working paper
final paper will appear in Springer Proceedings of ECAI Workshops

Guided Tours in ALeA

Assembling Tailored Educational Dialogues from Semantically Annotated Learning Objects

Jonas Betzendahl[1][0000−0001−6659−5308], Michael Kohlhase[1][0000−0002−9859−6337], and Dennis Müller[1][0000−0002−4482−4912]

Friedrich-Alexander University Erlangen-Nürnberg, Germany
Corresponding Author: jonas.betzendahl@fau.de

Abstract. In times of decreasingly homogeneous educational backgrounds and experiences and increasingly diverse educational target groups and circumstances, the need for educational content that caters to individuals and their specific situations rather than broad groups is rising.
We describe our approach to "guided tours", a framework of educational dialogues that are assembled and tailored on the fly to an individual learner's knowledge and educational experience as part of the intelligent tutoring system ALeA.

1 Introduction

The ALeA system (short for Adaptive Learning Assistant) is an intelligent tutoring system (ITS) that is primarily designed around adapting to the individual learner's knowledge level and preferences. It is an extension of the Mmt system [12] which supplies knowledge management functionality based on a domain model expressed as a Mmt/OMDoc theory graph [5].

One prominent feature of ALeA are educational dialogues which we call "guided tours". They are designed to allow any user of the system (university students, people studying for a certification, or merely curious souls, henceforth collectively called "learners") to have an interactive, dialogue-like learning experience that starts with *exactly* (ALeA's estimation of) their current understanding of the subject matter and ends with them understanding the concept they originally wanted to learn about.

The key distinction setting guided tours within ALeA apart from other educational dialogues in the context of online learning platforms, is that ALeA possesses a fine-grained (down to the concept level) understanding of any given learner's competencies, and also detailed information about available learning objects. This allows the system to cater to every individual's needs, preferences and current level of understanding.

Detailed information necessary for the above is harvested from *semantically annotated* (university) course materials. The annotations to existing course materials like lecture notes or slide decks introduce structural and topical information about the concepts being talked about in the material, which can then be used to offer a host of educational and didactic services, including guided tours.

Modelling Education We adopt the perspective of Berges et al. [2] that any good educator (human or not) relies on four models for successful instruction of an educatee, independently of concrete context and topic:

1. A **domain model** containing information about concepts in whatever domain of discourse we are interested in, and how they relate to each other.
2. A **learner model** that maintains an estimate of a learner's competencies in regards to the aforementioned concepts. This is achieved by paying attention to the learner and how they interact with the content or educator.
3. A **formulation model** consisting of "formulations" of this knowledge. This includes textual formulations (even in multiple languages) as well as video-files, audio-snippets, podcasts, stone carvings; even interpretive dance performances are thinkable. Anything that even loosely communicates the knowledge in question qualifies as a formulation.
4. A **didactic model** that organises these formulations and their relations to each other, as well as keeping track of their rhetoric classification (is this formulation an introductory video? A quiz question? A hint about a common mistake?) and didactic potential (how should the learner model change when the learner is given this question? When is it appropriate to recommend this musical piece?).

Structure We use the sTeX format [11, 7] for the annotation of the course materials. sTeX is a semantic extension of LaTeX, – realised as a LaTeX package – that provides macros and environments for embedding functional markup for knowledge items, learning objects, and their relations. sTeX sources can be transformed to semantically annotated HTML, from which MMT/OMDoc representations can be harvested for knowledge management purposes.

The union of all concepts introduced in these annotations and their specified relations to each other forms the ontology that the rest of the system operates on. This means there is no necessity to maintain (and ensure compliance with) a separate ontology. This allows for easy re-use of established concepts, but also (almost) equally easy specification of new concepts or alternative perspectives.

The domain model, as well as the formulation and didactic model, are in practice being managed, organised and served by the MMT system (which we will refer to as the "backend"). There exists an unconnected application managing the learner models (for reasons of separation of concerns and sensitive data) which we will call the LMS (Learner Model Server). Finally, the learners interact directly with a collection of web applications that connect to both the backend and the LMS, which we will call the "frontend".

We are currently working on a domain model for the canonical parts of the undergraduate computer science and AI curricula, and the mathematics they are based upon. The ALeA system is under continuous development and has already been deployed for six courses with over 1000 students total at FAU Erlangen-Nürnberg.

Contribution The ALeA system itself, its motivation, architecture and individual features, have been thoroughly described in other places in the literature (such as in [8]). Here, we want to focus entirely on the specifics of the tailored educational dialogues under the title “guided tours”. We explain the process of guided tours and point out how precise modelling of both learners and domain of discourse improves the experience.

Overview Section 2 goes further into detail about how individual learners are modelled in ALeA, and how this model can be interacted with and what it informs. In Section 3, we will explain the step-by-step flow of a guided tour, what information it draws on during each step, and how it is assembled. Finally, Section 4 concludes this paper and discusses ongoing and future work.

2 ALeA components

2.1 Learner Modelling

The ALeA system keeps a model of a given learner's mastery of concepts in the domain of discourse that they have interacted with before. This is represented as a set of triples $\langle C, D, p \rangle$, where C is a concept, D a cognitive dimension according to Anderson and Krathwohl's revision of Bloom's taxonomy for learning objectives [3, 1, 4], and $p \in [0, 1]$ a probability representing the learner's assumed competency in dimension D; with 0 indicating no previous contact with a concept, and a theoretical maximum of 1 indicating complete and utter mastery.

Specifically, the cognitive dimensions we use are `remember`, `understand`, `apply`, `analyze`, `evaluate` and `create`. Not all cognitive dimensions necessarily apply equally to all concepts (it would be difficult, for example, to assign a meaningful score for `create` for the Intermediate Value Theorem, even though other dimensions remain useful).

Priming of Learner Models When learners log into ALeA for the first time, the system knows little about them. In particular, there can be no such thing as a “default” learner model. It might be tempting to install one and assume that every user of the system has a certain educational background (such as arithmetic and trigonometry), but this would paint all learners with the same brush and work antithetically to the goal of providing individualised learning opportunities.

As an alternative, we use an on-boarding dialogue that explains the individual parts of the system to newcomers, but also offers them to *prime* their learner model with their educational history. We give a list of courses and programs the learner might already have completed (such as “Introduction to Databases”, but this could also include “Bachelor in Art History at XYZ University” or “German Abitur in Berlin”) and ask their grade as a percentage. We then perform a lookup for which concepts of our domain model are being discussed in the course or program in question and set their learner model correspondingly. This allows learners to start with a reasonably accurate and therefore genuinely helpful learner model (e.g. one that reflects a strength in theoretical computer science or a weakness in differential equations, instead of a passing familiarity with both) without having to amass hundreds of interactions first.

Learner Model Fine Tuning Exactly how a given interaction should affect the learner model is subject to ongoing research. For example, it might seem clear that giving a correct answer to a quiz question involving concepts C_1 and C_2 should increase their values in the learner model. However, it is not obvious by how much. Should they be raised to a set point? Increased by a percentage? If C_1 used to be higher than C_2, should that affect it? Should we also update concepts that one or both depend on, and if so, by how much? What do we do about modelling "forgetting"?

We hope to collect enough data to eventually be able to give evidence-informed opinions on this, by having multiple variations of learner models be informed by the same interaction logs collected over time, and see which predicts the exam grade the best. For now, we rely on an iterative, experimental approach that uses the "quality of semantic services" – e.g. of the generated guided tours as a dependent variable to be optimised.

However useful we might find the data for our own scientific purposes, first and foremost we want learners to be in control of their educational journey (for didactic as well as ethical and legal reasons) and therefore provide them with a way to inspect and change their learner model at any time. They can also initiate a "purge" that erases every bit of information the system stores about them (with the sole exception of the time that such a purge took place), completely resetting their learner model and deleting all interaction logs.

2.2 Learning Objects

In the following, we will use the phrase *learning object* for any formulation that has been semantically annotated with didactic/rhetoric information, all in their respective senses given above. It should be noted, that this differs from how parts of the didactic literature use the term, where it can also mean *"anything that is potentially of use during learning"*.

The information about a learning object L that can be queried in the system will serve as the basis for decisions about the shape and content of the guided tour. It includes:

1. a **string identifier** for L,
2. a **rhetoric classification** for L (definition, example, theorem, problem, ...),
3. the **prerequisites**, $P(L)$, that is to say a collection of pairs (C, D) where C is a concept and D is one of the cognitive dimensions, specifying what parts of the learner model L specifically relies on so that it can be attempted by the learner with reasonable chances of success,
4. the **objectives**, $O(L)$, also a collection of pairs (C, D) as above, but representing the skills displayed by completing the learning object[1],
5. a learner model $M(C, D) \in [0, 1]$ as discussed in Section 2.1,
6. a cool-down value $\mathfrak{CD}(L, M)$, that is to say an integer value associated with L per learner.

[1] These may well be different from the prerequisites of L and additionally can be further specified for individual answer classes, as discussed in [9].

3 Guided Tours

We will now explain the step-by-step flow of a guided tour, what information it draws on during each step and how it is assembled. Please also refer to Figure 1 where you can find a graphical representation of the process.

As a running example, we will show possible dialogue snippets that a learner asking for a guided tour about the Pythagorean Theorem might encounter[2]. We have chosen this example for general familiarity, its simplicity does not restrict the scope of the process. Learner responses in a dialogue like this would be chosen from a list of possible options, not entered as free-form text.

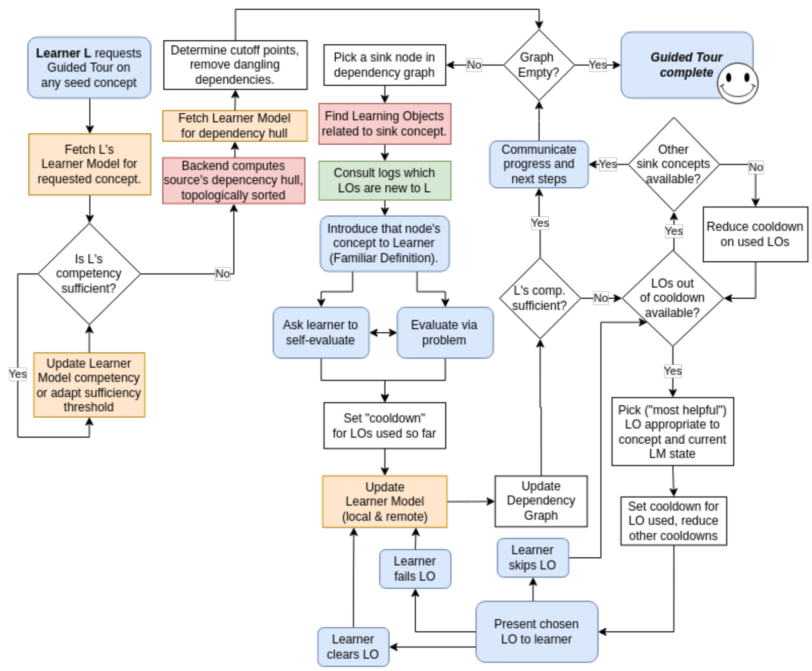


Fig. 1. Overview of the Guided Tours Algorithm. Colour indicates involvement of certain ALeA components. Blue: Frontend, Yellow: Backend, Red: Learner Model Server, Green: Interaction Logs

3.1 Initial Impetus

Guided tours (GT) can be initiated from any point within the ALeA system – and the respective context serves as an important parameter to GT generation. It could be during a lecture video, from a section in lecture notes, or possibly even from within a forum that students, tutors and professors all frequent.

[2] A more complete version of the prototype dialogue is at https://courses.voll-ki.fau.de/exp/pp_dialogue_tour.

Generally: any place where a learner can interact with a concept, it is possible for them to start a guided tour if they feel like they would benefit from understanding more about it. As soon as the learner opens the guided tour (in our case for the Pythagorean Theorem), they are taken to the dialogue setting.

> Hello!
> You want to learn more about the Pythagorean Theorem?
> Great! Let's get started!

Although it is an important aspect of the process, we will not discuss here what constitutes a "sufficient" level of knowledge for a given concept. The system is parametric in this regard and can adapt to this threshold changing shape or moving for various reasons. A hypothetical learner may, for example, be content with just remembering the definition of a concept as a working baseline during the semester, but insist on also acquiring a good score in other cognitive dimensions before the exam.

However, we do want to highlight one special case: it is of course possible that when a learner starts a GT on a certain subject, that their learner model already fulfils whatever standard we set for sufficient knowledge. In this case, we do not immediately terminate the tour, as might seem to be the obvious step to take, but instead we will either raise/change the standard or downgrade the learner's model for this concept. It is our understanding that the very act of requesting a GT on a concept communicates that the learner is not happy with their current understanding and wants to learn more about it.

3.2 Navigating Dependencies

Once a learner initiates a GT for a certain concept they want to learn about (which we will call the **goal concept**), we query the backend for the complete dependency hull of that goal concept, that is to say the transitive closure of the dependency relation[3]. This graph of concepts will serve as our information what concepts we still need to present to the learner. We also query the LMS for the current learner model for the set of concepts in the dependency hull and associate the values to their respective concepts.

To keep learners engaged in an educational dialogue, it is important that they believe they are not being asked to re-learn things they already understand well. However, in the (usually quite sizeable) dependency hull of the goal concept, there are bound to be concepts a given individual learner is already familiar with (such as "points", "lines" or "triangles"). To make sure not to re-present them to learners unless necessary, the next step involves so-called cut-off points.

For our purposes in guided tours, we use the word *cut-off point* to refer to those concepts in the dependency hull of the goal concept that the learner

[3] We use "dependency" here to mean that when the definition of a concept C_1 uses concept C_2, then C_1 *depends on* C_2. Note that the dependency relation is a strict partial ordering and thus acyclic.

already has a sufficient understanding of, meaning we don't need to introduce them or any of their dependencies during the course of the GT.

Note that this does *not* require that the learner's understanding of all dependencies of the cut-off point is sufficient. For example, to explain the Pythagorean Theorem, the learner needs to know about numbers. Those in turn could potentially be defined via (and therefore depend on) some concepts in set theory, but the set-theoretic aspects are of no consequence to the process of learning about the Pythagorean Theorem. So we deliberately make the choice not to require that the learner understands the complete dependency hull of the concept they are interested in. Should it later turn out that they do in fact need a better understanding of a certain concept that was behind a cut-off point and therefore not included in the original GT, they can indicate this at a later point and it will be incorporated in the tour then.

Before moving to the next step in the process, we remove all "dangling dependencies" from our graph, that is concepts that cannot be reached from the goal concept without passing at least one cut-off point.

3.3 Introducing Concepts

We are now ready to introduce the learner to a concept proper. For this, we can pick any of the "sink" nodes (i.e. nodes that other nodes may depend on but do not depend on other nodes themselves) in the dependency graph.

Once we have identified the first concept to talk about, we query the backend for all learning objects that teach something about it, and cross-reference them with the logs to see which of them the learner has interacted with in the past. We are especially interested here in which definition is familiar to the learner, as there can be multiple definitions for the same concept, and we want to start from a place of familiarity.

Next is the presentation and introduction of the concept to the learner. This takes the form of either giving the learner the familiar definition to self-evaluate on or a problem which the learner can answer as a proxy instead. Keep in mind that ALeA is (and therefore guided tours are) designed to work in tandem with traditional learning and studying methods, not to replace them. A student interacting with ALeA in the afternoon could have spent their morning practising a concept by themselves, or maybe weeks have passed without a thought given to this subject, forgetting almost everything. This would both not be reflected (yet) in their learner model, so this part of the interaction tries to catch up to the current state, akin to a human teacher or tutor gauging where exactly to start on a certain topic with their pupil.

My records show that you are already familiar with angles and triangles. The Pythagorean Theorem also concerns *right triangles*.

Do you already feel comfortable with that topic?

I'm not sure...

Okay. Let's find out! Please solve the following problem:

In a right triangle, one of the angles at the longest side is $60°$. What would that make the other angle on the longest side?

$30°$

That is correct!
Let's talk about the Pythagorean Theorem then.

At this point, we're also setting a "cool-down" $\mathfrak{CD}(L, M)$ for the learning objects used in the tour so far (or recently in other contexts, as informed by the interaction logs). This takes the form of an integer associated with both the learner and the learning object. This cool-down will be decreasing over time and is meant to help the system not to show the same learning object multiple times in close succession. This precaution is being taken to avoid frustration. Being presented the same question again that you couldn't answer a minute ago could be perceived as tone-deaf or even mocking by the system or at the very least supremely unhelpful otherwise.

Following the introduction of the concept, the learner has already interacted with one or more learning objects (such as the definition we showed them or the question(s) they self-evaluated with), either successfully or not (more on this below). Both outcomes necessitate an update of the learner model, so we sent the relevant updates to the LMS.

3.4 Working with Concepts

Necessary Updates After the learner model has been updated based on the newest interaction, we also query the LMS for an updated version of the learner's model for our dependency graph, which we will now have to re-evaluate. Firstly, because we need the updated values to proceed and do not want to re-create the functionality of the LMS within GTs, and secondly because, as discussed in section 2.1, ALeA does not prescribe one true way of updating learner models. Hence, the update we just queued might have affected other parts of the dependency graph. For example, it is possible that cut-off points have moved or that concepts we originally judged to not be sufficiently known now are and therefore don't need to be talked about during the GT any more.

When both learner model and dependency graph have been brought up to date with the newest information, we evaluate if the learner has now reached the "sufficient" level of competency with the concept that is currently in focus. If they have, there does not need to be any further discussion of this concept and we can move on to the next one (see section 3.5).

Choosing Learning Objects Should the learner's competency not meet our sufficiency standard, we need to select a learning object to improve it.

We originally queried the backend for all learning objects related to the current concept. The results of this query include not only identifiers for the learning objects, but also meta-information about how interacting with it might affect a learner model or which situations or classes of learners it is especially suitable for (see Section 2.2). This information can now be used to help identify the appropriate learning object for the current situation.

First, we make sure that there are learning objects available at all that are "out of cool-down". If there are indeed, we can pick the one out of the collection that is the most helpful given the current state of the learner model. There are many approaches to deciding which learning object is the "most helpful", since it strongly correlates with the learning standard we have set. A video detailing how to create your own Diophantine equations is helpful if that is actually what you are currently trying to learn. There might be circumstances though where a learner is happy with just remembering and understanding the definition, and in that situation, a poem that serves as a mnemonic for the definition would be better suited to the task. A "default" way of selecting a fitting learning object does arise, however, from the available information (see Section 2.2). Out of the suitable learning objects that involve the current concept and are out of cool-down, select those for which the fulfils the prerequisites $P(L)$. Out of those, filter for the ones with objectives $O(L)$ that best address the lowest scores in cognitive dimensions. If this still returns multiple learning objects, chose randomly.

Of course, theoretically, the judgement about what learning object is most appropriate can become arbitrary complex, all depending on the amount of data and computing resources available. Has the learner ever seen this learning object before? How have other learners with similar competencies reacted to this learning object? Who created this learning object and when and in what context? We hope to give this question of how to pick an appropriate learning object for a given situation the detailed treatment it deserves in the future.

Theorem (Pythagoras).
In a right triangle, the length of the hypotenuse is equal to the sum of the length of the other two sides.
Often, this is expressed with this formula: $a^2 + b^2 = c^2$

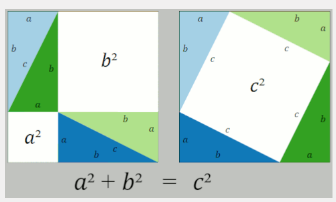
Should there *not* be any learning objects available that are out of cool-down, the GT can continue in two ways. If there are other sink concepts to talk about in our dependency graph apart from the current one, we switch. If we have both no learning objects out of cool-down and no other concepts to talk about instead, we accept our fate and decrease all current cool-downs by one level and check again. Maintainers of learning object corpora should make sure that this situation occurs rarely or never.

Interaction with Learning Objects Once a suitable learning object has been selected, we give it a cool-down value and decrease all other cool-downs by one level. The learning object we picked can now be presented to the learner. At this point, we distinguish three different ways to interact with the learning object:

1. **(Pass)** This is a positive interaction, indicating that one has better understood a concept. This should only increase a learner's modelled competency.
2. **(Fail)** This is a negative interaction, indicating that the learner has not mastered this concept yet. Hence the modelled competency would be reduced.
3. **(Skip)** This is a neutral interaction, merely indicating that the learner didn't fully interact with this learning object (no update to the learner model).

In our running example, one way of such an interaction would be the learner indicating that they do understand the material that was presented to them [4].

There are multiple proofs of this theorem.
Here is one of them:



Did that help you understand?

Yes

3.5 Da Capo al Coda

The last step in the cycle is communicating to the learner how the past interactions have shaped their learner model (this can include complimenting them or explaining that it might be better to move to a different topic right now), and moving the conversation to the topic of the next steps, so that the learner remains able to follow the dialogue well.

Having gone through the process of communicating about a concept with the learner once, we now return to our dependency graph of the original goal concept. If it is already empty, we have finished the GT, but in reality this will likely take several rounds of the process above, leaving us with the task of picking a new sink concept from the graph. The selection of a sink concept here could be random, since (to our knowledge) the learner already knows enough about their dependencies. However, we believe that the next concept should be picked by similarity to the last one,preferring concepts that are used or introduced close to each other in the relevant course context. The necessary information already exists in the system and can be queried from the backend, in the hopes of a more coherent narrative in the mind of the learner and avoiding "topic whiplash".

[4] In the interactive version of the dialogue, this diagram is actually an animation (licensed CC-BY, attribution William B. Faulk), which is easier to understand.

From here on out, the process detailed in Sections 3.3 and 3.4 repeats, until we have reached an empty dependency graph (the last concept having been the goal concept), at which point the guided tour is complete and we can congratulate the learner for their success. They can then either click away or follow a link to start one of the *other* GTs that are being suggested based off their current learner model…

4 Conclusion and Future Work

In this paper, we introduced and justified the concept of a guided tour, an educational dialogue in the context of the intelligent tutoring system ALeA, that is assembled on the fly and tailor-made specifically for an individual learner.

Precise modelling of the domain of discourse as well as learner's abilities allows us to make informed choices about which topics to discuss and which learning objects to present. This is all to the benefit of the learner, who gets to avoid tedious and unhelpful experiences in online learning, resulting in an educational dialogue that more closely resembles that of a human educator. We hope that this will lead to an increase in successful learning journeys, even in the face of ever more severe budget constraints in public education and ever more heterogeneous educational backgrounds.

Future Work Continuing work on guided tours, we hope to be able to increase our accuracy in measuring and modelling learner progress. One avenue is introducing another dimension of "certainty" to the learner model, that would allow to differentiate between scores in which we have high or low confidence, say, based on number of interactions involving some concept. This might enable us to skip certain parts of GTs where we gauge the learner's current level, or it might improve the quality of learning object selection.

We also hope to expand the corpus of annotated materials, allowing guided tours (in practice, not just in theory) to not be constrained to only one university course at a time, but be able to draw on connections to other courses and programs that learners have already completed (e.g. "This neural net has a similar structure to the one you know from last semester's course on Computer Vision, with the difference being…").

Acknowledgments

The work reported in this article was conducted as part of the VoLL-KI project (see https://voll-ki.de) funded by the German Research/Education Ministry under grant 16DHBKI089. The work reported here has profited significantly from discussion in the VoLL-KI group.

The design of the ALeA system inherits a lot of the intuitions from the ActiveMath/LeActiveMath system [10], which was based on an early version of OMDoc [6]. The concept of a guided tour discussed in this paper is informed by the realisation – initially suggested by the second author – in ActiveMath, but the representational basis and eventual shape are completely distinct.

References

[1] L. W. Anderson and D. R. Krathwohl. *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives.* New York: Longman, 2009.

[2] M. Berges, J. Betzendahl, A. Chugh, M. Kohlhase, D. Lohr, and D. Müller. "Learning Support Systems based on Mathematical Knowledge Management". In: *Intelligent Computer Mathematics (CICM) 2023.* LNAI. in press. Springer, 2023. URL: https://url.mathhub.info/CICM23ALEA.

[3] B. S. Bloom, ed. *Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain.* New York, Toronto: Longmans, Green, 1956.

[4] U. Fuller et al. "Developing a computer science-specific learning taxonomy". In: *SIGCSE Bulletin* 39.4 (2007), pp. 152–170.

[5] M. Kohlhase. "Mathematical Knowledge Management: Transcending the One-Brain-Barrier with Theory Graphs". In: *EMS Newsletter* (June 2014), pp. 22–27. URL: https://kwarc.info/people/mkohlhase/papers/ems13.pdf.

[6] M. Kohlhase. *OMDoc – An open markup format for mathematical documents [Version 1.2].* LNAI 4180. Springer Verlag, Aug. 2006. URL: http://omdoc.org/pubs/omdoc1.2.pdf.

[7] M. Kohlhase and D. Müller. *The sTeX3 Package Collection.* URL: https://github.com/slatex/sTeX/blob/main/doc/stex-doc.pdf (visited on 04/24/2022).

[8] T. Kruse, M. Berges, J. Betzendahl, M. Kohlhase, D. Lohr, and D. Müller. "Learning with ALeA: Tailored experiences through annotated course material". In: Lecture Notes in Informatics. in press. Gesellschaft für Informatik e.V. (GI), 2023.

[9] D. Lohr, M. Berges, M. Kohlhase, and F. Rabe. "The Potential of Answer Classes in Large-scale Written Computer-Science Exams". In: *Proceedings of the 10th Symposium on Computer Science in Higher Education HDI23.* Accepted. Aachen, Germany, 2023.

[10] E. Melis, J. Buedenbender, E. Andres, A. Frischauf, G. Goguadze, P. Libbrecht, M. Pollet, and C. Ullrich. "The ActiveMath Learning Environment". In: *Artificial Intelligence and Education* 12.4 (2001).

[11] D. Müller and M. Kohlhase. "sTeX3 – A LaTeX-based Ecosystem for Semantic/Active Mathematical Documents". In: *TUGboat* 43.2 (2022). Ed. by K. Berry, pp. 197–201. URL: https://kwarc.info/people/dmueller/pubs/tug22.pdf.

[12] *UniFormal/MMT – The MMT Language and System.* URL: https://github.com/UniFormal/MMT (visited on 07/20/2023).