Sauts (atlétisme) (1.00)	Sprung (1.00)
Jumping (1.00)	Saut en hauteur OR Sauts (athlétisme) (0.70)	Hochsprung OR Sprung (0.70)

However, using link qualities for automatic decisions is a fuzzy process and should be avoided where possible.

There are cases where only one SHL needs an OR operator:

LCSH	**RAMEAU**	**SWD**
Runners (Sports)	Coureurs	Läufer
Runners (Sports)	Coureurs	Langstreckenläufer

In such a case, it is more convenient to group the SWD entries together (*Läufer OR Langstreckenläufer*), avoiding duplication of the other SHL entries. But as soon as another SHL is added which also makes a difference between running and long distance running, the link *must* be split into two links like in the previous table. This splitting process (mostly a copy action) should be automatic. Collapsing a split link into a compound link is less likely to occur.

The bottom line is that the implicit (split links) approach works, and is a lot simpler in the end. The MACS LMI has been built around implicit OR. This means that inside expressions the OR operator is not allowed, and that link records will need to be partially

copied in such cases. But this approach avoids the combinatory explosion of changes when a single new SHL is added that happens to have more specific subject headings available.

2.3.2 AND Semantics

For an example of the AND operator, look at the following table.[3]

LCSH	RAMEAU	SWD
Child actors	Enfants acteurs	Kind AND Schauspieler
Actors	Acteurs	Schauspieler
Children	Enfants	Kind

It can be seen here that the German SHL cannot express the narrow concept *Child actors* with only one subject heading. Two subject headings must be combined with AND in order to arrive at the intersection of them. This is the reverse operation of the OR-combination in the previous examples. OR is used to combine specialized concepts into a more generalized one, AND is used to make a more specialized concept out of several generic ones.

Unlike the OR-examples, the Link containing *Child actors* cannot be deduced by the system, since the system has no knowledge about the subject headings – it cannot and should not try to parse the meaning of *Child actors*. Therefore, the German expression *Kind AND Schauspieler* must be added manually.

The LMI system only supports one single (AND) operator per compound expression, i.e., all subject headings in the compound expression are connected by the same (AND) operator. No parentheses, multiple operators, ORs or Boolean NOTs are allowed, just AND.

2.4 Subject Heading Semantics

As said before, the system does not assume *any* semantics for the subject headings. It treats subject headings as completely alien strings, not to be touched at all.

The Link Database contains unique subject heading identifiers, supplied by the respective authorities, that will refer directly to original SHL entries. There is no technical need to maintain a local copy of the orthography of the subject heading. For system performance and maintanance-friendliness, however, the original subject headings are copied into the Link Database. The external SHL will forever stay the only true authoritative resource for subject headings.

The Link Database maintains only one record for each individual subject heading of each SHL, so that changes in the SHL (like a change in orthography) can easily be followed. This centralized record also facilitates the display of all expressions, and therefore links, in which the subject heading participates.

[3]The subject headings *Actors, Children, Acteurs,* and *Enfants* have been added to complete the example. They do not necessarily exist in the SHLs, and their addition is fully at the discretion of the authority making the link.

Chapter 3

Workflow, System Access, and Rights Maintenance

This chapter explains in detail how the management of the system is organized and how the individual SHLs are virtually merged in a federated database, with specific procedures for conflict resolution.

3.1 Workgroup Organization

The mandatory independence of the contributing SHLs calls for a federated approach to the maintenance of the Link Database. In such a federation, each partner (authority list maintainer) has the right to pull out at any time, which means that the Link Database must never rely on any SHL being available. It also means that each partner is individually responsible for the maintenance of the links from/to the central (federated) Links and his own SHL. Although collaboration between partners is encouraged, it must be possible to operate the Link Database without such collaboration.

It is completely up to the authorities that maintain a SHL to decide what subject heading ID (authority number) and which readable text they put into the Link Database, and in which Link they put it. Of course, if the partners agree on a standard set of links or any form of central (editorial) management, this is possible, but not required. Because the contributing authorities are individually responsible for the links between the federated links and their own subject headings, they will individually benefit from the work they put into the federated database. The more work they put into it, the better their SHL connects to the others (both ways), so there is an incentive to do it right.

The next paragraphs discuss the tasks and authorities of individuals, federated groups, and supervisors from a high level. The next section goes into implementation detail.

3.1.1 Individual Actions

All participating authorities can *add* Links to the Link Database. Authorities are free to create new Links or to add new Expressions to existing Links. It is possible for an authority to add Links between SHLs that are not controlled by him; somebody from the Swiss National Library is allowed to create a link between subject headings of the Bibliothèque nationale de France and Die Deutsche Bibliothek.

Only editors of a SHL can *lock* expressions containing subject headings of their SHL. Each expression has an independent 'Lock' switch that can only be operated by the SHL authority the expression refers to. So, in the above example, the Swiss National Library can add two expressions but cannot lock any of them. The expression for Die Deutsche Bibliothek can only be locked by somebody from Die Deutsche Bibliothek, etc.

Authorities are expected to eventually review all unlocked expressions pointing into or out of their SHL and where appropriate to lock them.

SHL editors also have the right to *delete* any expression in their SHL. If another authority creates an expression that is considered totally unacceptable, it can be deleted by the appropriate authority without further negotiation.

Likewise, it is the authority that has the final word over the subject heading record. Although other parties can create such a record, filling it in with data from the SHL (possibly this is done by the system), only the authority 'owning' the subject heading can modify it. But the creation right makes it possible to add a link to the system which contains subject headings not yet in the local SHL copy.

3.1.2 Federated Actions

The federated model outlined above provides for largely independent updating of the Link Database by the individual partners, where nobody is required to put in work unless they want to gain extra benefits. However, some central management can be added at the partners' discretion. It is also possible to have more than one Link Database,[1] so that the management can be distributed over several central bodies according to subject field or other criteria. There is one, standardised management interface, but there is no need for standardised organisational management between Link Databases.

In case a centralised management effort is called for, organisational negotiations will need to take place about the central body's mandate and task. This body could play the role of an editorial board, with responsibility for the homogeneity of the Link Database. Given the nature of the international collaboration, the cultural differences will likely be considerable, and a wide spread in the levels of ambition and quality assurance by the different partners must be expected. Our approach allows the consortium to experiment with different levels of centralised editorial control, ranging from anarchy to totalitarian. The editorial staff could even double as a telephone/E-mail hotline and consulting service for the different authorities, ensuring that all involved parties share roughly the same ideas of how to do things.

Independent of the availability of centralised management, all federated authorities have access to the full database and can freely add links and delete links 'owned' by them. The central organization offers supervision services in good cooperation with the authorities. The federation model assumes a positive attitude towards cooperation from all members. There is no 'link police' in the current architecture.

3.1.3 Conflict Resolution

Despite the open model of cooperation, conflicts between parties surely arise. Most of these conflicts are rather mundane, at the level of (dis)agreeing about the validity of a proposed link. Since each authority can add links to/from other authorities, which

[1] Several virtual Link Databases can still run on the same server. It is possible to partition a database into smaller virtual databases as well.

means a *proposal* for a link, there will be cases where some authority does not agree with the proposal. The authority to whose subject heading a link member points has the final word over the validity of the proposal, by locking or deleting the expression.

However, with more than one party involved, possibly including a centralized management body, communication only by means of link addition, authorization, and deletion is certain to fall short. Next to the Link Database but tightly integrated, there is a discussion and negotiation system through which parties can communicate. It is easy to add annotations to links and to add comments to these annotations, to escalate discussions about a link into a bigger group, and to moderate these negotiations.

3.2 Work Flow Implementation

First we give a brief description of the basic ideas behind the LMI work flow implementation, followed by the detailed implementation in various parts of the system.

One partner (authority) is responsible for one SHL, and may assign work to other institutions or colleagues whenever required. Changes in the system that affect an authority can only be made by that specific authority. Authorities may *propose* additions and changes to other authorities, but these propositions are subject to review and approval. This system is referred to as "Trust and Control:" within an authority, all changes made by 'own (trusted) people' are considered valid, while cross-authority changes are subject to approval by the affected authority. The only way to propose changes to an already existing and approved (locked) link is by attaching an annotation.

It is absolutely mandatory that the scalability of the system is guaranteed. In particular, existing data should not require re-validation when another SHL is added to the LMI system. Only when a new authority proposes changes in existing data, or adds new data for another authority, explicit approval is required. In other words: the existing link base is considered correct and consistent until further notice, and adding a new SHL's subject heading to an existing link cluster cannot invalidate the cluster. Responsability for the correctness of the new subject heading is with this heading's authority.

Another very important scalability feature is the *Domain*. Domains are semantic areas which can be assigned to links (not to subject headings). There is a list of about 100 domains, that is fairly static, out of which the domains must be selected. Beyond the LMI, the domains are not valid or useful; they are not exported with the links to the Search Interface. They only serve as area restrictors. Each annotator/editor has one or more domains assigned to his/her actor profile, and by default only the system events within these domains are presented. For informational purposes, the domain filter may be bypassed. However, editors and annotators may only update/annotate links within the set of domains assigned to them.

The LMI has several *roles* of people who work with the system (more information about roles and actors will be given in Section 3.3.1). One of the roles is 'SHL editor.' All editors of one SHL have access to a *To Do* list with items that are mandatory to be reviewed or corrected, and a *To Know* list with recent changes. Several mechanisms are in place to limit the amount of information shown on these screens, to keep the system manageable and scalable.

3.2.1 Domain Filtering

It is important to note that there are two domain filters. The first filter determines which links can be *modified* by the editor. This filter is bound to the login account, and can only be changed by a system supervisor. The second filter determines which links are *visible* to the editor, and can be changed by the editor herself. This filter is used to reduce the amount of information presented to the editor during normal work. Directly after login, the presentation filter is set up to be the same as the modification filter.

The presentation filter is accessible from the *Search for Subject Heading* page, by clicking on 'Change' in the domains clause of the search details. A simple three-column page is shown where the editor can select which domains are to be presented. The 'Default Filter' button resets the filter to be equal to the modification filter. On the Search page there is also a selection option to show all domains without any filtering.

Some actors, such as SHL administrators, do not have a modification domain filter and therefore their default presentation domain filter is empty. They can still reduce the amount of visible links by setting their presentation domain filter.

For all display functions where a domain filter is applicable, the current domain filter is used.

3.2.2 Annotation Management

All roles of *Annotator* and higher (see Section 3.3.1) are allowed to enter annotations in the system. Annotations are always attached to a link, not to a subject heading (term). The creator of an annotation can indicate the urgency of an annotation by means of a three-way switch, from least urgent (least disturbing) to most urgent (many people will notice):

1. *Not on any ToDo list*. This is a generic annotation with the link, intended for general information to all people interested in the link. It can only be added to links in a domain that is in the annotator's domain set, issued by the supervisor. Annotations of the kind *Not on any ToDo list* are not part of the work flow. They do not show up on any *To Do* list, and do not need to be 'confirmed' or 'answered' by anybody. However, they will remain with the link, and their addition can be seen via the *To Know* page.

2. *To* <SHL> *admin only*. This can be done for each link, even if the annotator does not have the link's domain in his/her domain set. The annotation will only appear on the *To Do* page of the SHL administrator. The annotator selects the name of one authority from a list. The administrator can decide to ignore the annotation (take it off her *To Do* list), or to scale it up to the relevant SHL editor group, after which the annotation becomes *To all* <SHL> *editors* and appears to all relevant editors of this one SHL.

3. *To all* <SHL> *editors*. For links in the annotator's domain set, annotators can address an annotation to all relevant SHL editors, by selecting the name of one authority from a list. 'Relevant' here means that both an editor's SHL and domain must match that of the annotated link for a message to be shown in the editor's *To Do* list. This type of annotations persists on the *To Do* lists for as long as nobody explicitly reduces its urgency.

Since the LMI supervisor has nothing to do with system content, it is not possible to address annotations to the supervisor. Neither is it possible to address an annotation to all editors of all SHLs, as this would not scale well.

3.2.3 Link Review Management: 'Link Locking'

The prototype has taught us that the link review model employed there was too restrictive. For information, it was implemented such that *any* change to *any* part of a link invalidated all (SHL) parts of the link, so that all respective editors had to come back to review the change. Although this, technically, was not wrong, since the change could indeed have made the link meaningless, it generated too much checkup work for the editors. Also, it would be prohibitive when a new SHL would be added to the system: that would in one stroke invalidate all existing links. Hence, the introduction of the *Trust and Control* concept (Section 3.2).

The LMI relaxes the notion of (in)validation and replaces it by a locking mechanism. The first change is that each editor of a particular SHL can at all times change the part of the link that (s)he 'owns.' This is never signaled to anybody, except for the *To Know* list which logs all recent changes in the system. The second change is that nobody can touch the parts of a link that (s)he does not 'own.'

This works well for existing links, but would block creating new ones and proposing link parts of other people's SHL to gain time. So for *new* links, which can be created by any editor, the 'lock' on the parts that (s)he does not 'own' is lifted. For example, if an LCSH editor creates a new link, she can fill in the LCSH part because she 'owns' it. Additionally she can also fill in the RAMEAU and SWD parts, as *suggestions* to the respective SHL editors. These proposed link parts are signalled to the RAMEAU and SWD editors: they appear in their *To Do* lists. This is caused by their unlocked state combined with the presence of content. When these editors come around to review the proposed parts, they can edit them as required, and then *lock the link part*. After locking the RAMEAU part, nobody but RAMEAU editors can touch it any more. If the original LCSH editor does not agree with the editing of her proposal, the only thing she can do is to add an annotation (addressed to the RAMEAU editors).

As as special case of link locking, the part of a new link that falls under the jurisdiction of an editor will be automatically locked, so that there is no need to first add the link part and then explicitly lock it.

The lock state of link expressions after a batch load is not predetermined. This depends completely on the situation. In some cases, validated data will be loaded which warrants a locked state. In other cases, the LMI will be used to validate loaded data, and therefore the batch loader might need to deliver links unlocked. The only proper solution is to give the batch loader an option to lock or not lock a link.

3.2.4 Work Pending: 'To Do'

A very important work flow management tool is the page that shows work pending, also known as *To Do*. This page is not configurable. It always shows the items that are 'open' for the given actor, depending on his/her role and SHL authority. As long as there is mandatory work to do, it will stay on this page. In order to prevent this page from overflowing, a maximum item limit can be specified using the `TODO_ITEM_LIMIT` setting in the LMI configuration file. A typical scenario would be that a new SHL gets added by a batch load, and that for one reason or another the locked state is not set by the loading process. This would cause a very long 'To Do' page to appear, possibly so

long that it causes a system malfunction somewhere. The maximum item count will prevent the first problem (unlocked mass additions) from cascading into the second one (LMI malfunction).

For a SHL editor, the following items appear on the *To Do* page, with a maximum item count per group and a clear warning signal in case there are more items than the limit allows.

- Invalid SHL term identifiers in relevant domains. Note that only links are associated with domains, so SHL terms that linger deep in the LMI vault which are not used by any link will not disturb the editor. Neither do terms that do not belong to his/her domains. However, by setting the domain filter page to 'all domains,' they will show up.
- Annotations to SHL editors in relevant domains. These annotations are addressed to the group of editors, filtered by domain. By resetting the domain filter, other domains' annotations become visible. Annotations *Not on any ToDo list* will never appear (they can be found on the *To Know* page by date range, or by visiting the appropriate link directly).
- Unlocked link parts for his/her SHL filtered by domain as above. These are 'open' link expressions that have been entered by editors not from the SHL, as proposals. They need to be reviewed, and locked when correct.

For SHL administrators, the list above is copied and extended with:

- All domains selected by default, so domain filter overridden.
- Annotations addressed to the SHL admin (added by annotators not a member of the SHL).

Since the *To Do* list contains mandatory work, i.e., items that require correction, review, or an answer, it can not be limited to a date range. Ideally this page is always nearly empty, and it definitely should be empty for any consolidated (read-only) database such as the production database.

Contrarily, the *To Know* list will nearly never be empty, as this contains a personalised log of system activity in a working database. The exception is, of course, a read-only production database. There is still some discussion about the batch transfer of links with open annotations from work to production.

3.2.5 Interesting information: 'To Know'

Inquisitive people might want to look beyond their own mandatory work as dictated by the *To Do* page and their boss, and visit the *To Know* page. What appears here is never meant to be taken as 'must do,' but as 'nice to know' instead.

Since the list will get very long, it is required to have some sort of limiter on it, and even a very restricted one by default. A date range filter is a proper solution. It comes up with a one day time span, extensible by the actor to, if required, several months back in time. On top, the standard domain filter applies as in the *To Do* list. If the number of results still is higher than the `TOKNOW_ITEM_LIMIT`, the LMI will truncate the list and put up a message. The actor may lift this restriction if (s)he really wants to. There is also a `TOKNOW_ABSOLUTE_LIMIT` which can never be exceeded, even if a user asks for such a long list.

The *To Know* page shows the following items, filtered by date (of last modification to the link or annotation) and domain, limited by maximum item count:

- All links that do not have a part of the actor's SHL filled in. This is to signal possible gaps in link coverage.
- All links that have an actor's SHL part, and have been modified in any way. The action of modification is reported, not the state.
- All links that have an actor's SHL part, where the actor's SHL part has been modified by some editor not from the SHL. Note that this is only relevant for links that were unlocked at the time of modification. And again, only the action is reported, not the state.
- All links that were locked in his/her own SHL. Action, not state. This does not show all currently locked links.
- All links that have a part of the actor's SHL that have been annotated. Action, not state. Does not show all annotated links.

All these items are individually selectable by means of check boxes, and are limited by a maximum item count that can be overridden. The *To Know* page has the ability to show very large amounts of data and the actor is responsible to make reasonable requests.

3.3 Access Control Implementation

The issue of Access and Rights Maintenance has been debated a couple of times before, but it was only after some implementation work that a clear picture emerged. The following is an outline of the generic Access and Rights System that has been implemented in the LMI.

In order to minimize confusion, we speak of *actors* when we mean the operators of the LMI (mostly MACS partners), and of *users* when we mean operators of the Search Interface (mostly interested fellow librarians or real library end users).

Access to the LMI is controlled by individual logins. There are no shared (group) logins, except for common read-only access by guests.

3.3.1 Actors and Roles

Access rights and privileges are managed by a role-based system. Functional roles in the LMI have been established, and people are granted rights according to 'the role they play.' As a special case of role-based access, the LMI has a strictly hierarchical role system, where the same person can 'play' all roles lower in the hierarchy than his assigned role. There can be only one assigned role, the top role, per person.

Instead of granting access per SHL, such as in the prototype, the LMI system grants access based both on SHL and on domain. This leads to a cross matrix with access to rows (SHLs), columns (domains), or cells (intersections of a SHL and a domain). Such a system can grant access on the basis of a cell (for normal actors), or a whole row (for SHL administrators). The potential option to grant access on a complete column ('domain administrator') is not useful and has not been implemented.

	100	110	120	130
LCSH	x	x		x
RAMEAU			x	x
SWD	x	x	x	

Table 3.1: Access Rights Matrix

The following roles have been implemented. All roles subsume the rights of the previous roles: an annotator is also a reader, etc. Please note that the actor and role administration is linked to a particular database; it is very well possible that the same actor exists in more than one database, but with different roles and rights.

blocked This is a special role, used to disallow access to the database by this actor. A blocked actor is equal to an actor that does not have a valid password. The other details of the actor remain stored, which is useful for editor reference of a link.

reader Read-only guest access to the whole database, not restricted to any specific SHL or domain.

annotator May add annotations only. Write access restricted to exactly one SHL and several domains (multiple cell access in one row).

editor May create and change links. Write Access restricted to exactly one SHL and several domains (multiple cell access in one row).

admin All functions within one SHL (full row access; no domain restrictions).

super Supervisor, has access to all possible functions including actor maintenance and link deletion.

From an organisational point of view, these roles can be mapped to specific people. Readers are typically guest actors, who want a glimpse under the hood of the LMI system. Annotators are external reviewers whose opinion is asked on LMI content. Editors work on a specific SHL and domain set, and are responsible for their own SHL as a group. The group is headed by the SHL administrator, who functions as the decision maker when the group does not reach conclusion, and can also override all actions taken by one of his/her editors. Specifically, only the administrator may modify a link's domains. There is usually one administrator per supported SHL. The system supervisor does not have a content role, but takes on the administrative duties to keep the LMI running. There can be more than one supervisor if required, but their number should be limited and they should stay in close contact. Normally, a supervisor is not concerned with the link content of the LMI.

Again, be aware that in different databases, different role and rights assignments may be in effect.

3.4 Maintenance Interface

All actor access and rights maintenance is carried out by supervisors only, using a dedicated maintenance interface, accessible via the *Maintenance* selection in the main LMI menu. For a sketch of the pages, that does not include all fields for clarity reasons, see Figure 3.1.

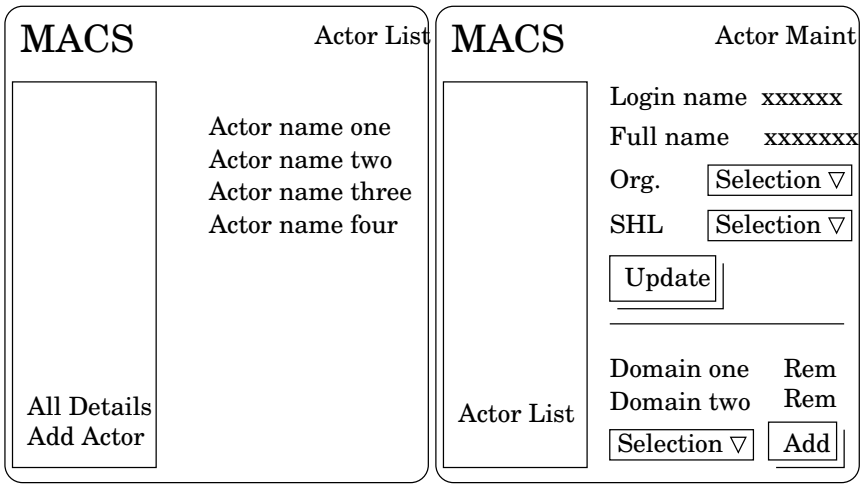

Figure 3.1: Sketch of Actor List and Actor Maintenance Pages

3.4.1 Actor Maintenance

First page to appear is a complete actor list, displaying the full actor name and a few other details (limited by available typical browser space). The page title is *LMI Actor List*. It is a simple table, consisting of one actor per row. The actor login name is clickable and links to the actor maintenance page, described below. On the right is a *switch* link for each user. Supervisors can use this link to switch to the identity of another user, without knowing the associated password. This is very handy for system testing and support, without causing a security hole. Part of the page (in the left menu bar) is an *Add New Actor* selection.

A followup page page shows all available details of one single actor and allows for editing of most of them (by a supervisor). The top panel of the page contains the fixed fields, such as actor login name, full actor name, the related organisation and authority, and the role selector. When a new actor is added from the Actor List, all items in this panel come up blank or with a default. There is one *Update Actor* button that either adds or modifies the values for the given actor. There is no *Delete* option, as removing a actor would mean that all the related data (changed by, annotations, etc.) would either disappear or become orphaned. Actors that should be disallowed future system access need to get the role *Blocked*. This deletes their password, so an actor that gets unblocked needs to get a new password issued by the supervisor. The password system is standard: the supervisor can set a new password, but it is technically impossible to read an existing password (they are stored one-way MD5-hashed in the database).

The Actor Maintenance page also shows an Authority selection to indicate the actor's SHL. This is only applicable to annotators, editors, and SHL admins. Readers and supervisors do not use an associated SHL; but one can be assigned anyway, since people can change roles.

Likewise, for all 'annotator' and 'editor' actors (and also for the other roles, but there they are ignored), there is a *Domains* panel that should contain at least one domain. Since an annotator or editor without any domain does not make sense, but does not cause system malfunctions either, the domain list may technically remain empty.

3.4.2 Transliteration Table

The transliteration table is an essential element to support cross-character-set sorting and searching. The LMI database is fully Unicode (UTF-8) compliant, so it can store virtually any character from any language. However, Unicode does not offer a standard sort order that is logical and meaningful to people. It has not been internationally standardised whether certain Japanese characters appear before or after Cyrillic ones (national standards obviously do exist). With the transliteration table, the LMI offers a concept to reduce this sorting problem.

Next to each Subject Heading entry in the LMI, which is stored in UTF-8 as entered by the actor, the LMI keeps a transliterated copy of the entry for sorting and searching. The actual transliteration can be viewed at any time by looking at the *Subject Heading Details* page for a subject heading. Basic transliteration is done by removing all diacriticals (é becomes e, ç becomes c, and so forth), and making all individual characters lower case where appropriate. For this job we use the GNU recode package,[2] which provides standard transliteration from nearly every known character encoding to nearly every other encoding. In this case we transliterate from UTF-8 to flat, US 7-bit ASCII.

Not all characters are correctly converted by GNU recode, and in some cases you might want to help or correct the package. For this purpose, the LMI has a transliteration table per database. You can review the table via the *Maintenance* choice if you are a supervisor. The transliteration table is a simple list of 'to' and 'from' entries, where you can fill in which single characters should be rewritten to which other characters. In some cases, you want to replace a single character with *two* characters, such as the 'æ' and maybe the 'ß.' Whether or not you want to rewrite the German umlaut characters into 'ae' or 'oe' is your own choice. We advice not to do it, since the LMI does not know about language-specific rules, and the best approach to transliteration seems to be a very straightforward, logical, consistent and therefore predictable 'remove all accents' approach. Please take note that the LMI is not a library system; as long as the resulting sort order is predictable, trained people will have little trouble finding the entries they are looking for.

On the transliteration page, you can enter any UTF-8 expression and see what the transliteration would do to it. You can also re-transliterate the whole term database, which obviously takes a while, but can be run as often as you wish without interrupting the LMI. You may also interrupt a re-transliteration run and re-run it at a later, more convenient time.

Search operations are done on transliterated input. This means that, for example, the input 'Théâtre' is transliterated to 'theatre' and subsequently matched to the transliterated terms. In other words, the matching is done on the minimal form, with all accents removed. This assures that the retrieved terms have the best chance of corresponding to what the actor really wanted to retrieve.

3.4.3 Bulk Uploads

It happens regularly that external bulk data needs to be fed into the LMI. Typically this is raw SHL data (subject headings with authority numbers), sometimes also complete links. The LMI has a few canned upload routines aboard to accomodate these bulk operations. It is strongly suggested that future upload routines are added to this menu, so that there is a common point of access.

[2] `http://www.gnu.org/software/recode/recode.html`

When writing this documentation, the only bulk uploader available is for 'Combined RAMEAU-LCSH UNIMARC, dumped to line-delimited UTF-8.' Typically this is the format as delivered by the Bibliothèque nationale de France. The code for this particular bulk loader, which can be used as a template for more bulk loaders, can be found in the `upload-001.html` file. In the directory `/usr/local/share/rameau` is a copy of the original RAMEAU data as delivered to Tilburg University, the converted files in UTF-8, and the software used to do this conversion. We used `marcdump` (part of the YAZ Z39.50 toolkit[3]) to go from UNIMARC to line-delimited records, and a patched version of GNU recode to further convert to UTF-8. These last files were fed to the bulk loader via a web browser.

A bulk uploader typically offers a file name entry box to feed it the raw input file via your web browser, a log level selector to select the amount of diagnostic output, and a checkbox to tell it whether you really want to update the database or only to run the diagnostics on the input file. It is strongly suggested to first run a few heavy diagnostics before attempting to add the new data to any database. The 'Error' log level only flags unrecoverable errors, 'Warning' the unrecoverables plus potentially erroneous situations, 'Normal' all of the above plus progress messages, etc. Not all bulk uploaders necessarily support all log levels.

3.4.4 zThes Bulk Downloads

The export of LMI data to the Search Interface is done offline via a (modified) zThes dump. The user can select which authority to export, and the database to export (the system has been designed to allow external programs to request these dumps, too).

Exporting to zThes is a lengthy process and it depends on the intended purpose whether additional features will be required, such as delta export only (incremental exports). The zThes format itself is not truly stable yet either. See Chapter 5 for the XML format.

[3] `http://www.indexdata.dk/yaz/`

Chapter 4

Link Management Interface

To stay consistent with the rest of this manual, the word user *will be replaced by* actor *when we talk about the* LMI*, but remain unchanged when we talk about the Search Interface.*

The main purpose of the Link Management Interface is to offer actors (librarians maintaining the SHLs and Link Database) a clear, aggregated, helpful view of the stored links. The relational model underlying the Link Database is far too technical; it is machine-friendly, but definitely not human-friendly. The Link Management Interface is the bridge between the machine-level data storage and the human-level link management. The zThes communication record is in between, but more on the aggregated link level (human) than on the normalized data level (machine).

The Link Management Interface is not the human interface for retrieval purposes. The 'Search Interface' is a separate system.

4.1 Actor Model

The interface assumes the viewpoint of an actor who is responsible for the maintenance of *one particular* SHL and its relationships to other SHLs. Entry into the system is usually with a subject heading from a single SHL, usually the SHL the actor 'owns.'

The system collects all occurrences of the given heading (i.e., every expression in which the heading is an argument) and presents them in a table which shows all Links and Expressions. For a good example, we made up some fictitious subject heading entries to show the various combinations.

LCSH	**RAMEAU**	**SWD**
Theater – Financial awards	Théâtre AND Prix et récompenses	Theater AND Kulturpreis
Theater – Non-financial awards	Théâtre AND Prix et récompenses	Theater AND Kulturpreis

The actor creating these two links decided to create separate meanings for the LCSH entries, under the assumption that other SHLs would be added in the future that would directly map to both meanings. The prototype supports only this 'implicit OR' approach.

When entering the system from the RAMEAU side, using the heading *Théâtre*, the system would produce all RAMEAU headings in which the given heading is the argument of an expression. It is important to note here that *true subheadings are not shown* since the system cannot 'look into headings,' and subheadings are a different heading (using '–' within the heading instead of 'AND' to connect two separate headings). Searching for *Théâtre* would retrieve links such as *Théâtre AND Biographies*, but not *Théâtre – Distribution*. The last (sub)heading would categorize alphabetically under Théâtre, but not semantically, unless there is a link between them in RAMEAU itself and the Link system would be able to tap into this (which it is not). It is a matter of actor preference whether (s)he wants to search the Link Database with *Théâtre* (exact matches only) or with *Théâtre*$*$ (alphabetically truncated, to see the subheadings as well). With just *Théâtre*, the system would produce:

RAMEAU	**SWD**	**LCSH**
Théâtre	Theater	Theater
Théâtre AND Bibliographie	Theater AND Bibliographie	Theater – Bibliography
Théâtre AND Biographies	Theater AND Biographie	Theater – Biography
Théâtre AND Prix et récompenses	Theater AND Kulturpreis	Theater – Financial awards
Théâtre AND Prix et récompenses	Theater AND Kulturpreis	Theater – Non-Financial awards
⋮	⋮	⋮

The list would contain all entries in the Link Database in which *Théâtre* is present in the RAMEAU expression.

Note that the above list is generated from the RAMEAU point of view. Even if other SHLs provide narrower terms that could be interlinked at a finer level, RAMEAU cannot cope with them, and the list does not show these links. When another SHL is chosen as the view point, and this SHL does contain narrower terms, two or more separate links would show up.

4.2 Basic Features

The actor interface can be described by the following features.

1. Actor-specific login. No group logins allowed, except possibly for read-only purposes. We need exact authentication and authorization per person.
2. Selection of SHL for point of view. Default is the SHL 'owned' by the person logging in. The SHL used for the point of view is called the *focus* SHL.
3. Standard 'application view' with typical search/modify/remove buttons in a concise layout.
4. List presentation of found SHL entries in the focus SHL, with the expressions in the linked SHLs next to them, organized per link.
5. Limited browsing options, essentially the list for '$*$' or other filters, ordered alphabetically per SHL.

6. Clickable individual expression arguments in all SHLs on display, which take the actor directly to the appropriate view of those arguments (focus SHL and search expression implied by the single mouse click).
7. Detail view links that show the administrative details of the various expressions and subject headings.
8. Easy access to modification pages for all relevant items, with individual security features depending on the actor's authorization.
9. Easy access to relevant annotation and negotiation/discussion modules.

Limitations of the HTML forms interface put constraints on exact layout and functional behaviour (look and feel) of the application. For example, it is difficult to create self-adapting forms or look-ahead lists with HTML. We try to stick to plain, standard HTML as much as possible, and not use any (semi-)proprietary extensions such as Java/EcmaScript or ActiveX. This will ensure the widest browser support available.

Chapter 5

The Link Database

The Link Database is the technical core of the LMI system. It contains the data required for linking up the several participating subject heading lists, plus some administrative data that is required to manage the system (both technically and organisationally).

Although the Link Database is implemented as a relational database system, the interface to the outside world is not in SQL but in a Z39.50-inspired protocol called zThes. Most of the user interfacing is through HTML forms, using a common Web browser.

5.1 Data Schema

The basic schema of the database, in simplified notation and with some support tables left out, shows six tables that are interrelated. The base table is the **Links** table, which contains the MACS system identifiers of the links.

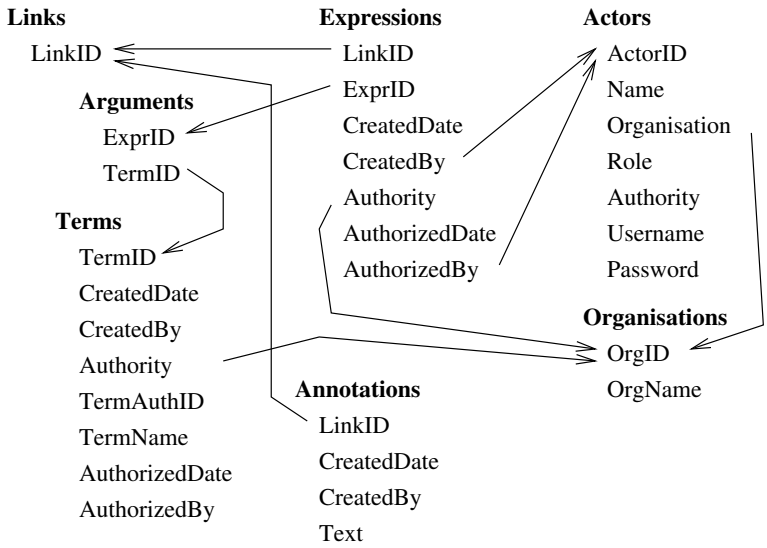

Figure 5.1: Simplified Schema of the Database

Related to links are **Expressions**, each with their own system identifier as well. A link typically has several expressions, one for each SHL. Every expression has several data values, such as the expression creation date and the ID of the person who created the expression. The Authority field contains the identifier of the authority in whose SHL the expression is written. AuthorizedDate and -By fields indicate the date of the authorization (locking) of the expression by the authority, and the identifier of the authorizing person. Both fields are empty if an expression is not authorized (locked) yet.

Each expression has one or more **Arguments**. The Arguments table only maps from one expression to several terms, it does not add more information. With the help of this table, the **Terms** table can remain completely independent of the Expressions table, we enable easy reuse of existing terms, and we facilitate maintenance.

The actual subject headings are stored in the **Terms** table. This table is by and large a partial copy of the SHL databases. It should not contain any subject heading (term) that is not available in the SHLs. The Terms table also contains typical Link Database information, such as the creation date of the copy of the term (this could also be the date of the last change), the ID of the person originally creating this copy, the owning authority of the term, etc.

All people IDs are referring to the **Actor** table, where all people authorized to use and/or modify the system are represented. Actors can play roles, e.g., *Administrator, Editor*, or *Reader*. People can leave the operational system, while their name should still be recorded. Therefore the Organisation and Role fields might be empty. Basic items such as username, encrypted password, and 'owning' authority are included as well. A decision was made to allow only one authority per actor.

The **Annotations** table simply holds plaintext annotations linked with a Link record. Annotations are part of the discussion mechanism.

Lastly, the **Organisations** table holds all participating organisations. Typically, an organisation is connected to a SHL authority, but this is not a hard link.

5.1.1 Design Decisions

A few design decisions have been made in the schema that are worth noting.

The decision to give authority IDs to expressions as well as to terms. This is done to rule out the possibility that the same expression contains terms from several authorities (which would be nonsensical). The application should only allow terms of the same authority as an expression to act as arguments.

Likewise, we duplicated the Organisation ID in the Actors table because it is possible that an organisation gives certain update rights to people working for another organisation. Also, when somebody moves from one organisation to another but is still allowed to maintain the system, it is handy to have his/her organisational affiliation change. When a person moves and is not longer allowed to access the system, his/her login rights should be taken away, but the person's data should remain in the Actors table, without affiliation to any organisation.

5.1.2 Full Data Schema

The simplified data schema suffices to gain a working understanding of the underlying data model, but does not convey all details. For the sake of completeness, we include

the full data schema here as well. Note that this is not material that should be exposed to LMI actors.

```
/*
  Table structure for the MACS2 Link Database.
  Data types for PostgreSQL.
  $Id: macs-2.sql,v 1.9 2005/09/08 09:15:48 hoppie Exp $
*/

create table organisations (
  orgID   int4      not null,
  orgName varchar(64) not null,
  primary key (orgID)
);
grant select on organisations to "www-data";

create table authorities (
  authID   int4  not null,
  authName varchar(64) not null,
  primary key (authID)
);
grant select on authorities to "www-data";

create table actors (
  actorID      int4  not null,
  name         varchar(64)  not null,
  organisation int4,
  role         varchar(16),
  password     varchar(64),
  username     varchar(16),
  authID       int4,
  lastlogin    timestamp,
  primary key (actorID),
  foreign key (organisation) references organisations (orgID)
);
create sequence actors_actorID_seq;
grant all on actors to "www-data";
grant all on actors_actorID_seq to "www-data";

create table translit (
  fromchar       char(1)   not null,
  tochar         char(2)   not null,
  primary key (fromchar)
);
grant all on translit to "www-data";
create index translit_tochar on translit(upper(tochar));

create table terms (
  termID         int4  not null,
  createdDate    timestamp    not null,
  createdBy      int4         not null,
  authority      int4         not null,
  termAuthID     varchar(32)  not null,
  termName       varchar(128) not null,
  termSorted     varchar(128),
  primary key (termID),
  foreign key (authority) references authorities (authID)
);
create sequence terms_termID_seq;
grant all on terms to "www-data";
grant all on terms_termID_seq to "www-data";
create unique index terms_authority_termSorted_idx on terms
  (authority,termSorted);
```

```
create index terms_termSorted_idx on terms(termSorted);
create index terms_authority_idx on terms(authority);

create table domains(
  domainID int4 not null,
  label varchar(64) not null,
  primary key (domainID)
);
create sequence domains_domainID_seq;
grant select on domains to "www-data";

create table links(
  linkID int4 not null,
  status int4,
  actorID int4,
  modDate timestamp,
  foreign key (actorID) references actors (actorID),
  primary key (linkID)
);
create sequence links_linkID_seq;
grant all on links to "www-data";
grant all on links_linkID_seq to "www-data";

create table changeLog(
  linkID int4 not null,
  actorID int4 not null,
  changeDate timestamp,
  logtype int4,
  text varchar(128),
  foreign key (linkID) references links(linkID) on delete cascade
);
grant all on changelog to "www-data";

create table link_domains(
  domainID int4 not null,
  linkID int4 not null,
  primary key (domainID, linkID),
  foreign key (domainID) references domains (domainID) on delete cascade,
  foreign key (linkID) references links (linkID) on delete cascade
);
grant all on link_domains to "www-data";
create index link_domains_domainID_idx on link_domains(domainID);
create index link_domains_linkID_idx on link_domains(linkID);

create table expressions (
  exprID         int4      not null,
  linkID         int4       not null,
  createdDate    timestamp  not null,
  createdBy      int4       not null,
  authority      int4       not null,
  authorizedDate timestamp,
  authorizedBy   int4,
  primary key (exprID),
  foreign key (createdBy) references actors (actorID),
  foreign key (authorizedBy) references actors (actorID),
  foreign key (authority) references authorities (authID),
  foreign key (linkID) references links (linkID) on delete cascade
);
create sequence expressions_exprID_seq;
grant all on expressions to "www-data";
grant all on expressions_exprID_seq to "www-data";
create index expressions_linkID_idx on expressions(linkID);
```

```
create index expressions_authority_idx on expressions(authority);

create table arguments (
  exprID int4 not null,
  termID int4 not null,
  rank   int4 not null,
  primary key (exprID,termID),
  foreign key (termID) references terms (termID) on delete cascade,
  foreign key (exprID) references expressions (exprID) on delete cascade
);
grant all on arguments to "www-data";
create index arguments_exprID_idx on arguments(exprID);
create index arguments_termID_idx on arguments(termID);

create table annotations (
  annID          int4       not null,
  linkID         int4       not null,
  createdDate    timestamp  not null,
  createdBy      int4       not null,
  text           varchar(2048) not null,
  authID         int4,
  todoList       varchar(16),              /* '','admin','editors' */
  primary key (annID),
  foreign key (linkID) references links (linkID) on delete cascade,
  foreign key (authID) references authorities (authID)
);
create sequence annotations_annID_seq;
grant all on annotations to "www-data";
grant all on annotations_annID_seq to "www-data";

create table actor_domains (
  actorID      int4 not null,
  domainID     int4 not null,
  primary key (actorID,domainID),
  foreign key (actorID) references actors (actorID) on delete cascade,
  foreign key (domainID) references domains (domainID) on delete cascade
);
grant all on actor_domains to "www-data";
```

5.2 zThes Record Structure

Whereas the tables contain references to other tables in order to store repeating elements, the Z-like record structure contains the repeating elements in a hierarchical tree format. The record structure is an extension of the zThes record structure and will be used by the Z39.50 database that contains the link information. A Z39.50 client can search this database for a subject heading from one of the SHLs, and the expressions in the other SHLs that are linked to the subject heading are returned as a result of the search.

A zThes record relates a subject heading of a given SHL to expressions in the other SHLs. This means that each link in the Link Management System will be represented by one or more zThes records; each simple, non-compound expression of the link will result in a zThes record. For notation purposes, we will use a subset of XML to represent the tree format. The system itself is not based on XML, we only use XML as a vehicle to get data across.

The following example in XML representation shows a link between the LCSH heading "Jumping", the RAMEAU expression "Saut en hauteur OR Sauts (atlétisme)", and

the SWD expression “Hochsprung OR Sprung”.

```
<zThes>
 <authority>LCSH</authority>
 <termId>sh85070999</termId>
 <termName>Jumping</termName>
 <link>
  <authority>RAMEAU</authority>
  <exp>
   <operator>or</operator>
   <term>
    <termId>frBN0039853452</termId>
    <termName>Saut en hauteur</termName>
   </term>
   <term>
    <termId>frBN012985577</termId>
    <termName>Sauts (atletisme)</termName>
   </term>
  </exp>
 </link>
 <link>
  <authority>SWD</authority>
 <exp>
  <operator>or</operator>
  <term>
   <termId>041374908</termId>
   <termName>Hochsprung</termName>
  </term>
  <term>
   <termId>041289900</termId>
   <termName>Sprung</termName>
  </term>
  </exp>
 </link>
</zThes>
```

The zThes record is completely self-contained. There is no direct connection between the database structure and the zThes record structure. zThes presents the records at a more aggregated level.

The subject heading from the SHL that has the focus is at the top and the structuring follows the zThes profile. The expressions are included as subrecords, in the same way as the terms that are related to the term in the top of the record are included in relation subrecords. In this way a zThes record can express for a given subject heading its relations to other subject headings in the same SHL, and its links to expressions in other SHLs. In most cases the internal relations of a SHL and the external links to other SHLs will be in separate databases and thus in separate zThes records. But in principle the two kinds of information can be merged into one zThes record.

The expressions themselves are clearly separated from the rest by `<exp>` tags. In order to be able to generate Z39.50 type-1 queries, the element must be structured. The query/expression ‘B1 AND (B2 OR B3)’ is represented as:

```
<exp>
 <operator>and</operator>
 <term>B1</term>
 <exp>
  <operator>or</operator>
  <term>B2</term>
  <term>B3</term>
 </exp>
</exp>
```

The element <exp> is recursive. A DTD can define this recursiveness. The element <operator> has two possible values: 'or', and 'and'. In the prototype, 'or' is never stored explicitly in the database, but since the zThes file is generated relative to a focus SHL, the generator can convert the implicit 'or's into explicit ones.

The simple, non-compound expression that contains only the term S will be represented as

```
<exp>
 <term>S</term>
</exp>
```

- o - o - o - o -