Deliverable n°: D6

LeActiveMath Structure and Metadata Model

The LeActiveMath consortium

December 2004

Version

Main Authors:
George Goguadze (University of Saarland)

Other Contributors:
Carsten Ullrich, Erica Melis, Jörg Siekmann (DFKI), Christian Gross (UA), Rafael Morales (UNN)

Project funded by the European Community under the Sixth Framework Programme for Research and Technological Development

Project Ref.No.	IST-507826
Project Title	LeActiveMath- Language-Enhanced, User Adaptive, Interactive eLearning for Mathematics

Deliverable Status	Final
Contractual Date of Delivery	Project Month N°12 - Date 31.12.2004
Actual Date of Delivery	Project Month N°12 - Date 31.12.2004
Deliverable Title	Structure and Metadata Model
Type	Report
Version	3.0
Number of Pages	32
WP Contributing to the Deliverable	2
WP/Task Responsible	University of Saarland
Author(s)	George Goguadze
EC Project Officer	Colin Stuart
Keywords	Semantic Knowledge Representation, Educational Metadata, Mathematical Metadata

Contents

1 Executive Summary **5**

2 Ontologies of Mathematical Knowledge **6**

3 Macro- and Micro- Structure of Mathematical Knowledge Items **6**

3.1 OMDoc Knowledge Items . . . 7
3.2 LeActiveMath extensions of OMDoc micro-structure . . . 8
3.3 OMDoc macro- structure and content packages . . . 8

4 Metadata of Mathematical Resources **9**

4.1 General Administrative Metadata . . . 9
4.1.1 Creator . . . 10
4.1.2 Contributor . . . 10
4.1.3 Coverage . . . 11
4.1.4 Date . . . 11
4.1.5 Description . . . 11
4.1.6 Format . . . 12
4.1.7 Identifier . . . 12
4.1.8 Keyword . . . 12
4.1.9 Language . . . 12
4.1.10 Publisher . . . 13
4.1.11 Rights . . . 13
4.1.12 Relation . . . 14
4.1.13 Source . . . 14
4.1.14 Subject . . . 14
4.1.15 Title . . . 15
4.1.16 Type . . . 15
4.2 Domain-Specific Metadata . . . 15
4.2.1 Relation . . . 15
4.2.2 Theories and Imports . . . 17
4.3 Educational Metadata . . . 17
4.3.1 Learning Context . . . 18
4.3.2 Field . . . 19
4.3.3 Difficulty . . . 19
4.3.4 Semantic Density . . . 19

4.3.5 Competency . . . 20
4.3.6 Competency Level . . . 22
4.3.7 Typical Learning Time . . . 24
4.3.8 Exercise Type . . . 24
4.3.9 Exercise Purpose . . . 25
4.3.10 Interactivity Type . . . 25
4.3.11 Interactivity Level . . . 25
4.3.12 Representation . . . 25
4.3.13 Abstractness . . . 26
4.3.14 Relation . . . 26
4.4 Metadata and Diagnosis for Exercise Steps . . . 27
4.4.1 Metadata Elements in Interactions . . . 28
4.4.2 Diagnosis Elements in Interactions . . . 29
4.5 XML-Binding . . . 30

1 Executive Summary

The goal of this report is to define knowledge representation for maths education content of LeActiveMath. This representation should allow for search, retrieval, reuse of mathematical knowledge, adaptive course generation and presentation within an interactive learning environment. This knowledge representation should allow for interactive exercises and other items of explorative character, such as interactive concept maps.

Such representation consists of typed items of mathematical knowledge, annotated with metadata information. These metadata annotations define mathematical and educational properties of knowledge items and relate knowledge items to each other, establishing ontologies of mathematical and educational knowledge.

Metadata provides a basis for reusability and automatic course generation and presentation adapted to different individual parameters of the learner. We define a metadata element set and the usage of the defined metadata elements within LeActiveMath.

LeActiveMath knowledge representation is based on and extending `OMDoc`- standard for mathematical knowledge representation, metadata standards such as Dublin Corefor general administrative metadata, creative commons metadata for copyright information, IEEE learning object metadata (LOM) for educational metadata.

Several subject classification systems such as library of congress subject headings (LCSH), Deway Decimal System (DDC) and Mathematical Subject Classification (MSC) are used for classifying content. For specifying the values of metadata elements, several standard vocabularies, such as Dublin Corevocabularies, Vocabulary of Machine Readable Cataloging Record (MARC), and also several standard syntax specifications of W3C and iso are used.

Some extensions for educational metadata are made, such as `competency` and `competencylevel` influenced by Program for International Student Assessment (PISA) and National Council of Teachers of Mathematics (NCTM).

Although LeActiveMath metadata elements are mostly adopted from standards, there is a difference in LeActiveMath definition of the meaning of the imported metadata elements. Some metadata elements can have values only relative to the value of other metadata elements, for example `difficulty` of an exercise can be dependent on the `learningcontext`. Namely, it can happen, that the exercise is hard for the students in the context of the "University First Cycle", but it is easy for students in "University Second Cycle".

In order to reach reusability, we allow to annotate the metadata elements dependent on learning context with a *learningcontext* attribute to make it clear for to which learning context this metadata element is applicable.

Internal structure of the knowledge items is encoding semantics of their content rather then presentation. In this way the presentation of the same content in different presentation formats is supported.

We describe the knowledge items used in LeActiveMath and their internal structure. There are two types of items in LeActiveMath- concepts and satellites. Concepts are the main items of mathematical ontology and the satellites are additional items that are grouped around the concepts.

The internal structure of the most of the items is simple - they consist of text in different languages mixed with formulas in OpenMath format. Interactive items such as exercises have more complex internal structure which is needed for expressing their interactivity.

2 Ontologies of Mathematical Knowledge

Mathematical language is well structured and implicitly carries an amount of semantics. However, it is not structured enough for being used by computers in an educational application requiring semantics of mathematical documents.

A goal of the LeActiveMath project is to design a refined representation format for mathematical knowledge, annotated with additional educational information in order to provide a representation for reusable mathematical knowledge to be used in a learning environment.

According to the requirement analysis of LeActiveMath, such a format should allow for reusability of mathematical knowledge and adaptive course generation and presentation. Among the dimensions for such an adaptivity are learning context, competences to be achieved, field of interest of the learner, difficulty and abstractness of the learning material, different presentation styles. The precise definition of these dimensions and relations between them is described in this report.

In order to enable such a reusability, the envisioned markup has to be machine-readable and independent of the system component using it.

The knowledge representation in LeActiveMath is based on `OMDoc` - a semantic markup language for mathematical documents. `OMDoc` has evolved as an extension of the OpenMath standard for mathematical formulas (see [14]). In `OMDoc` mathematical knowledge is represented by a collection of typed items of mathematical knowledge and mathematical relations among them. This creates a domain ontology of mathematical knowledge (see [7]). This ontology comprises different kinds of mathematical items such as theorems, proofs, and examples. On one hand, these items may have a micro-structure themselves. We discuss such a micro-structure in the sections 3.1 and 3.2. On the other hand, knowledge items can form collections of documents for which the order and occurrence of particular items is defined. This macro-structure and organization of knowledge items are considered in the section **??**.

In the remainder of this report we distinguish between the following terms:

Resource a knowledge item or a collection of items, to which a metadata record can be assigned.

Item an item is every `OMDoc` item that corresponds to a single instructional object (e.g., `definition`, `theorem`, `example`, `exercise`, `introduction`, ...)

Concept a concept is a principal building block of the domain ontology. in `OMDoc` these are `definition, assertion` (including its subtypes such as `theorem`), `method, proof, problem`.

Satellite a satellite element is an auxiliary element is related to one or more concepts. in `OMDoc` these are `example, exercises, omtext` (and its subtypes such as `introduction, elaboration, motivation`). We change the name of the `OMDoc` item `omtext` to `omitem`, because for LeActiveMath the name `omtext` it is misleading - this item does not necessary have textual content.

Due to the different syntaxes used by different standards we partially adopt, for LeActiveMath we introduce a normalized syntax for all non-numeric values of attributes. The values are represented in English using non-capital letters. In case the value consists of several words, the words are separated by a '_' sign.

3 Macro- and Micro- Structure of Mathematical Knowledge Items

In this section we consider the atomic items of mathematical knowledge used in LeActiveMath.

Under micro-structure of knowledge items we understand the internal structure of these items, designed for adaptive (sometimes interactive) presentation.

Names and micro-structure of most of the elements in LeActiveMath is adopted from `OMDoc` standard. we give a brief overview of elements, available in `OMDoc` and define some extensions.

3.1 `OMDoc` Knowledge Items

Atomic elements of `OMDoc` ontology are divided into **concepts** and **satellites**. The following concepts in `OMDoc` markup used in LeActiveMath:

- `symbol` represents an atomic mathematical concept around which the ontology is built and is defining a syntax element of a formal theory
- `axiom` is a postulate about the symbols of a current theory
- `definition` is a statement, defining a meaning of symbols of a current theory
- `assertion` is a statement about the symbols of a current theory. There can be several types of assertions such as 'theorem', 'lemma', 'corollary', 'conjecture' etc.
- `proof` represents a proof of an assertion and is always connected to the assertion it proves with the relation of y type 'for'

The satellite elements of `OMDoc`, commonly used in LeActiveMath are :

- `example` is an example illustrating a concept. It is always assigned to this concept using relation of a type 'for'
- `exercise` is another important satellite that can train various competencies of the learner w.r.t. the concept it is assigned to.
- `omtext` represents various types of textual notes, such as 'elaboration', 'introduction', 'conclusion', 'motivation' and others.

Most of the knowledge items in `OMDoc` have a simple internal structure. They consist of metadata and two types of building blocks - CMP (commented mathematical property) consisting of text with formulas in OpenMath format, and/or FMP (formal mathematical property) which we do not currently use. FMP consists only of formulas in OpenMath format.

Using the OpenMath language for representing mathematical formulas is suitable for goals of LeActiveMath. It provides a semantic representation for mathematical formulas that can be unambiguously communicated with mathematical systems connected to LeActiveMath. For the translation into the syntax of mathematical systems used, the so-called phrasebooks are used.

The CMP component might have a language attribute. Thus, each item can consist of CMPs in different languages.

Several mathematical items (proofs, exercises, examples) possess a complex internal structure. For instance, the `proof` element in `OMDoc` can have a complex internal structure. It can be represented as a directed acyclic graph (DAG) of derivation steps, connected with cross-links. Such a representation allows for different possibilities for presenting a proof to the learner. It can be a complete proof at once, but also a step by step replay of the proof process, or an unraveling proof-plan.

3.2 LeActiveMath extensions of OMDoc micro-structure

In order to enable problem-oriented scenarios, we introduce a new element, called `problem`. This element will represent some mathematical problem statement. When selecting `problem` as a learning goal, the course generator may collect all the prerequisites needed in order to understand this problem and all satellites that illustrate the problem or train to understand it.

LeActiveMath extends the `OMDoc` item `example`. A detailed explanation can be found in [11]. In case of a worked-out example, the micro-structure of this element is enriched with a `solution` that has a structure similar to a `proof`. The `solution` element can consist of a DAG of derivation steps connected with cross-links. It differs from the `proof` element. The solution might not only prove some statement, but also calculate the value of some expression or explore the properties of a particular structure (e.g. curve discussion). This representation allows for different presentations, and serves as a basis for the automatic generation of exercises by fading some parts of the structure of a worked-out example.

The interanl structure of an exercise is refined too. Interactive exercises play an important role in LeActiveMath. The micro-structure of an interactive exercise has to allow for different kinds of interactivity, checking the correctness of the answer, providing feedback, etc.

The processing of the exercise in LeActiveMath is represented as a graph of interactions between the exercise subsystem and a learner. It represents feedback and several types of hints. This interaction graph can be automatically filled with information by the exercise subsystem components that can communicate with external systems in order to generate feedback to the user.

A description of LeActiveMath language for exercises can be found in [4] and [5].

3.3 OMDoc macro- structure and content packages

`OMDoc` possesses the mechanism of structuring mathematical knowledge within theories and establishing semantic connections between items of mathematical knowledge and theories containing these items.

The structure element `theory` of `OMDoc` models the concept of a formal theory. By importing mathematical theories and mapping their semantic constructions one can build the hierarchies of mathematical theories (see [6]).

Another way of structuring the content is defining so-called content packages. A content package is the way of organization of content items, describing the physical resources of the knowledge items and relations between items and the views of content using so-called *organizations*.

A LeActiveMath content package is called a *collection*. It consists of the collection of content items, metadata annotations for these items, presentation information and organizations of two types - so-called pre-recorded books and tables of content.

LeActiveMath is not using IMS content packages format of IMS global learning consortium[1], since the IMS content packages does not encode sufficient information for representing a LeActiveMath content package. For example, IMS content packages can not represent a difference between the pre-recorder book and the table of contents, it can not represent the difference between the content items and the metadata items or presentation information which are stored separately in one package.

[1] see http://www.imsglobal.org/content/packaging/

4 Metadata of Mathematical Resources

Various properties of knowledge items and the way of their organization can described using metadata annotations, "data about data".

Metadata is the set of annotations that serve to facilitate the use and reuse of the database of mathematical knowledge, search, retrieval and adaptive presentation of the knowledge within an interactive learning environment.

Metadata can be dependent on the context, and therefore, should be stored separately from the content of the resources they is assigned to.

The metadata can be assigned to individual items as well as to their collections, that is, metadata is assigned to a *resource*. An *inheritance* mechanism for metadata can be defined for the collections of items. At the moment, we use a hierarchical inheritance, i.e. The elements of a content package inherit the metadata of a whole package as soon as no other metadata is assigned to this element.

Metadata is assigning some properties to the items and the way of the organization of these items depending on the goals of the application using the mathematical knowledge. This means, it does not only reflect the mathematical structure of the content, but annotates them with additional information, needed in order to facilitate the management of the content. The pure mathematical structure of the connections between items can be expressed by the constructions of `OMDoc` via the theory-import mechanism.

In LeActiveMath we extend/modify the general and mathematical metadata of `OMDoc` in order to adapt it to the goals of the project and we define some additional educational metadata.

The metadata in LeActiveMath is classified in the following categories: general administrative metadata, mathematical and educational metadata. We define these metadata in the subsequent sections.

A metadata record in LeActiveMath can consist of one or more instance of each type of metadata element.

Note that when annotating a content resource, not necessarily all metadata elements are to be provided. Some basic elements, however, such as `title`, `creator`, `date` have to be present.

In the following section we define a metadata model for LeActiveMath. We use the following syntax: each metadata element description consists of a **Name** of an element, its' **Purpose**, **Relation to existing metadata standards**, LeActiveMath **values**, elements it **Applies to**, **Number of occurrences** of the element per metadata record or per resource, and some additional **Note**.

4.1 General Administrative Metadata

For describing the basic properties of a resource, such as its creator and date of creation or modification, etc., we use Dublin Coremetadata [2].

Dublin Coremetadata initiative[2] suggests a minimal metadata element set that contains basic metadata needed by most Web applications. It contains the administrative information about the document. According to the version 1.1 of Dublin Coremetadata element set, there are 15 metadata elements: **creator, contributor, coverage, date, description, format, identifier, language, publisher, rights, relation, source, subject, title, type**.

Below we describe in more detail administrative metadata used in LeActiveMath, which include Dublin Coreelements, some refinements and additions.

[2]see `http://www.dublincore.org`

4.1.1 Creator

Purpose refers to a person, organization, or a service, primarily responsible for creating the content of a resource.

Name `Creator`

Relation to existing metadata standards defined in Dublin Core, using role refinements from open eBook with values from MARC standard

LeActiveMath **values** text.

Applies to all resources.

Number of occurrences one or more per resource.

Note `Creator` has an identifier (ID) which is unique per resource. The `Date` metadata might refer to it (see 4.1.4).

`Creator` also has a *role* attribute with values, used in `OMDoc` and defined in open eBook publication structure specification ([?]). The values for the *role* property are a subset of a large vocabulary of so-called relator codes of MARC (machine-readable cataloging record).[3] `role` property can have the values:

aut	author - person mainly responsible for the content of the resource
ant	scientific antecedent - author of the work the resource is based upon
clb	collaborator making a limited contribution to the resource
edt	editor who prepares a document for publication
ths	thesis advisor under whose supervision the resource is created
trc	transcriber - making a copy of the original material in another format
trl	translator of the resource content from one language to another

4.1.2 Contributor

Purpose refers to a person, organization, or a service, responsible for making contributions to the content of the resource.

Name `Contributor`

Relation to existing metadata standards defined in Dublin Core, using role refinements from open eBook with values from MARC standard.

LeActiveMath **values** text.

Applies to all resources.

Number of occurrences one or more per resource.

Note just as `Creator`, `Contributor` element has an identifier, unique within the resource and a *role* attribute with the same values as for `Creator`.

[3] see `http://www.loc.gov/marc/marc.html`

4.1.3 Coverage

Purpose refers to a geographical location the content of the resource is suited for.

Name `coverage`

Relation to existing metadata standards defined in Dublin Core

Applies to all resources.

LeActiveMath **values** text.

Number of occurrences one or more per resource.

4.1.4 Date

Purpose refers to the date of creation or modification of the content of the resource.

Name `Date`

Relation to existing metadata standards defined in Dublin Core, extended by `OMDoc`

LeActiveMath **values** we use the date-time format for values of `date`, as defined in w3c specifications[4]. this format represents the [iso 8601] extended format ccyy-mm-ddthh:mm:ss where "cc" represents the century, "YY" the year, "mm" the month and "DD" the day.

Applies to all resources.

Number of occurrences one or more per resource.

Note the `Date` element has two attributes – *action* and *who* – specifying which actions are performed and who made this changes. The value of who is a reference to an identifier of a `Creator` or `Contributor` element, and *action* contains a keyword for the undertaken action with recommended values 'updated', 'new', 'imported', 'frozen', 'normed'. The *action* and *who* attributes are adopted from `OMDoc` standard.

4.1.5 Description

Purpose refers to an account of the content of the resource.

Name `Description`

Relation to existing metadata standards defined in Dublin Core

LeActiveMath **values** text.

Applies to all resources.

Number of occurrences one or more per resource.

[4] see http://www.w3.org/tr/xmlschema-2/#datetime

4.1.6 Format

Purpose refers to the physical or digital manifestation of the resource. As defined in Dublin Core, it may be used to identify the software, hardware, or other equipment needed to display or operate the resource.

Name `Format`

Relation to existing metadata standards defined in Dublin Core

LeActiveMath **values** MIME types of the media, contained in the resource.

Applies to all resources.

Number of occurrences one or more per resource.

4.1.7 Identifier

Purpose refers to an unambiguous reference to the resource within a given context.

Name `identifier`

Relation to existing metadata standards defined in Dublin Core

Applies to all resources.

Number of occurrences there can be only one `identifier` per resource.

Notes the context for an atomic content item in LeActiveMath is determined by the identifier of an `theory` this item belongs to. The context of a theory itself is determined by the content package (so called 'collection').

4.1.8 Keyword

Purpose defines a keyword for the resource content facilitating keyword search

Name Keyword

Relation to existing metadata standards taken from LOM

LeActiveMath **values** values from LCSH, DDC, MSC classification systems as in 4.1.14

Applies to all resources.

Number of occurrence zero or more per resource.

4.1.9 Language

Purpose refers to the language of the intellectual content of the resource.

Name `Language`

Relation to existing metadata standards defined in Dublin Core

Applies to all resources.

Number of occurrences one or more per resource.

Notes since content in LeActiveMath is internationalized and each content item consists of text in different languages, the usage of this element on the top level of the content item is questionable. One or more instances of the `Language` element can be present in order to specify, in which languages this content item is available. However, this information can be automatically extracted from the content.

4.1.10 Publisher

Purpose refers to an entity responsible for making the resource available.

Name `Publisher`

Relation to existing metadata standards defined in Dublin Core

Applies to all resources.

Number of occurrences one or more per resource.

4.1.11 Rights

Purpose refers to information about rights held in and over the resource.

Name `Rights`

Relation to existing metadata standards defined in Dublin Core, refined by LOMand creative commons metadata standard.

LeActiveMath **values** for this element we introduce a refinement different from the ones in LOM. We adopt the metadata for rights from the creative commons metadata standard. It divides the license characteristics in three types: `permissions`, `prohibitations` and `requirements`.

Permissions are the rights granted by the license and might have the following values:

- 'reproduction' - the work may be reproduced
- 'distribution' - the work may be distributed, publicly displayed, and publicly performed
- 'derivative_works' - derivative works may be created and reproduced

prohibitions are the things the license prohibits. It can have a value 'commercial_use' stating that rights may be exercised for commercial purposes.

Requirements are restrictions imposed by the license with following values:

- 'notice' - copyright and license notices must be kept intact
- 'attribution' - credit must be given to copyright holder and/or author
- 'copyleft' - derivative works, if authorized, must be licensed under the same terms as the work

Applies to all resources.

Number of occurrences one per resource.

Notes the values of types and their definitions are cited from the creative commons metadata draft (v 1.0b2)[5].

[5] see `http://creativecommons.org/metadata/spec-1.0b2`

4.1.12 Relation

Purpose defines a reference to a related resource.

Name `Relation`

Relation to existing metadata standards defined in Dublin Coreand refined by LOM

LeActiveMath **values** one or more references to identifiers of the related resources. Relations can have types with following values from Dublin Coreand LOMextended with some LeActiveMath values and adapted to LeActiveMath syntax: 'references', 'is_referenced_by, 'prerequisite', 'is_prerequisite_for', 'requires', 'is_required_by', 'for', 'counter', 'is_special_case_of', 'has_special_case'.

Applies to all resources.

Number of occurrences one or more per resource.

Note we refine the Dublin Corerelation element with a type attribute, having values from LOMand some additional values. Since for LeActiveMath the `Relation` element falls into a categories of domain-specific and educational metadata, we will discuss the types of relations in a subsections 4.2.1 and **??** in more detail.

4.1.13 Source

Purpose refers to a unique reference to a resource from which the present resource is derived.

Name `Source`

Relation to existing metadata standards defined in Dublin Core

LeActiveMath **values** the reference will be provided in the form of an identifier, unique in the given context.

Applies to all resources.

Number of occurrences one or more per resource.

4.1.14 Subject

Purpose refers to a topic of the content of the resource.

Name `Subject`

Relation to existing metadata standards defined in Dublin Core

LeActiveMath **values** there exists several well-established Mathematical Classification Systems. `Subject` can contain a keyword from a controlled vocabulary. The vocabulary depends on the *schema* property of the `Subject` element. We use the following mathematical classification systems:

- LCSH - Library of Congress Subject Headings[6]
- MSC - 2000 Mathematics Subject Classification[7]
- DDC - Deway Decimal Classification[8]

[6] For the outline see http://www.loc.gov/catdir/cpso/lcco/lcco.html
[7] For the reference, see http://www.ams.org/msc
[8] For information on DDC, see http://www.oclc.org/dewey

Applies to all resources.

Number of occurrences there can be one or more `Subject` elements per metadata record, pointing to different subjects or to the same subject in the different classification scheme.

Note the choice of the classification systems is due to the wide usage of this standards and common practice, e.g. in the EULER project that approves these classification systems to be well-suitable for the management of digital libraries of mathematical publications.

4.1.15 Title

Purpose refers to a name given to the resource.

Name `Title`

Relation to existing metadata standards defined in Dublin Coreand refined in LOMand `OMDoc`

LeActiveMath **values** `Title` can have a language attribute so that for each language one can provide a separate `Title` element.

Applies to all resources.

Number of occurrences multiple occurrence of `Title` is accepted, but not more then one per language.

4.1.16 Type

Purpose refers to the nature or genre of the content of the resource.

Name `Type`

Relation to existing metadata standards defined in Dublin Core

LeActiveMath **values** we adopt the values from the DCMI Type Vocabulary Recommendation and adapt them to LeActiveMath syntax. These are 'collection', 'dataset', 'event', 'image', 'interactive_resource', 'moving_image', 'physical_object', 'service', 'software', 'sound', 'still_image', 'text'.

Applies to all resources.

Number of occurrences one or more per resource.

4.2 Domain-Specific Metadata

Domain-Specific annotations for mathematical items in `OMDoc` are partially present in the markup for these items itself. For example, the types of mathematical items, like "definition" or "proof" are present in the names of the corresponding markup elements, and other properties, such as *type* of a definition with values like "simple" or recursive.

4.2.1 Relation

Here we describe mathematical relations in LeActiveMath.

domain_prerequisite / is_domain_prerequisite

Purpose The described resource requires the referenced resource to support its function, delivery, or coherence of content [9] (reverse for the reverse relation). We refine the LOM relation type 'requires' into two sub-types. In the case of domain-specific prerequisites we are interested in concepts that are mathematically necessary in order to understand the current resource. The other sub-type is an educational prerequisite and is considered in the subsection **??**

Relation to existing metadata standards refinement of Dublin Core/LOMrelation type 'requires'.

LeActiveMath **values** the ID or the list of IDs of concepts the current resource relates to.

Applies to all resources.

Note the inverse relation is automatically computed by the system.

for

Purpose describes the relation to concepts that the itemis for. The difference from 'requires' is that the current resource does not only require the concept it relates to, but also serves as a satelite to this concept. Therefore, this type of relation is present mostly in satelites. Moreover, each satelite must have at least one relation of a type 'for' in order to specify which concept it is a satelite for. The only two concepts that must also have a relation of a type 'for' are `proof` itemreferring to the assertion it proves and `definition` itemreferring to the `symbol` it defines.

Relation to existing metadata standards not existent in other metadata standards as they do not distinguish between concepts and satellites. The relation type `for` is adopted from `OMDoc`.

LeActiveMath **values** the ID or the list of IDs of concepts the current resource relates to.

Applies to satellites, `proof`, `definition`

Number of occurrences one or more per resource.

counter

Purpose describes the relation to concepts for which the current itemserves as a counter example.

Relation to existing metadata standards not existent in other metadata standards as they do not distinguish between concepts and satellites.

LeActiveMath **values** the ID or the list of IDs of concepts the current resource relates to.

Applies to examples

Number of occurrences one or more per resource.

4.2.2 Theories and Imports

In LeActiveMath there is also a need to relate resources, such as collections of items in order to establish a mathematical ontology and connections between areas of mathematics.

OMDoc element theory serves the purpose of assemling knowledge items in mathematical theories.

Smaller theories can be assembled to bigger ones via the imports mechanism of OMDoc. The smallest theories can consist of a single symbol and the biggest could define mathematics, importing all the other theories in one. The authors, however, must be careful when defining very big theories in order to assemble theories in a mathematically consistent way.

There is a lot of room for individual decisions of authors which collection of items is sensible to give a separate name and, therefore, separate it in a theory. There are a lot of well established mathematical theories, but the borders of these theories are yet not well-defined.

Some theories are relateviely small and specifically concerned with particular concepts, so they are called theories, e.g. the theory of complex numbers. Others, such as mathematical analysis are too large and covering too many different concepts and, therefore, are called areas of mathematics, or subjects.

Similarly to this unprecise handling of theories, there exist various mathematical classification systems, trying to provide a classification of mathematical subjects. However, these classification systems are designed for the purposes of search and retrieval of mathematical information from the database of knowledge and do not serve the purpose of semantic classification of mathematical concepts.

Concluding, LeActiveMath is using OMDoc theories to build ontologies of mathematical knowledge, connecting theories with each other and building new theories from existing ones using the OMDoc import mechanism.

The theory import mechanism works in the following way: each time an author defines new concepts, he introduces a new theory containing these concepts. In order to specify mathematical prerequisites for this theory he can, on one hand, explicitly provide relations for each concept he introduces to existing concepts from other theories, as described in the subsection ??. Alternatively, the author can provide an import statement that establishes a relation to the whole theory he wants to import. By doing this he automatically imports all the concepts defined in the theory he refers to.

There is also a possibility to import theories only partially. For this, inside an import statement the morphisms of symbols should be provided. The author defines new symbols in his current theory and in the import statement specifies which symbols from the theory he imports are mapped to the corresponding symbols in the current theory. After doing this, only the symbols, declared in the import statement and all the assertions and other items containing these symbols are imported from the related theory. In this way one can also construct mathematical theories from existing ones. For example, one can construct a theory of fields in the following way: first the author defines a new theory with four symbols symbols "+", "·", "1" and "0". Then he imports the theory of commutative groups two times. In the first import statement the author maps the group operation "*" to "+" and the unit "e" of a group to "0" in current theory, and in the second import statement he maps the group operation "*" to "·" and the unit "e" of a group to "1" in current theory. By doing this, all the assertion valid for "*" and "e" are automatically valid in the current theory for both operators "+" and "·" and units "0" and "1" respectively.

For more information on theory imports mechanism, see [7].

4.3 Educational Metadata

LeActiveMath is adopting educational metadata from standards such as LOM, restricting or refining it in order to adapt to the goals of the project, adding new metadata.

IEEE Learning Object Metadata (LOM) together with IMS Learning Resource Metadata suggests a metadata element set needed to annotate learning objects. According to the LOMspecification,[9] the purpose of this standard is to facilitate search, evaluation, acquisition, and use of learning objects by learners or teachers. Another goal is to facilitate the sharing and exchange of learning objects.

The elements of the Base Scheme of LOMare grouped in nine categories: `General`, `Lifecycle`, `Meta-metadata`, `Technical`, `Educational`, `Rights`, `Relation`, `Annotation` and `Classification`. Each of this categories groups data elements. Every element has an *explanation* that defines this element, *size* indicating the number of values, *order* of this values, *value space*, *data type*, and an illustrative *example*.

Dublin Coreelement set is represented as a subset of LOM, and some refinements of DC elements are provided. For instance, the element `relation` is refined into separate category, including important properties of relation such as *kind* specifying the nature of relation, *keywords*, or other classification Al metadata.

Metadata describing a learning situation are defined. For example, one can specify the type of learning resource (exercise, table, self assessment etc.), intended role of the user (learner, teacher, author etc.), learning context (secondary education, university first cycle, etc.), semantic density and technical difficulty of the content.

In the following sections we describe the educational metadata of LeActiveMath where we adopt parts of metadata, refining it in some cases with LeActiveMath-specific values.

4.3.1 Learning Context

Learning context in LeActiveMath is using values, as partners have agreed that they approximately characterize corresponding learning contexts in all partner countries.

Purpose describes the educational context of the intended target audience of a resource [?].

Name `learningcontext`

Relation to existing metadata standards provided by Ariadne [1] (didactical context: a triple of country, context, and level) and IMS LOM(values: Primary Education, Secondary Education, Higher Education, University First Cycle, University Second Cycle, University Post grade, Technical School First Cycle, Technical School Second Cycle, Professional Formation, Continuous Formation, Vocational Training)

LeActiveMath **Values** taken from LOMand adapted to LeActiveMath syntax: 'primary_education', 'secondary_education', 'higher_education', 'university_first_cycle', 'university_second_cycle', 'university_post_grade', 'technical_school_first_cycle', 'technical_school_second_cycle', 'professional_formation', 'continuous_formation', 'vocational_training

Applies to all resources

Number of occurrences one or more per resource

The values mostly used in LeActiveMath will be 'Secondary Education', 'University First Cycle', 'University Second Cycle' and 'Professional Formation'.

[9] see `http://ieeeltsc.org/wg12LOM/`

4.3.2 Field

Purpose to provide content that corresponds to the personal interests or major field of study (university setting) of the learner

Name `field`

Relation to existing metadata standards does not exist in current standards

LeActiveMath **values** values from LCSH, DDC and MSC classification systems as in 4.1.14.

Applies to all resources.

Number of occurrences one or more per resource.

Note According to a previous version of this paper, the metadata `field` should only be applied to satellites and problems. Yet there seems to be agreement that also concepts themselves might be relevant only for certain users who can be specified using the `field`. E.g., there are theorems and definitions that are destined only for an extended curriculum (a German high school "Leistungskurs" or a university course for mathematicians).

4.3.3 Difficulty

Purpose how hard it is to work through the learning object for the typical target audience [9] (the target audience is given via learning context)

Name `difficulty`

Relation to existing metadata standards adopted from LOM

LeActiveMath **values** 'very_easy', 'easy', 'medium', 'difficult' or 'very_difficult'

Applies to examples / exercises.

Number of occurrences one or more per resource (possibly annotated with learning context)

4.3.4 Semantic Density

Purpose ratio of usefulness compared to size/duration

Name `semanticdensity`

Relation to existing metadata standards adopted from LOM

LeActiveMath **values** 'very_low', 'low', 'medium', 'high' or 'very_high'

Applies to examples / exercises.

Number of occurrences one or more per resouce (possibly annotated with learning context)

4.3.5 Competency

Purpose describes the mathematical competencies an exercise or example trains.

Name `competency`

Relation to existing metadata standards does not exist in current metadata standards. Adopted from PISA [8].

LeActiveMath **values** at least one from: 'think', 'argue', 'model', 'solve', 'represent', 'language', 'communicate', 'tools'.

Applies to examples / exercises.

Number of occurrences one or more per resouce (possibly annotated with learning context)

Note the complete names of the competencies are: mathematically thinking, argue mathematically, model mathematically, solve problems mathematically, use mathematical representations, deal with symbolic and formal elements of mathematics, communicate, usage of tools and aids.

The following description was adopted from [12] and [8].

The competencies arise from the current international discussion in mathematics education (with strong influence from PISA and from the NCTM). Approximatively you can consider them as learning objectives on a higher level. The advantages besides being up to date and contributing to the contemporary developments in mathematics education are the internationality and the fact that these competencies are (almost) not age- or content-specific.

Our competency-levels directly derive from the PISA-levels. They are descriptions of behavior a student on level x should show. The competency-levels specify the competencies' standard. Having some calibrated examples form TIMSS and PISA makes those competency-levels easy to handle and supports comparison and development of adequate content for each level.

We need competencies and competency-levels for assessment and evaluation.

The different competencies can be characterized in the following way:

Think mathematically includes the ability to

- pose questions that are characteristic for mathematics ("Are there . . . ?", "How does. . . change?", "Are there exceptions?")
- understand and handle the scope and limitations of a given concept
- make assumptions (e.g. extend the scope by changing conditions, generalize or specify, with reasons)
- distinguish between different kinds of mathematical statements (e.g. conditional assertions, propositional logic)

Argue mathematically includes the ability to

- develop and assess chains of arguments (explanations, reasons,proofs)
- know what a mathematical proof is and what not
- describe solutions and give reasons for their correctness or incorrectness
- uncover the basic ideas in a given line of arguments
- understand reasoning and proof as fundamental aspects of mathematics

Solve problems mathematically includes the ability to

- identify, pose and specify problems
- self-constitute problems
- monitor and reflect on the process of problem solving
- endue strategies / heuristics
- solve different kinds of problems (with various contexts outside of mathematics, open-ended exercises)

Model mathematically includes the ability to

- translate special areas and contents into mathematical terms
- work in the model
- interprete and verify the results in the situational context
- point out the difference between the situation and the model

Use mathematical representations includes the ability to

- understand and utilize (decode, interpret, distinguish between) different sorts of representation (e.g., diagrams and tables) of mathematical objects, phenomena, and situations
- find relations between different kinds of representation
- choose the appropriate representation for the special purpose

Deal with symbolic and formal elements of mathematics includes the ability to

- use parameters, terms, equations and functions to model and interpret
- translate from symbolic and formal language into natural language and the other way round
- decode and interpret symbolic and formal mathematical language and understand its relations to natural language

Communicate includes the ability to

- explain solutions
- use a special terminology,
- work in groups, including to explain at the adequate level
- understand and verify mathematical statements of others

Use tools and aids includes the ability to

- know about the existence of various tools and aids for mathematical activities, and their range and limitations;
- to reflectively use such tools and aids

Accordning to the abilities each competency contains each of the competencies can have subcompetencies, with values listed below:

- think
  - scope
  - formulate

  - generalize
- argue
  - judge
  - find_choose
- model
  - decode
  - encode
- solve
  - apply_algorithms
- represent
  - decode_understand_representations
  - decode_understand_relations
  - switch_choose
- language
  - translation
  - interpret_manipulate
  - formal_rules
- communicate (no subcompetencies)
- tools
  - calculator
  - search
  - concept_map
  - CAS
  - DGS (dynamic geometry software)
  - plotter

4.3.6 Competency Level

Purpose describes the different levels of mathematical knowledge an exercise or example requires/trains.

Name `competencylevel`

Relation to existing metadata standards does not exist in other metadata standards. Adopted from PISA [8], without the dependencies to the content.

LeActiveMath **values** one among: 'elementary', 'simple_conceptual', 'multi_step', 'complex'.

Applies to examples / exercises.

Number of occurrences one or more per resource.

Note the complete names of the competency levels are: Level I, Computation at an elementary level; Level II, Simple conceptual solutions; Level III, Challenging multi-step-solutions; Level IV, Complex processings (modelings, argumentations).

We define four competency levels (level II and III of the PISA-competency-model are combined as the new level II). They were adopted from the PISA competency level model but we removed the dependency on the original content (which was geometry at grade 8).

Level I: Computation at an elementary level To achieve this competency level, students have to do apply arithmetic knowledge (factual knowledge, schematic applicable procedures). This level comprises knowledge learned by heart that is easy recallable and can be applied directly in a standard situation. The problem which is solved points to a certain standard form of mathematization from the outset. Conceptual modeling is not required.

Example 1 *Please compute the derivation of*

$$f(x) = 3x^5.$$

To obtain a more difficult problem, one could use e.g. higher numbers or negative exponents.

Level II: Simple conceptual solutions The simplest forms of conceptual modeling and solutions that include only a few steps are involved as well as sufficient factual knowledge. Either the task is to select the correct solution from several alternatives or the students are provided with structural aids, graphical hints, etc. to develop an own solution.

Example 2 *In the table you find some data of a car.*

time (sec)	*20*	*40*	*60*	*80*	...
distance (km)	*0,02*	*1,2*	*3,5*	*5*	...

Velocity can be defined as rate of change of displacement. In mechanics the average speed v of an object moving a distance d during a time interval t is described by the simple formula:

$$v = \frac{d}{t}.$$

Please compute the average speed up to 40 sec.

The level of difficulty changes, e.g. if the formula/unit of velocity is deleted/given.

Level III: Challenging multi-step-solutions Students at this competency level are able to perform more extensive operations, and are able to solve a problem with several intermediate steps. Students are also able to deal with open-ended modeling tasks that can be solved in various ways, but that require to find a solution of their own. High level modeling on inner-mathematical connections can be asked for.

Example 3 *Find the maximum of*

$$f(x) = \frac{2x}{1 + x^2}.$$

Level IV: Complex processings (modelings, argumentations) Students, who solve exercises of this final competency level successfully are able to work on open-ended tasks, choose adequate models and construct models themselves where necessary. Conceptual modeling at this highest level often includes mathematical justification and proof as well as reflection on the modeling process itself.

Example 4 *Prove that the derivative of a hyperbola - not using the formula.*

$$f(x) = \frac{1}{x}$$

is

$$f'(x) = -\frac{1}{x^2}.$$

Proof (not shown to the students):

$$\frac{\frac{1}{x} - \frac{1}{x_0}}{x - x_0} = \frac{x_0 - x}{xx_0} \times \frac{1}{x - x_0} = \frac{-1}{xx_0}.$$

If x converges to x_0 the assertion is correct.

Another example is an open-formulated optimization exercises.

4.3.7 Typical Learning Time

Purpose approximate or typical time it takes to work with the resource [9].

Name `typicallearningtime`

Relation to existing metadata standards adopted from LOM, refined with LeActiveMath attributes *min* and *max* for minimal and maximal times additional to the general average value.

LeActiveMath **values** ISO8601: xx:yy:zz (e.g., 01:30:00), the same for values the *min* and *max*.

Applies to all resources

Number of occurrences one or more per resouce (possibly annotated with learning context).

Note a better name for this metadata is "average time on task" but we chose to adopt names in existing standards.

4.3.8 Exercise Type

Purpose refers to the type of interactive elements used in the content of the exercise

Name `exercisetype`

Relation to existing metadata standards does not exist in standards.

LeActiveMath **values** 'mcq_sigle_answer', 'mcq_multiple_answer', 'fill_in_blank', 'puzzle', 'drag_and_drop', 'concept_map'.

Applies to exercises.

Number of occurrences one or more

Note this metadata can be generated automatically from the exercise representation.

4.3.9 Exercise Purpose

Purpose refers to the target educational purpose of an exercise which can, for example, specify whether an exercise is suitable for assessment

Name `exercisepurpose`

Relation to existing metadata standards does not exist in standards.

LeActiveMath **values** 'training', 'assessment'

Applies to exercises.

Number of occurrences one or more

Note in some cases this metadata can be generated automatically from the exercise representation.

4.3.10 Interactivity Type

Purpose refers to the type interactivity of an item

Name `interactivitytype`

Relation to existing metadata standards taken from LOM

LeActiveMath **values** 'active', 'exposive', 'mixed' and 'undefined'

Applies to exercises.

Number of occurrences one per exercise.

4.3.11 Interactivity Level

Purpose refers to the type interactivity of an item

Name `interactivitylevel`

Relation to existing metadata standards taken from LOM

LeActiveMath **values** 'very_low', 'low', 'medium', 'high', 'very_high'.

Applies to exercises.

Number of occurrences one per exercise.

4.3.12 Representation

Purpose represents the principal representation type of the element. Serves to offer a rich mixture of types and learning opportunities.

Name `representation`

Relation to existing metadata standards does not exist in other standards, but the need for it was confirmed in personal discussion several times.

LeActiveMath **values** `verbal, visual, numeric, symbolic`

Applies to all resources.

Number of occurrences one per resource.

Note this is different from abstractness as very abstract exercises can be posed completely verbal.

4.3.13 Abstractness

Purpose the measure of abstraction of the content item.

Name `abstractness`

Relation to existing metadata standards does not exist in other standards, but the need for it was confirmed in personal discussion of LeActiveMath consortium.

LeActiveMath **values** 'abstract', 'neutral', 'concrete'

Applies to all resources.

Number of occurrences one or more per resouce (possibly annotated with learning context)

4.3.14 Relation

Relations in LeActiveMath can be of various types. We take the values for these types partially from Dublin Coreand LOMand add some LeActiveMath values. The values are normalized w.r.t. LeActiveMath syntax for values.

Adopted from LOMare the following pairs of types of relations: 'references', 'is_referenced_by', 'prerequisites', 'is_prerequisite_for'.

For each pair of mutually inverse relations, only one of them needs to be authored, the reverse relations can be generated automatically.

An additional pair of types of relations in LeActiveMath is 'is_special_case_of' and 'has_special_case'.

Other LeActiveMath values such as 'counter', 'for', and theory/imports- mechanisms of `OMDoc` as another kind of relation are described below in more detail.

Below we define the LeActiveMath usage for each of the types of relations in detail. The Dublin Coredefinitions used are taken from the Dublin CoreMetadata Registry.[10]

prerequisite / is_prerequisite_for

Purpose serves to represent educational prerequisites for the current resource.

Relation to existing metadata standards not present in standards

LeActiveMath **values** the ID or the list of IDs of concepts the current resource relates to.

Applies to all resources.

Note the inverse relation is automatically computed by the system.

references / is_referenced_by

Purpose the described resource references, cites, or otherwise points to the referenced resource [2]. In LeActiveMath it serves to represent educational references between items.

Relation to existing metadata standards name adopted from Dublin Core/LOM

LeActiveMath **values** the ID or the list of IDs of concepts the current resource relates to.

[10] see http://dublincore.org/dcregistry

Applies to all resources.

Note the inverse relation is automatically computed by the system.

The difference between LeActiveMath relations 'educational_prerequisite', 'references' and 'domain_prerequisite' is as follows. The first one means that the current concept is educationally based upon the concept it relates to, but the related concept is not necessarily a prerequisite required for understanding the current concept. For example, the notion of a group is based upon the notion of a monoid, but can be defined without using this notion. The value 'references' means just an explicit reference suggesting the usefulness of the related item in context of the current one. The value 'domain_prerequisite' means that the referred item is a mathematical prerequisite for the current one.

It can happen that satellites also have prerequisites (additional to those of the concept they are for). For instance, exercises that continue previous exercises or require additional results, e.g., an exercise for the quotient rule: "Compute the derivative of tan=sin/cos". Knowing the derivatives of sin and cos are prerequisites for solving this exercise, but not for the corresponding concept "quotient rule".

is_part_of / has_part

Purpose the described resource is a physical or logical part of the referenced resource [2] (reverse for the reverse relation).

Relation to existing metadata standards adopted from Dublin Core/LOM

LeActiveMath **values** the ID or the list of IDs of concepts the current resource relates to.

Applies to all resources.

Note the inverse relation is automatically computed by the system.

is_special_case_of / has_special_case

Purpose the content of the described resource is a special case of the referenced resource (reverse for the reverse relation).

Relation to existing metadata standards does not exist in standards

LeActiveMath **values** the ID or the list of IDs of concepts the current resource relates to.

Applies to all resources.

Note the inverse relation is automatically computed by the system.

4.4 Metadata and Diagnosis for Exercise Steps

In order to assess the student's mastery the steps of interactive exercises have to be annotated with additional information about the concepts and their competencies, connected to each step. Here we define the set of metadata elements for annotating the steps of interactive exercises and the set of diagnosis elements, assessing the learner mastery in each step. All of these additional data are optional. Not every step of the exercise but only some "important" ones might need to be annotated.

In each step the exercies system sends the exercise step event. This even has the following parameters:

- String value of the exerciseId,
- List of metadata elements represented as jdom objects
- List of user input omobjects represented as jdom ombjects
- diagnosis element represented as jdom ombjects

4.4.1 Metadata Elements in Interactions

Competency

Purpose describes the mathematical competencies a step of an exercise trains

Name `competency`

LeActiveMath **values** as defined in 4.3.5

Number of occurrences one per interaction

Each of the competencies might have subcompetencies. The values of subcompetencies are also defined in 4.3.5 and are represented as subvalued of competency elements. Example:
`<competency value='argue' subvalue='understand_a_given_proof'/>`

Competency Level

Purpose describes the different levels of mathematical skills a step of an exercise requires/trains

Name `competencylevel`

LeActiveMath **values** as described in 4.3.6

Number of occurrences one per interaction

Relation of type 'for'

Purpose describes the relation to concepts for which some competency in the current exercise step is trained.

LeActiveMath **values** the ID or the list of IDs of concepts the current resource relates to.

Number of occurrences one or more per interaction

Difficulty

Purpose how hard it is to work through the exercise step for the typical target audience (the target audience is given via learning context)

Name `difficulty`

LeActiveMath **values** as defined in 4.3.3

Number of occurrences one per interaction

Abstractness

Purpose the measure of abstraction of the exercise step

Name `abstractness`

LeActiveMath **values** as defined in 4.3.13

Number of occurrences one per interaction

Typical Learning Time

Purpose estimate the time needed by the user for this interaction step. Also, after finishing this step the Learner Model can compare the actual needed time to this value and draw appropriate conclusions.

Name `typicallearningtime`

LeActiveMath **values** as defined in 4.3.7

Number of occurrences one per interaction

4.4.2 Diagnosis Elements in Interactions

Each `condition` in the `answer_map` element of the exercise `interaction` can be annotated with `diagnosis` on the user's action in this step. The `diagnosis` element consist of several child elements described below.

Relation to a misconception

Purpose a type of relation that describes the relation to the misconception which occured in the current step.

LeActiveMath **values** the ID of the misconconception the current resource relates to.

Number of occurrences one or more per diagnosis record

Achievement

Purpose This diagnosis element is used to specify the measure of achievement of the learner in his action w.r.t. the task. For instance, if the task of the learner was to apply the chain rule and the learner made a typical error and applied a so-called buggy-chain-rule, then his achievement rate w.r.t. the task of applying chain rule is low.

Name `achievement`

LeActiveMath **values** 0.0 to 1.0

Number of occurrences one per diagnosis record

Relevance

Purpose This diagnosis element is used to specify the measure of relevance of the learner's action w.r.t. the task. For instance, if in the step the learner has repeated the problem statement of the task, he has made no error, but this step was irrelevant for the solution process so the relevance of this step is low.

Name `relevance`

LeActiveMath **values** 0.0 to 1.0

Number of occurrences one per diagnosis record

4.5 XML-Binding

The general representation of Dublin Coremetadata is adopted from `OMDoc`. The only differences occur when using metadata not defined in `OMDoc`. Among the Dublin Coreelements we only refined `Rights` and `Relation`.

The `Rights` element in LeActiveMath can have three child elements - `permissions`, `prohibitations`, and `requirements` with values, described above in the section 4.1.

The `Relation` element has a type attribute, specifying the type of relation used and contains one or more `ref` elements pointing to the related resource.

For other LeActiveMath metadata, different from Dublin Core, the general scheme of the XML-binding of the metadata is as follows:

```
<metadata-name value="value-name"/>
```

The metadata, using one of the Classification Schemes will have an additional attribute *schema* with values 'LCSH', 'DDC' and 'MSC' e.g.,

```
<field schema="DDC" value="514"/>
```

Multiple values are represented by multiple occurrences of the corresponding element with each occurrence having one of the values, e.g.,

```
<field schema="DDC" value="514"/>
<field schema="DDC" value="515"/>
```

Below is an example of a (multiple) metadata record assigned to an exercise:

```
<metadata>
 <Title xml:lang="en">An easy exercise about limits</Title>
 <Creator role="aut">George Goguadze</Creator>
 <Relation type="for">
  <ref xref="limit1/limit">
 </Relation>
 <learningcontext value="University first Cycle"/>
 <difficulty value="easy">
 <abstractness value="neutral"/>
<metadata>
```

```
<metadata>
 <Title xml:lang="en">Hard, but interesting exercise about limits</Title>
 <Creator role="aut">George Goguadze</Creator>
 <Relation type="for">
  <ref xref="limit1/limit">
 </Relation>
 <learningcontext value="Secondary Education"/>
 <difficulty value="medium">
 <abstractness value="abstract"/>
<metadata>
```

References

[1] ARIADNE Foundation. ARIADNE educational metadata recommendation - v3.2. Technical report, ARIADNE Foundation, 2002.

[2] The Dublin CoreMetadata Initiative. Dublin Coremetadata initiative - home page, 1998. http://purl.org/DC/.

[3] Duval, E., Hodgins, W., Sutton, S., Weibel, S.L., "Metadata Principles and Practicalities", D-Lib Magazine, April 2002, Volume 8, Number 4, ISSN 1082-9873

[4] Goguadze, G., González Palomo, A., "Knowledge Representation of Interactive Exercises in ActiveMath", SEKI-Report, Nr. SR-04-10, University of Saarland, December 2004.

[5] Goguadze, G., Melis E., Cairns, P., Ullrich, C., " Problems and Solutions for Markup for Mathematical Examples and Exercises", In Proceedings of the Second International Conference on Mathematical Knowledge Management, MKM03, Andrea Asperti (ed.)., 2003.

[6] Hutter, D., "Reasoning about theories", Deutsches Forschungszentrum für Künstliche Intellligenz (DFKI), 1999

[7] Kohlhase, M., "OMDoc: Towards an OpenMath Representation of Mathematical Documents", Seki Report,FR Informatik, Universität des Saarlandes, 2000.

[8] E. Klieme, H. Avenarius, W. Blum, P. Döbrich, H. Gruber, M. Prenzel, K. Reiss, K. Riquarts, J. Rost, H. Tenorth, and H. J. Vollmer. The development of national educational standards - an expertise. Technical report, Bundesministerium für Bildung und Forschung / German Federal Ministry of Education and Research, 2004.

[9] IEEE Learning Technology Standards Committee. 1484.12.1-2002 IEEE standard for Learning Object Metadata, 2002.

[10] Melis, E., Büdenbender, J., Andres, E., Frischauf, A., Goguadze, G., Libbrecht, P., Pollet, M. and Ullrich, C., "ActiveMath: A Generic and Adaptive Web-Based Learning Environment", Artificial Intelligence and Education, Volume 12, Number 4, 2001.

[11] Melis, E., Goguadze, G., "Towards Adaptive Generation of Faded Examples", International Conference on Intelligent Tutoring Systems (ITS), 2004.

[12] M. Niss. Mathematical competencies and the learning of mathematics: the danish KOM project. Technical report, 2002.

[13] "The Open eBook Group. Open ebook[tm] publication structure" 1.0 Draft recommendation, The OpenEBook Initiative, 1999. http://www.openEbook.org

[14] "The OpenMath Standard" version 2.0, The OpenMath Society, edt. S. Buswell, O. Caprotti, D. P. Carlisle, M. C. Dewar, M. Gaëtano and M. Kohlhase, June 2004