Deliverable N°: D5

Requirement Analysis

The LeActiveMath Consortium
June 2004

Main Authors:
LeActiveMath-Partners, responsible DFKI

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	Restricted
Contractual date of delivery	June 30th 2004 (Month 6)
Actual date of delivery	June 30th 2004
Deliverable title	Requirement Analysis
Type	Report
Status & version	1.0
Number of pages	31
WP contributing to the deliverable	WP 1
WP/Task responsible	T1.1, T1.2, T1.3, T1.4, T1.5, T1.6
Author(s)	LeActiveMath-Partners, responsible DFKI
EC Project Officer	Hans-Georg Stork
Keywords	Requirement analysis

Contents

1 Introduction

For the requirement analysis of LeActiveMath, the consortium of the LeActiveMath project adopted Carroll's *Scenario-Based Design and Claims Analysis* methodology [1, 2] which was already used successfully in the EU-project NIMIS [3] and suggested by Paul Brna. The pedagogical claims methodology involves the construction of a form of design rationale. The methodology associated with pedagogical or didactic claims is an adaptation of a design methodology that was first suggested for user interface design by [1]. In a later paper [2] the pedagogical claims were produced from a participatory design process which involved teachers from an early stage.

The methodology includes describing a learning scenario, looking for activities and valuable outcomes, selecting key moments in a scenario for special attention,[1] speculating on how a design decision might affect the situation.

Ideas for the claims came from several places (project's description of work (DoW), experience with related work, literature, etc.). It is important to judge the likely effects of ideas being implemented. This will help to avoid large changes after the first implementations. Moreover, the problematic issues have to be faced from the beginning because they may prevent a useful implementation altogether.

A large number of claims has been generated from user stories directly and then (recursively) for the content and for key components such as exercise repository, tutorial component, dialogue manager, learner model, the architecture, several tools, and some of their interfaces.

This document – as a deliverable of the project – is a selection of the claims that were considered essential by the consortium. These turn into *requirements*. It provides an account that is structured according to the work packages of the project's DoW. Note, that at places a claim stated in one section could belong to another one as well, e.g., metadata claims are in section Knowledge Representation but could also belong to the section Content.

As an explanation for the reader, since LeActiveMath will be a complex system we might not have described all the requirements. Moreover, some of the claims cannot be realised in LeActiveMath because this would exceed the resource limits (and DoW) considerably. That is, they cannot be evaluated for LeActiveMath but maybe realised by other related projects and so, they were not selected as requirements and are not contained in this official document.

For the requirements we use the following schema. The structure of the document already indicates to which work package and tasks each claim belongs primarily.

Requirement 1.1	Design feature.
Supports	Pedagogical (or system) aim.
Because	Connecting reason.
Check-rule	Method of evaluation.
Issues	Possible problematic issues.

Abbreviations

- Learner Model (LM)
- Open Learner Model (OLM)

[1]key means important to achieve a good result, important problem to avoid

- Situation Model (SM)
- Learner History (LH)
- Dialogue Manager (DM)
- Tutorial Component (TC)
- Content Knowledge Base (KB)
- Natural Language (NL)
- Natural Language Generation (NLG)
- Natural Language Understanding (NLU)
- OpenMath Content Dictionary (CD)
- Multiple Choice Question (MCQ)
- Fill in Blank Exercise (FIB)
- Computer Algebra System (CAS)

More abbreviations of some standardisation organisations and standards.

- IEEE and IEEE-LTSC Learning Technology Standards Committee `http:/ltsc.ieee.org/`
- IMS Global Learning Consortium which describes (CP content packaging, SS simple sequencing, QTI Question and Test Interoperability etc.) `http://www.imsglobal.org/`
- LOM (Learning Object Metadata) developed out of recommendations of the European ARIADNE project and foundation `http://www.ariadne-eu.org/` and IMS
- Educational Modeling Language (EML) `http://eml.ou.nl/`
- Dublin Core Initiative (DC) `http://purl.org/DC/`

2 Knowledge Representation

The XML-based language, OMDoc, will be the basis for the knowledge representation in LeActiveMath. It has to be extended in several dimensions, e.g., metadata and structure of instructional items.

This section concentrates on metadata as a part of the knowledge representation. Metadata are important for realising adaptivity, for the diagnosis in learner modelling, and for reusability of learning objects. Other sections will point to representation needs as well, e.g., §7.

Requirement 2.1	Metadata of instructional items have to comply with standards. If not, it has to be clearly substantiated.
Supports	Reusability and standards.
Because	There are tools for supporting standards and reuse of learning objects require standards.
Check-rule	Report on metadata has to explain the cases, where standards are not met.
Issues	Currently, standards are too large and may not include everything necessary.

Requirement 2.2	Metadata should contain the Dublin Core subset.
Supports	Basic administration of resources.
Because	Dublin Core is compact and well-suited for basic administration purposes.
Check-rule	
Issues	Needs extensions for proper copyright metadata.

Requirement 2.3	Metadata have to characterise competencies - learning activities such as model, compute and test.
Supports	• Adaptation to competency-based learning scenario. • Generation of well-balanced courses in terms of competencies.
Because	TC can choose and assemble appropriate learning objects.
Check-rule	Check exemplary learning objects and overall content.
Issues	• More annotation means more work for authors. • Complex exercises may have to be split. • Not yet in all standards.

Requirement 2.4	Metadata should indicate suitable *learning context* and learning object's connection to a *field* outside of mathematics.
Supports	Adaptation to learning context and interest of learner.
Because	TC and DM can use the metadata.
Check-rule	Check content.
Issues	• More annotation means more work for authors. • Learning context needs proper set of values suitable for all countries or has to be defined by pairs (context, country). • 'Field' not in standards.

Requirement 2.5	Metadata should characterise representation forms (verbal, graphical, numeric, symbolic) and abstractness.
Supports	Adaptation to cognitive variables.
Because	If learner does not understand one form, another form can be chosen for or by her.
Check-rule	Check content.
Issues	• Work for authors. • Full automatic adaptation to learner style is questionable. • If abstractness will be used it has to be exactly defined (maybe it depends on learning context, etc.).

Requirement 2.6	Metadata should support relations of different kinds.
Supports	Mathematical and educational ontologies.
Because	Semantic relations between learning objects are essential for learning, knowledge management and visualisation.
Check-rule	Check content.
Issues	• Multiple definitions can be linked to the same concept. This makes management much more difficult. • Avoid confusion of educational and purely domain-related relations.

3 General Technical Requirements

This section contains the requirements for the overall basic system and its tools corresponding to workpackage three. Requirements for the components learner model, exercise repository, and natural language facilities are listed in their respective sections.

Requirement 3.1	The server software developed in LeActiveMath should be able to run on Internet Explorer and Netscape/Mozilla platforms.
Supports	Widest usage and testing.
Because	If system were platform-specific, deployment would be reduced and organisation of testing more difficult.
Check-rule	Software tests.
Issues	This requirement restricts the external software that can be used (including commercial tools for prototyping).

Requirement 3.2	The client software should run on the Web, have a friendly installation, and be cross-platform.
Supports	Dissemination, widest usage and testing.
Because	Limitations of schools.
Check-rule	Software tests.
Issues	• Not clear yet whether more than Mozilla and Java 1.3. will be required by LeActiveMath components. • Maybe the equipment of schools varies a lot, maybe more than we expect. Klett will find out.

Requirement 3.3	LeActiveMath is multilingual except the prototypical dialogues.
Supports	Europe-wide dissemination and usage.
Because	Learning maths in a foreign language may produce an additional obstacle and overload.
Check-rule	Record and evaluate the questions of learners.
Issues	• This requirement refers to content in different languages and to an internationalised system. • The translations of content should be of high quality. • This shall be only evaluated when the full content is ready to be presented.

Requirement 3.4	Usability of the system includes: rendering of maths-specific parts (readability of mathematical terms) as well as more general GUI-requirements (colours, fonts, structure, navigation, etc.).
Supports	Focusing the learner on learning rather than understanding the user interface.
Because	Human-computer interaction.
Check-rule	Usability tests.
Issues	

Requirement 3.5	Components need to be accessible by standard communication protocols such as XML-RPC.
Supports	Communication between separate modules and reliability of the components.
Because	Standards should be used.
Check-rule	Proof of existence.
Issues	Performance issues when communication of the net.

Requirement 3.6	Query functionality: components of LeActiveMath can query the knowledge base(s) for items with particular metadata, types, (multiple) relations. It returns an ID or a list of IDs.
Supports	Separation of the functionalities of the different LeActiveMath components.
Because	Different components (TC, DM, evaluation) will need information from the knowledge base(s) and LM/SM.
Check-rule	Check architecture.
Issues	Mapping between ontologies may be necessary.

4 Components and Tools

4.1 Notes

Prototypical notes exist in the previous ActiveMath (AM1). They have to be improved for LeActiveMath.

Requirement 4.1	Students can make private and public notes.
Supports	Public notes support student-student and student-teacher interaction. Private notes increase motivation, can provide individual help, support remembering previous thoughts/ideas/insights.
Because	Student can state individual view better than with a printed book. Can be used as a communication means.
Check-rule	Test how note are used by learners.
Issues	• The availability of notes has to be indicated to the learner. • Public notes may need approval.

4.2 Input Editor

The following reflects requirements as stated in the tender and contract of the subcontract.

Requirement 4.2	Integration of Input Editor in LeActiveMath.
Supports	Student's input of mathematical expressions in appropriate input formats.
Because	Facilitates input of mathematical terms.
Check-rule	Proof by existence.
Issues	• Difficult to please every student/author/exercise. • There may be a restriction to frequently used terms, like integrals, vectors, etc. More complex terms like e.g. matrices should be handled with the classical way of multiple choice. We must find balance between our wish "to supply a real situation" and to avoid "keyboard-acrobatic" by the user.

Requirement 4.3	The Input Editor's output is OpenMath.
Supports	Semantic output.
Because	OpenMath was designed to serve as the lingua franca (semantic XML) for maths systems.
Check-rule	Proof of existence.
Issues	

Requirement 4.4	Variety of input possibilities: OpenMath input, (e.g., paste/drop or copy & paste from LeActiveMath) (theory-based) pallette-based structured input facility, as well as linear keyboard input.
Supports	Needs of different users.
Because	Users have different experiences and skills.
Check-rule	Proof of existence.
Issues	

Requirement 4.5	A priori selection of theory (e.g., via button) in Input Editor.
Supports	Restrict the choices and disambiguate the input.
Because	Theory restricts set of symbols and OpenMath CDs.
Check-rule	Proof of existence.
Issues	Overlap and inheritance between theories.

Requirement 4.6	Input editor has to guarantee extensibility and include creation tools for new OpenMath CDs.
Supports	Extends usage to several maths domains and new symbols.
Because	Mathematics should not be restricted.
Check-rule	Proof of existence.
Issues	

Requirement 4.7	The input has to be converted to a proper mathematical symbol presentation in the user interface. For new symbols xslt-presentation rules can be created by user.
Supports	Reading of mathematical expressions.
Because	Proper rendering of mathematical expressions.
Check-rule	Proof of existence.
Issues	There should be default rules for new symbols (most users are not experts in xslt), e.g. infix, prefix, sub and sup, with or without brackets.

4.3 Dictionary

A prototypical dictionary exists in the previous ActiveMath (AM1). This has to be improved for LeActiveMath.

Requirement 4.8	Dictionary offers access to all the mathematical content items.
Supports	Explorative learning; access to knowledge.
Because	The content should be available not only in form of a course.
Check-rule	Test.
Issues	The user interface must be simple and usable.

Requirement 4.9	Dictionary search is reasonably tolerant.
Supports	User satisfaction.
Because	Strict search will fail too provide some relevant results.
Check-rule	Observe learners using the dictionary.
Issues	• The relevance of search results may be difficult to rank. • Learners may have to be taught how to use the dictionary.

4.4 Presentation

Requirement 4.10	Technically modular presentation component.
Supports	Reuse of component.
Because	Could be employed for different purposes in LeActiveMath.
Check-rule	Try to use the tool for different presentations.
Issues	Component needs to be efficient and general.

Requirement 4.11	It must be easy to obtain a printable version.
Supports	Work less dependent on temporary availability of computers. Moreover for student's preferences.
Because	Current experience shows that online and hard copy versions are used in parallel or mixed.
Check-rule	Collect views from teachers and students.
Issues	Work with print version and other external material cannot be monitored by system. This could lead to problems with the assessment tool and the user model. Maybe an active request for the inspection and modification of the user model could help.

4.5 Tutorial Component

Requirement 4.12	The tutorial component allows to represent and execute different pedagogical strategies, e.g. problem-based, traditional didactical approach.
Supports	Different ways of teaching/learning.
Because	Different ways of teaching use different pedagogical strategies.
Check-rule	Empirical test with small groups using different strategies.
Issues	• Long-term effects are hard to evaluate. • Pedagogical strategies can become very complex and hard to formalise. • Large collections of learning objects are required.

Requirement 4.13	Allow for reactivity: tutorial component should react to learner's progress and problems.
Supports	Individual needs of the student.

Because	Course generation (which happens before the student accesses the course) may be based on assumptions that change during learning.
Check-rule	Test with small group of students (reactivity vs. non-reactivity).
Issues	• Changes to the course have to be indicated to the learner. • Reactivity may create navigation problems.

Requirement 4.14	Allow for interactivity: student should be able to ask (actively) for specific additional content and integrate it into her course.
Supports	Individual needs of the student. More interactive and self-directed way of learning.
Because	Active requesting.
Check-rule	Test: do learners use the feature?
Issues	• Maybe this feature is appropriate only in some pedagogical scenarios. • Interactivity may create navigation problems. • Do we need NLG for this feature?

Requirement 4.15	Need ontology of instructional objects.
Supports	• Integration of third-party content. • Provides adequate level of abstraction for authors/learners to talk about learning materials.
Because	There exist many knowledge representations of e-learning content and using a shared ontology supports integration. Users do not want to know about internal knowledge representations.
Check-rule	Check: can the learning material be described by the ontology? Do users understand the ontology?
Issues	• Semantics of ontologies is partly not properly defined. • Mapping of ontologies may be ambiguous.

Requirement 4.16	TC generates symbolic representations for transitions, introductions, and summarisations that the NLG front-end then verbalises into English (WP5).
Supports	Better readability and coherence.
Because	Transition and summarising texts make the transition between the learning objects smoother.
Check-rule	Test effect with small group.
Issues	Canned text can be too boring.

Requirement 4.17	The TC offers scenarios that target competencies.
Supports	Competency-level pedagogy as used in PISA and other studies.
Because	LeActiveMath should be able to conform with didactical standards.
Check-rule	Compare results with national curricula and courses that comply with such standards.
Issues	• Formalisation and encoding of competency standards. • Needs appropriate content. • Variation in different European countries (has to be discovered by Augsburg in the first place).

Requirement 4.18	The TC needs to have access to all instructionally relevant information about the user.
Supports	Adaptivity.
Because	Without information about the user, the TC cannot provide adequate adaptations.
Check-rule	
Issues	• The relevant information includes the learner's current knowledge state, her preferences, her history, her learning/cognitive style, misconceptions, static profile and traits. • We may need for an ontology similar to the ontology of instructional objects in case we want to switch between different learner models that use different representations of individual information.

Requirement 4.19	Action selection and presentation is determined by the TC (and DM) which in turn are informed by the LM/SM.
Supports	Adaptation of content and form of interaction.
Because	• Tutorial actions should depend on the current situation. • Guidance and encouragement (autonomy and approval) depend on such variables. • Factors such as exercise difficulty, achievement and confidence are not independent of each other.
Check-rule	Sensitivity study of the effect of varying factor values on content choice, expert evaluation by teachers.
Issues	• Operational definitions of factors; defining dependencies between factors. • Determine appropriate set of pedagogical and communicative strategies and other actions. • Ratings of possible actions and strategies. • Generation of actions and application of strategies by TC.

4.6 Interactive Concept Map

Requirement 4.20	Provision of a mind map tool.
Supports	The construction of dependencies of concepts in learner's mind.
Because	Helps students to visualise dependencies and have an individual way of approaching a topic.
Check-rule	Test how students use the tool.
Issues	• There are two usages of such tool: one for free brainstorming and another for exercises requesting the construction of a map, in which feedback is given. • Students may not understand the intention of the tool and would have to be instructed.

4.7 Assessment Tool

Requirement 4.21	Assessment tool should be able to run a variety of exercise types and exercises with several competencies.
Supports	Diagnosis of different competencies.
Because	If MCQs are used only, then assessment of knowledge and competencies is limited.
Check-rule	
Issues	• May require major changes to Siette. • Currently, Siette is envisioned as assessment tool only rather than as 'the exercise system' of LeActiveMath.

Requirement 4.22	Integration of assessment tool into LeActiveMath and communication with other components (learner model, tutorial component).
Supports	A single learning environment with several services.
Because	Initialises learner model, test can be requested by tutorial component.
Check-rule	Architecture and test.
Issues	• Translation of data structures. • Communication protocol. • Changes in Siette. • Assessment may cause replanned curriculum.

4.8 Assembly Tool

Requirement 4.23	The assembling tool will allow the student to produce her own copy of learning material from LeActiveMath 'books' and external objects and to annotate it with personal notes.
Supports	Planning of learning, memorisation, and self-regulated learning.
Because	Because of the act of producing/assembling material herself.
Check-rule	Tests how students use the tool.

Issues	• Requires a data structure for manually adding/deleting of `OMDoc`-elements and a database for elements created by student. • May be too difficult for novices without instruction by teacher. • May require too much server space.

5 Learner Model

Requirement 5.1	The learner model (LM/SM) will feature beliefs about the learner's knowledge, skills, competencies, competency levels, academic interests, media competencies, affective and motivational states.
Supports	The selection of pedagogical strategies and tactics to support the learner's progress.
Because	Other components of the system, such as the DM and the TC can utilise these beliefs to adapt the available options and to discuss matters with the learner more appropriately.
Check-rule	The existence of the LM, whereas the quality of the beliefs will be checked by some appropriate empirical method.
Issues	• The selection of particular aspects to hold beliefs on. • The usual problems of maintaining consistency and updating the learner model.

5.1 Related to Learner History

Requirement 5.2	The LM will need to access the LH to analyse the past history of interaction.
Supports	Looking for patterns in the interaction.
Because	We want to detect the learner's shifts in behaviour and determine, if these are indicate learning or something retrograde.
Check-rule	The proof will be in terms of the patterns that can be detected reliably so checking could be via simulating students learning with LeActiveMath.
Issues	Some patterns may be time-dependent so we need the LH with sufficiently accurate time stamped data.

Requirement 5.3	The LM will have access to time stamped records of learner behaviour inside sessions, ranging from some low-level interface events to high level interpretations of learner behaviour (such as content review and performance at exercises).
Supports	Derivation of relevant features and patterns in LM which, in turn provide support for system adaptation to learner needs.
Because	Learner's shifts in behaviour needs to be detected as well reasons for the beliefs in LM.
Check-rule	Empirical tests to find out whether records are appropriate and sufficient to infer relevant information and patterns.
Issues	• Low-level of action may need to be recorded at client side. • Interfaces needed for the LM to access this information. • There may be problems with translating from low-level events to higher-level ones.

Requirement 5.4	Updates to the LM beliefs should be supported by information stored in the LH.
Supports	Justification of the learner model beliefs by learner behaviour.
Because	Transparency is very important for acceptance by students and teachers.
Check-rule	The explicit representation of the interpretation of the information in the LH as well as the provision of this evidence when necessary (in the open learner model).
Issues	• The design of justifiable interpretation. • Depends on how much explanations the (open) learner model is supposed to provide.

5.2 Realated to Open Learner Model

Requirement 5.5	The OLM will be able to present the learner with beliefs about the learner's knowledge, skills, competencies, competency levels, academic interests, media competencies, affective and motivational states.
Supports	The learner thinking about what they know, their interests, preferences, competencies, affective and motivational states, promoting in this way the development of metacognitive skills.

Because	A learner can reflect on the domain being learnt, as well as on their affective state and motivation for learning, acquiring in this way metacognitive knowledge and skills. In addition, the open learner model may be a way for learners helping the system to improve its learner models.
Check-rule	The existence of the corresponding interfaces, as well as qualitative semi-structured interviews which may show the effects on learners.
Issues	The possibility that the learner would consider the learner model too inaccurate, it has to be decided what should be accessible by learners and the extent and mechanisms by they can alter the learner model.

Requirement 5.6	The open learner model provides the learner with a user interface(s) for displaying and manipulating each of the various aspects of the learner's model properly (e.g. student's domain understanding, affective state, motivation, etc.).
Supports	Focusing the student on different aspects of her learning.
Because	It would be confusing to mix these aspects together and each of these aspects may require a particular mode of representation.
Check-rule	It would be possible to do an experimental study based on two different interfaces
Issues	There are a number of issues in defining how to present the different parts of the OLM and what is an adequate representation.

Requirement 5.7	The OLM user interface provides mechanisms supporting negotiation with the learner about the beliefs stored in the LM.
Supports	Accountability of the learner model on the face of the learner, as well as its accuracy, helped by the learner. It also supports metacognition.
Because	The system will need to provide evidence justifying the beliefs in the learner model, and vice versa, the learner will need to justify their claims by providing their own evidence. In addition, negotiating the system beliefs, as stored in the learner model, may encourage learners to think more deeply about the nature of domain knowledge and their comprehension of it.
Check-rule	The existence of the facilities for negotiation should be complemented with empirical studies of their effect on the learner model accuracy and trustworthiness, as well as the level of learner engagement achieved by the negotiation and its effects.
Issues	Negotiation is a complex activity, which demands some sort of natural dialogue structure, even if not carried out in natural language. There is also the issue of who is going to have the last word, and the consequences of it.

Requirement 5.8	The OLM user interface may hide some of the beliefs found in the LM.
Supports	Preserving the "face" of learners and preventing overload.

Because	Too much "honesty" might be depressing/demotivating.
Check-rule	Experimental study based on two different interfaces.
Issues	Research problem: this is a serious and difficult issue - lots of problems in deciding, in a principled way, what to hide. Also the hiding must be adaptive, but how?

5.3 Related to Communication and Interfaces

Requirement 5.9	The LM information helps initialising some properties of exercises for the TC and DM. For instance, the LM will help to derive estimates of the appropriateness of the exercise for the learner.
Supports	Selection of the next task (sequence of tasks) by the TC or DM that might be offered to the learner.
Because	The student may need a particular trajectory for her learning.
Check-rule	Proof by existence.
Issues	Decisions about choice of metadata in exercises.

Requirement 5.10	Self-assessment can be used to provide input to LM at various stages.
Supports	Initialising the learner model.
Because	The student should notice the adaptivity as an advantage.
Check-rule	Small lab experimental design.
Issues	The student's attitudes and preferences may change during her learning. By asking too many questions the students may feel disturbed.

Requirement 5.11	The SM provides the DM with values of student situation. for guidance and encouragement (autonomy and approval).
Supports	Adapting the dialogue? generation to the learner.
Because	• We need to communicate with learners in different ways to respond to their *affective* needs, e.g. less confident students may need more encouragement (i.e. more explicit approval). • We need to communicate with learners in different ways to respond to their *cognitive* needs, e.g. students achieving less may need more structured and detailed guidance (i.e. less autonomy).
Check-rule	Sensitivity study to show that the recommendations of values for autonomy and approval change according to the factor values in different situations.
Issues	Ratings of possible actions and strategies in terms of levels of autonomy and approval values.

6 Natural Language Facilities

According to workpackage five three natural language facilities will enhance LeActiveMath: exercise dialogue, natural language generator and learner model dialogue.

6.1 Natural-language enhanced dialogue

This type of dialogue will take place during learner's problem solving in exercises.

Requirement 6.1	Natural-language enhanced "local" tutorial dialogue.
Supports	Effective learning using student-adaptive and situation-adaptive feedback/dialogue.
Because	Linguistic cues give tutors information about the student's understanding of the material, and can be used to determine whether a tutorial strategy is working or needs adapting. NL dialogue offers both indirect and targeted feedback techniques for informing students about their performance, and remedying problems. Indirect feedback rather than direct negative feedback will improve student motivation.
Check-rule	Learning gains and student satisfaction (measured by questionnaire) are observed compared to tutoring the same material without natural-language enhanced dialogue interaction.
Issues	Dialogue needs to support student's progression and motivation and shall not discourage. It shall be given in an informative way and take the student's prior performance into account. Also, Feedback may not follow every little step. • Requires a rich representation indicating how humans solve mathematical problems and how tutors teach students to solve them. • Realistically, only exercises for a well-defined class of problems with appropriate reasoning and diagnosis can be supported. • Requires development of a mathematical input editor to allow student construction of solution and intermediate steps.

Requirement 6.2	DM needs access to the LM and SM information for realising the local tutorial dialogue.
Supports	The generation of student-adaptive and situation-adaptive hints.
Because	Dialogue has to take the student's prior performance and situation into account.
Check-rule	Test how student-adative and situation-adaptive dialogue affects student motivation and learning gains

Issues	• Definition of relevant information. • Mapping the values of the LM/SM to appropriate dialogue strategies and their corresponding/appropriate NL verbalisations.

Requirement 6.3	For local tutorial dialogue, the DM needs access to a rich representation of exercises as well as to domain reasoner.
Supports	The generation of problem state specific tutorial dialogue.
Because	Without a diagnosis tool that traces the student's behaviour through the set of possible solutions (and misconceptions), it will be hard to determine appropriate tutorial feedback
Check-rule	Test appropriateness of dialogue and feedback.
Issues	Alternatively, relevant information had to come from other sources that can be timconsuming, such as • Authored exercise solution graphs that contain frequent solutions for a given problem (for a restricted domain the dialogue works for). • Solution graphs enriched with buggy rules to allow the tracing of typical student misconceptions. • Solution graphs enriched with problem specific remedial feedback. • The natural-language enhanced tutorial dialogue will be feasible for a restricted domain of mathematics only.

Requirement 6.4	For local tutorial dialogue, the DM has to communicate with and interprete the learner's exercise area.
Supports	Natural language dialogue with the exercising learner who inputs English text in the chat window and terms/formulae in the input editor.
Because	Student's input in the chat window might mix English with mathematical formulae and mathematics input with the input editor.
Check-rule	

Issues	• The mixture of English and formulae creates problems for the NLU of student input. This may show to be quite complex from the interpretation point of view. Synchronising the input modalities may be needed. • The input via the input editor delivers OpenMath. It needs to be interpreted together with the English contents of the chat area. • We may need to restrict the student input.

Requirement 6.5	NL-Dialogue subsystem interfaces with and informs the LM and SM and the exercise component.
Supports	Proper update of information about the learner's progress, actions, goals.
Because	Interactions, progress, interpretable judgements, etc. take place during dialogue.
Check-rule	A small scale pilot study using qualitative methods could be used to check that the LM is being maintained in a plausibly accurate and timely manner.
Issues	• Determine exactly which information is available and can be passed. • It is highly likely that there will be inaccuracies. • LeActiveMath has to function properly also without a DM which produces English only.

6.2 Inspection of OLM through NL Dialogue

This type of dialogue is about the system's and learner's beliefs about the learner.

Requirement 6.6	Communication about LM through natural-language enhanced dialogue.
Supports	Student's inspection and modification of the learner model.
Because	NL-enhanced inspection of LM might be more effective than without NL-enhanced dialogue.
Check-rule	Test how NL dialogue can be more effective than LM inspection without dialogue.

Issues	• Pointing may be more effective than natural language. • From the NLU point of view, lots of complexities involved in using NL dialogue to help LeActiveMath to diagnose the student's state of knowledge or expertise. May require the capability to reason about how the OLM came up with its values in the first place. • Need corpus study to inform the design of such as learner model NL-enhanced dialogue component, and to determine the form of dialogue that we can realistically support. • The diverse content of the learner model will need a wider range of dialogue strategies than normally considered within ITS. • Strategy selection given dialogue context and learner model.

6.3 Natural Language Generation

The text will be generated for the assembled course.

Requirement 6.7	NLG will provide the TC with a front-end to verbalise introductions, transitions and summaries in English. The corresponding requirement from the TC side can be found in Section 4.5.
Supports	Better learning and readability.
Because	Advance organisers stimulate learning. Transitional texts make the course better readable.
Check-rule	Comparison of courses with/without NLG.
Issues	• Domain representation may have to be extended (it has already features such as concept-identifiers referring back to their definitions). • A NL lexicon may provide various ways to refer to domain concepts. • Representation of the curriculum may have to be enriched.

7 Exercises and Exercise Repository

The exercise repository is planned as an independent open component. It will contain interactive exercises and include search facilities. Since exercises are important ingredient of learning, they have additional requirements.

7.1 Exercises

Requirement 7.1	The exercise subsystem will enable the learner to perform various types of exercises
Supports	A variety of interactions and learning of various competenciesIncrease student motivation.
Because	Student needs to be actively confronted with problems in addition of the presentation of reading material.
Check-rule	Run a variety of authored exercises. Check with student whether and which types of exercises they use.
Issues	• Not everything will be possible without making exercise authoring too complicated, authoring highly interactive exercises seems to be very difficult. • Same applies for evaluation of user's actions in exercises.

Requirement 7.2	Some types of exercise should be supportable by back-engines which can serve as an aid for the student to compute the answer, to practice, or to verify online the correctness of a student's answer.
Supports	Interactive and exploratory exercising.
Because	
Check-rule	Check roles of back-engines in the exercises.
Issues	• In an exam students may not be allowed to use the back-engines. • Exercise system needs information as to which back-engines are available in a session, which back-engines can serve a particular exercise and the function of the back-engine. • Needs a way to code capabilities of back engines.

Requirement 7.3	Feedback has to be presented.
Supports	Supports discovery of errors.
Because	Empirical evidence.
Check-rule	Check with students and/or authors whether the feedback is suitable.

Issues	• Diagnosis is a hard issue in general. • Different types of feedback may be necessary. • DM needs to cope with the information underlying the feedback as well.

Requirement 7.4	Diagnosis of student's input in exercises.
Supports	Provision of elaborate feedback and tutorial dialogue.
Because	The correctness has to be judged.
Check-rule	Few tests comparing the system's diagnosis with teacher's diagnosis.
Issues	• Requires input from authors/teachers on frequent mistakes and possible solutions. • A correct diagnosis may be very difficult to compute. Therefore, it simplifies diagnosis, if the learner is asked to concretise her problem or to make more detailed steps. • Relevance of steps can only be judged approximately in some cases.

Requirement 7.5	A history of a student's activities in an exercise has to be stored.
Supports	Diagnosis and feedback.
Because	
Check-rule	Test which of this information is actually needed.
Issues	The local history has to be stored separately in order to be accessible to different components of the system.

7.2 Exercise Repository

Requirement 7.6	The repository needs to be extensible/updateable.
Supports	Evolution of content and asynchronous update.
Because	Proper versioning.
Check-rule	

Issues	• Management of change may be needed. • Simple approaches are candid and suffer several drawbacks such as the inappropriateness of text-based differences or the inability to search several versions.

Requirement 7.7	The repository is accessible and searchable by computational systems. In particular, it is well integrated with LeActiveMath.
Supports	Usability and interoperability of the system.
Because	Availability of the exercises to the TC and DM to integrate them into interactive courses and suggestions.
Check-rule	Check whether all exercises can be selected in tutorial choices og LeActiveMath.
Issues	• Exercises need to be encoded in the OMDoc exercise format with ActiveMath extensions (structure and metadata). • Need to display the copyright statement attached to each exercise. • Useful to have a single naming scheme for mathematical notions. • The repository should distinguish between a user who directly access and a user who access repository via LeActiveMath. • Service approach may be well-suited.

Requirement 7.8	The exercise subsystem communicates with the LM.
Supports	• Adaption of exercises and feedback in exercises. • Learner model update.
Because	Both directions of communication are needed.
Check-rule	Proof by existence.
Issues	Determine the relevant information on both sides.

Requirement 7.9	The repository is freely accessible and searchable by humans. This includes download exercises, upload exercises.

Supports	Proper usage of repository by teachers and self-directed learners.
Because	Repository is supposed to be open.
Check-rule	
Issues	Need to display the copyright statement attached to each exercise. We may adopt the multiple-choice-approach of creative commons.

7.3 Authoring Tools for Exercises

Requirement 7.10	Provision of template-based authoring tool.
Supports	Authoring of exercises.
Because	Interactive exercises will be difficult to author.
Check-rule	Check what the skills for authoring are needed.
Issues	Presumably, animations and interaction will have to be programmed anyway. Maybe in a more abstract language.

8 Content

The requirements on content are diverse. They range from pedagogically usefulness for moderate constructivism, the overall instructional design, over annotating instructional objects with metadata and providing a basis for adaptivity (a large range of types of exercises and examples, full range of competencies and presentations covered by instructional items) to the information about feedback and solution space for exercises. Partially these requirements are already included in other section.

Requirement 8.1	Sufficient content for demonstrating how several pedagogical/tutorial strategies work.
Supports	Adaptation to pedagogical strategies.
Because	Content is needed for realisation.
Check-rule	Run TC with such strategies.
Issues	

Requirement 8.2	Sufficient number of exercises to serve adaptive assessment.
Supports	Individual assessment.
Because	Adaptation needs many exercises to choose from.
Check-rule	
Issues	Some exercises could be generated rather than authored in detail.

Requirement 8.3	Sufficient and appropriate content (including references to student's reality) to show how moderate constructivist approach works.
Supports	Deep learning with moderate constructivist courses.
Because	Cognitive learning theories and evidences. System alone is insufficient.
Check-rule	Test courses.
Issues	

Requirement 8.4	Sufficient content to serve as a basis for proper classroom evaluation.
Supports	Evaluation.
Because	Proper classroom evaluation cannot be restricted to a small fraction of content since students have to work over a period with the system.
Check-rule	Test coverage of courses.
Issues	University and school content may differ.

Requirement 8.5	Sufficient variety of content to show how multimedia, interactivity and LeActiveMath components can be used in learning. This includes exercises with concept map tool, dictionary, computer algebra systems.
Supports	Construction of knowledge and motivation of students. Moreover it is needed for evaluation of the system and its proper usage.
Because	The possibilities of LeActiveMath have to be exploited including visualisation through multimedia content and interactivity.
Check-rule	Test the adapted courses. Count percentage of interaction in overall content.
Issues	• Multimedia content is more difficult to author, may be retrieved from Web but only with agreement of authors. • May need different levels for university vs. schools. • Coordination with TC needed.

Requirement 8.6	Sufficient variety of content to serve all (target) competencies (including change of representation).
Supports	Student's acquisition of different competencies and balanced courses.
Because	Adaptation for courses and scenarios targeting particular competencies.
Check-rule	Test resulting courses.

Issues	• Competencies have to be clearly defined based on pedagogic literature and experience. • Instructional objects have to be annotated.

Requirement 8.7	Provision of different types of mind mapping exercises: with some concept descriptors missing, with some or all relations missing, as well as free input (exploration).
Supports	Learn relations between just learned concepts.
Because	Conceptual interrelations get visualised.
Check-rule	Investigate student's understanding of domain concepts in pre and post tests.
Issues	• It may be difficult to create an appropriate pre & post test, because: how to decide, whether the learning increase is due to the the concept mapping or of another part of the system. • Drag&drop from LeActiveMath (Browser) to concept map tool needed.

Requirement 8.8	Exercise should provide (at least one) model solution.
Supports	Meet expectations of learner.
Because	If example is not deliberately erroneous, then a solution should exist.
Check-rule	
Issues	• How can authors be supported in encoding generalised solutions? • Automatic generation of solution paths for a large maths domain is extremely laborious and resource-intensive – too much for a project like LeActiveMath.

References

[1] J.M. Carroll and M.B. Rosson. Getting around the task-artifact cycle: How to make claims and design by scenario. *ACM Transaction on Information Systems*, 10:181–212, 1992.

[2] G.Jr. Chin, M.B. Rosson, and J.M. Carroll. Participatory analysis: Shared development requirements from scenarios. In S. Pemberton, editor, *Proceedings of CHI'97: Human Factors in Computing Systems*, pages 162–169, 1997.

[3] B. Cooper and P. Brna. Classroom conundrums: The use of a participant design methodology. *Educational Technology & Society*, 3(4):85–100, 2000.