Using Augmented Reality in Software Engineering Education?

First insights to a comparative study of 2D and AR UML modeling

Rebecca Reuter
OTH Regensburg
Germany
rebecca.reuter@oth-regensburg.de

Florian Hauser
OTH Regensburg
Germany
florian.hauser@oth-regensburg.de

Daniel Muckelbauer
OTH Regensburg
Germany
daniel3.muckelbauer@oth-regensburg.de

Theresa Stark
Regensburg University
Germany
theresa.stark@stud.ur.de

Erika Antoni
OTH Regensburg
Germany
erika.antoni@oth-regensburg.de

Jürgen Mottok
OTH Regensburg
Germany
juergen.mottok@oth-regensburg.de

Christian Wolff
Regensburg University
Germany
christian.wolff@ur.de

Abstract

Although there has been much speculation about the potential of Augmented Reality (AR) in teaching for learning material, there is a significant lack of empirical proof about its effectiveness and implementation in higher education. We describe a software to integrate AR using the Microsoft Hololens into UML (Unified Modeling Language) teaching. Its user interface is laid out to overcome problems of existing software. We discuss the design of the tool and report a first evaluation study. The study is based upon effectiveness as a metric for students performance and components of motivation. The study was designed as control group experiment with two groups. The experimental group had to solve tasks with the help of the AR modeling tool and the control group used a classic PC software. We identified tendencies that participants of the experimental group showed more motivation than the control group. Both groups performed equally well.

1. Introduction

Computer science has shown to be a field of study provoking mixed feelings with students. This is mainly on account to the fact that the curriculum of this subject primarily consists of abstract concepts, which can be hard to grasp at first sight. This is especially true for the domain of software engineering: While being of major importance for small to large-scale software projects, commercial and FOSS (Free and Open Source Software) alike, it appears that only few students can relate to these (teaching) contents. We firmly believe that this is due to the method of teaching, rather than the contents of the subject, which makes this topic especially attractive to our research.

Successful learning—and motivation to do so—of course, is always linked to the ability of students to experiment with the learned material firsthand. This requires software used in courses to be easily accessible for learners: The software must be "forgiving" regarding errors in usage: Students should not be frustrated too easily, when clicking on wrong buttons or using the software in the wrong way. Overly complicated software hinders the students' progress through tasks or assignments. In addition, the software should be usable with as little previous knowledge as possible. Summing up, expressive and usable software with a learning curve that is not too steep is favorable.

Within the scope of our work, we focus on teaching UML. One can imagine an ideal software for working with UML in classroom settings as one which is as easily usable as a whiteboard while at the same time allowing the students the expressive power of drawing or text editing programs, where they can copy or shift around parts of the diagrams at ease.

According to [1], the big advantage of 2D drawings is that they only require paper and pencil. However, designing UML using paper and pencil only is a cumbersome chore, which can be greatly simplified using software: Elements in the model can be easily resized, moved around, edited and removed. Smart layout mechanisms may additionally support the user in creating readable diagrams. There is, of course, already existing software implementing these features.

URI: https://hdl.handle.net/10125/60218
ISBN: 978-0-9981331-2-6

However, there are some arguments in favor of extending UML models to the third dimension, which is AR in our case [2]:

- It is possible to layout objects in 3D more consistently than in a single layer [3].
- In a 3D environment, order in space can be expressed and understood much better than in 2D.
- 3D models allow semantically richer visualization than 2D models [4] in terms of perception of 2D, 3D reduces cognitive overload and improves user perception.
- In the 2D view, only one diagram can be displayed at a time, while in the 3D view, several diagrams can be displayed in a single scene.
- Three-dimensional views may be important for understanding the relationships between different models or diagrams [5].
- According to [3] 3D objects have a visual appeal and can be more attractive for students and thus more beneficial for the learning process.

We introduce a first step to setting up such a 3D modeling environment: Therefore, we developed an augmented UML editor.

We believe that by working with UML charts as with real-life objects, these in return become more graspable for students as well. Students are enabled to shift around entities by physical actions, thus they seemingly exist in the real world, rather than being abstract colored fields on a computer screen. We are therefore interested in new ways to teach and learn abstract concepts, by treating them as experienceable real life objects.

We started this work based on the assumption that various learning obstacles exist in the subject of software engineering. In our previous research, we found that there are in general six dimensions of learning obstacles [6]. Within the scope of this paper we focus on didactic learning obstacles with focus on UML modeling: Keng and Siau [7] found in 2006 that students struggle when learning UML because of a lack in adequate teaching material and because of an overload in existing software. This leads us to assume that better software is necessary to motivate students and to further their learning success. This is true for teaching in general but especially for highly software-intensive topics like UML modeling. Instead of just learning how to use software students should be enabled how to actively express the learned material using the software. We do not envision to provide industrial-grade software solutions for teaching, but rather a software prototype suited for application in classroom settings for evaluation. To this end we also want to integrate AR technologies since they have been shown to be beneficial regarding motivational aspects e.g. [8] and learning success e.g. [9].

Beside the motivational aspect to be a successful learner, we tried to design a software that overcomes existing problems when learning UML as described in the background section. To this end, we want didactical learning obstacles (e.g. learning material) to be reduced. Therefore, we have developed two software tools, one to use as a classic 2D PC software, one in AR. As a direct comparison of the two prototypes should be possible, we ensured that both software solutions offer almost the same functionality.

This paper includes the following contributions: An empirical survey using a control group experiment setting. The study investigated whether and to what extent the use of augmented reality in modeling with UML is suitable for students of software engineering. It measures the success and gathers feedback from the students, documenting their point of view. We focused on the variables *motivation* and *learning* for this survey.

A tool for the integration of AR in software engineering teaching: The software makes it possible to use both, PC-based 2D editing and Augmented Reality in classroom settings and allows for seamless exchanges between one and another. This will be used for more extensive experimental settings in the future.

The remainder of the paper is structured as follows: We first provide a background section to clarify our motivation for this research. Afterwards, we give a short overview of existing approaches for visualizing UML in the related work section. Then we present the approach and describe the software tools. Furthermore, we introduce our used methodology and the evaluation of the system. Finally, we point out threats to validity and limitations of our work and present further considerations for future research.

2. Background

In this section we describe the necessary information to understand our research interests. Learning obstacles, especially didactic learning obstacles in learning UML are the conceptual basis for our research.

2.1. Learning Obstacles

We define learning obstacles as follows:

“A learning obstacle can be at least assigned to one of the five dimensions—namely emotional, epistemological, didactical, resource-related, and metacognitive—and represents an obstacle that prevents a learner in any manner from learning."[6]

We have derived the five learning obstacle dimensions in a previous work [6] from *Cognitive Load Theory* by Sweller [10]–[12] as well as the *Learning Strategy Classification* by Weinstein and Mayer [13]. Additionally, we referenced and mapped them to the *Learning Dimensions* to secure the coverage of all aspects in learning.

These obstacles are namely:

1. **Emotional & Motivational Learning Obstacles** are concerned with motivational and emotional aspects, which covers the internal learner's attitude.
2. **Epistemological/Cognitive Learning Obstacles** describe a misjudgement of the learning object and/or the individual competencies related to the learning object.
3. **Didactical Learning Obstacles** describe external interferences regarding structure, setting, and type of material for a course.
4. **Resource-Related Learning Obstacles** are the most complex ones, as they are concerned with the internal resources (effort and time management) as well as the external (information gathering, cooperation, and environment).
5. **Metacognitive Learning Obstacles** deal with self-controlling.

Therefore, a learning obstacle can be multidimensional; e.g., if the learner cannot find the correct information to a problem, he/she might get stressed, i.e. this obstacle might have a resource-related as well as an emotional aspect.

2.2. Problems in Teaching and Learning UML

UML is the current *de facto* as well as *de jure* standard (ISO/IEC 19505:2012) notion for visualizing models in software design. This emphasizes its importance as part of the curriculum when learning software engineering. In order to be able to design adequate software to teach software design with UML and to achieve successful learning for students it is necessary to understand and alleviate difficulties in learning UML for novices. In [7] the concept mapping technique was used to develop categories of difficulties encountered by the subjects: "The participants in this study were students who had completed the Object-Oriented System Analysis and Design (OOSAD) class at a large Midwestern (U.S.) university [and most of them had no prior knowledge before]. The course focuses on introducing the concepts, syntax, semantics, and diagramming techniques of UML; it also covers object-oriented concepts [7, p. 44]." The study was conducted in six steps, as specified by Trochim [14], which were carried out in two phases: Phase one consists of "Prepare Project" and "Generate Statements", phase two is called "Computing and Utilizing Concept Maps" and starts with step three "Structure Statements". Step four is called compute maps, then the interpretation follows before step six "utilize maps" is proceeded[7].

They derived 15 clusters and categorized them in five meta-regions, which represent the major difficulties perceived by the students: Training Material, Prior Knowledge, UML Diagrams, UML Semantics and UML Constructs [7].

In a further step, they consolidated the meta regions into inherent and peripheral categories. Category one includes issues related to UML diagrams, UML semantics and UML constructs. Category two consolidates peripheral issues in learning UML which are issues related to training material/software and absence/presence of prior knowledge [7].

Forty-nine students enrolled in the object-oriented systems analysis and design (OOSAD) course were recruited to participate in phase 1 of the study. For phase 2, another 30 students who had taken the course were recruited [7].

In our study we relate to this second category concerning the training material. Considering the learning obstacle dimensions we try to reduce existing didactical learning obstacles that may lead to motivational or epistemological obstacles.

3. Related Work

We based our search on two recent literature reviews on the application of AR/VR technology in the general educational context [15], [16]. Based on these reviews, we found some interesting information: First of all, it is striking that since 2013 the number of publications containing augmented reality and its use in teaching has been increasing steadily. Due to the increase in popularity and common availability of sufficient AR/VR technology in recent years, this was to be expected. Second, only a small share of this body of research considers empirical data. Finally, the teaching of computer science in general, and software engineering in particular, is still hardly represented in this research.

Regarding the usage of UML in combination with Augmented Reality McIntosh and Hamilton proposed an approach for visual debugging for Lego NXT using UML Mechatronic Diagrams in 2010 [17]. In this instance, the use of 3D allows UML diagrams (here: state machines) to be associated with hardware objects moving in a 3D space. In our approach, we focused on an interaction of the students/participants with the tool, which is not possible here.

In [2] a VisAr3D environment is presented that has been developed for classroom usage to provide a 3D visualization of UML models. The 3D visualizations of the UML models were presented in a virtual reality setting. It is a 3D viewing environment and therefore it does not allow editing of models in 3D. Its target audience is the teacher and graduate students. As we want to implement a new way of modeling, interaction with the tool has to be possible. This publication also relates to the above named one [17], which shows that there is a lack in research.

Although these papers deal with 3D visualizations, they do not always refer exclusively to augmented reality. None of the publications considers validated pedagogic instruments for measuring e.g. motivation. We try to base our approach on a validated questionnaire and derive our self-assessment questionnaire based on learning goals. This is also not clearly defined in the related publications.

4. Modeling UML Diagrams using Augmented Reality vs. Modeling UML Diagrams in Ariadne

Ariadne, which is an eponym for the Ariadne thread through the software development process, is a self-developed software integrating both, 2D and augmented reality UML modeling.

In the design phase, students are able to use both, the 2D UML editor and the AR UML editor, both providing them with the same tool suite. The idea is to reuse all the existing and relevant information related to the modeling elements of a system. These important data will be associated to the corresponding modeling elements and the user should be able to explore and interact with them in the third dimension.

4.1. System Overview

Ariadne consists of two modules: The 2D environment and the AR environment. The desktop and AR software versions both share an interface for accessing data and results in a common database. Thus, students can work using both, the desktop version and the AR version of the software and seamlessly migrate between them.

The 2D environment is a *windows presentation foundation* (WPF) application based on the *NClass* [18] UML editor. It consists of editors for capturing requirements and use cases in text form, a modeling environment, and a class editor.

The AR environment is developed with the *Unity* game engine for the *Microsoft Hololens* AR glasses.

With Ariadne, students should be confronted with the necessary complexity, but the software only offers the necessary functions. This reduction in functionality allows the students to explicitly focus on their task.

Software development itself is rather complex and includes many tasks, e.g. requirements analysis, modeling the software design, testing, to just name a few. Within this paper we focus on the modeling aspect by means of the UML editor:

With increasing complexity of the software design, the UML models increase in size. These models come with their own share of problems: Crossover-free relationships between entities in the model are desirable, but not always possible. Given a larger number of classes, the user/student has to zoom out to see all classes at the same time, this is for example not necessary in the AR environment due to the possibility to move around which might be seen as a more natural type of interaction / navigation.

Summing up, we would like to find out whether and to what extent the learning content of software engineering courses can be conveyed better by using an AR interface. "Better" in this case refers to comparing students' motivation and/or performance for the setting.

4.2. Interaction Design

This section covers the interaction design of the AR UML modeling software. While the Ariadne 2D user interface is similar to known UML editors, such as *Astah* or *Enterprise Architect* (with a limited range of function), control concepts for desktop computers do not trivially extend to the domain of AR user interfaces.

As we have described in ch. 1, one major issue in learning UML stems from the complexity of creating and editing UML in common UML editor software. Therefore, we aimed at simplifying the user interface as far as possible, by keeping the tooling as expressive, as possible. This was also mentioned as a problem in learning UML[7].

In AR, there is no such thing as a mouse one can use for pointing or for performing different actions (left and right clicking, double clicking etc.). Rather, these actions are realized in the AR editor as gestures, the only remaining interface buttons are the save and reset buttons, since these are not context sensitive: One may not save a single class or relation, but rather the diagram as a whole. An example for the gestures in the system is shown in figure 1. The student sees an augmented part of reality through the *Microsoft Hololens* (that is, he sees the room and the UML classes floating in the room). Using the gestures, he may change properties of the diagram.

We decided to provide a stripped-down interface in the AR editor integrating only a reset and save button. This was done in order to provide as much modeling area as possible and to reduce this complexity by additional buttons. The remaining functionality was realized via gestures and drop-down menus during modeling.

The user initially sees only an empty area, a reset and save button. Modeling takes place using gestures. For new users, a tutorial and a cheat sheet are provided in advance so that they can familiarize themselves with the controls. In contrast to virtual reality, the user sees his real environment, which is why he can also use the cheat sheet during modeling.

4.3. Prototype Implementation

The prototype of the 2D editor works as one would expect a UML class diagram editor to work: There is a toolbox with the given elements (classes, interfaces, inheritance and association relationships). The user can create new elements by a click to a given element icon and by a second click to the desired position in the modeling area.

To create a connection between two classes the user must first choose the desired type of association. Then he/she has to click on the source class and next click on the destination class.

Diagrams can be stored to the database via a click to the save button.

By double clicking classes or interfaces, new attributes and methods can be added. This feature is also available by doing a right click to these elements.

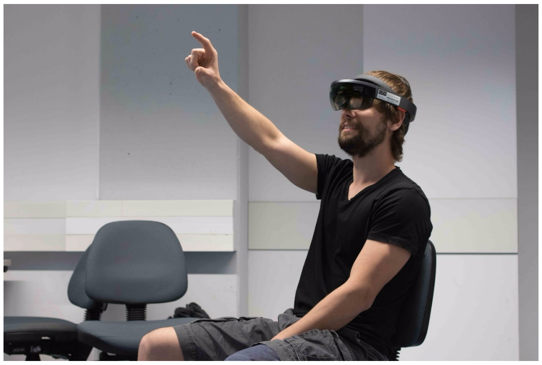

Figure 1. Demonstration of a user performing gestures while modeling a class diagram

The other basic functionality such as creating a new class, is realized via different gestures. Where multiple options are possible (e.g. setting whether a relation between two classes should be a specialization or an association). Possible options are presented in a dropdown menu. The user may then select the desired option from this menu. We chose to realize these actions via a dropdown menu, since this massively reduces the number of gestures a user must perform to execute the desired action. In figure 2, a dropdown menu is shown in action. The user may assign the visibility of certain features of a abstract class. Take note that this screenshot is taken from the *Unity* environment rather than from the real system in action. Therefore, the background is rendered in a blueish gradient.

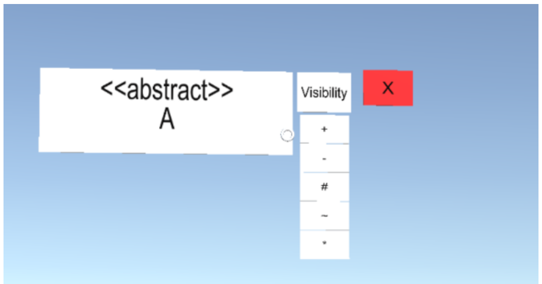

Figure 2. Example for a dropdown menu to assign the visibility of attributes in a abstract class

Dropdown menus are, however, a concept from the 2D world, and we are currently not certain whether this feature extends properly into the 3D world. We plan to evaluate its suitability in further experiments.

5. Methodology

In this section, we describe the methodology used in our empirical study. We start with general aspects and explain the detailed components in the subsections. Since the software is already designed for two variants PC (2D) and Hololens (augmented reality), we decided to use a control group-based experiment which allows for a differential analysis of the two software solutions tested.

Description of the target group: The target group were participants of a software engineering course. They were undergraduate students with a non computer science major course. Please note that this subject is in the 5th semester and the students are studying electrical engineering and information technology (EEIT) or mechatronics. The students were novices with respect to UML modeling. They only had one teaching unit before this course on class diagram modeling. This lecture unit takes place during a course on object-oriented programming in the second semester.

Description of the setting: The module consists of a lecture and an auxiliary exercise. The experiment was fully integrated into the curriculum, i.e. students listened to the lecture on UML class diagram modeling and a week later entered the associated exercise where the experiment took place.

Goal: The aim of this experiment was to find out whether the experimental group of the experiment *performed more successful, performed equally, was more motivated, was similarly motivated, was more interested in the content, was similarly interested in the content, performed more successfully in the next tasks (UML II)...* than the control group.

Procedure: To get an impression of how the participants assess themselves with UML class modeling after the lecture and to ask for possible previous knowledge, we carried out a self-assessment using a questionnaire directly after the lecture— i.e. one week before the experiment. The students were unaware of the upcoming experiment.

We assigned students randomly to the two groups (experimental (AR), control (PC-2D)).

For the experiment, the groups were divided into three different rooms: The 2D group solved the task in individual work in a computer room. The AR group was again divided into two rooms, since only two Hololenses were available for the experiment, and we wanted to avoid priming the waiting participants. They were given a non-topic related task that they could work on during the waiting period.

Both groups had to work through the tutorial for the software (Ariadne 2D or Ariadne AR) before dealing with the task. Afterwards the participants received the identical task to model a computer chess program for both groups.

No time limit has been set. Emerging software problems were logged.

After completing the task, all participants received the MUSIC® Inventory (see section 5.2) to record their motivation in relation to the experiment.

In order to be fair, the submission of the task was excluded from the normal evaluation overview. In addition, all data (self-assessment questionnaire, submission of the task, MUSIC® Inventory) were submitted in anonymised form.

To sum up, the concrete measurement was as follows:

1. Self-assessment questionnaire (one week before the experiment took place)
2. Task procedure depending on the group assignment
3. Interview (directly after the task completion)
4. MUSIC® Inventory

5.1. Self Assessment

The self-assessment questionnaire is self-created and consists of 15 items. The items were created with reference to the learning objectives of the lecture and exercise unit on class diagram modeling. Thereby, we refer to the learning goal taxonomy SOLO (Structure of the Observed Learning Outcome) of Biggs [19] respectively the adapted version of Brabrand [20] with the help of which it is possible to formulate outcome-oriented learning goals.

The SOLO taxonomy has five levels from incompetence (SOLO 1) to the ability to generalize knowledge onto a new domain (SOLO 5). It is used to qualify learning outcomes of students in terms of their complexity. We deliberately asked for low SOLO levels only, as they were novices in this field.

Here are a few examples of items: I can...
...identify the arrow type of a composition (SOLO 2).
...distinguish different types of associations (SOLO 4).
...use an aggregation connection correctly (SOLO 3).

5.2. Questionnaire

The MUSIC® Model of Academic Motivation Inventory (MUSIC® Inventory) is a questionnaire for assessing students' perceptions of the MUSIC® components for an activity or course [21]. The MUSIC® Inventory is a research-based questionnaire producing reliable and valid scores with scales proved by confirmatory factor analysis. Until now, these results were reached at English, Icelandic, Arabic and Spanish speaking countries with Electrical, Systems and Industrial Engineering students [21]–[24]. In the present work, a German translation of the MUSIC® Inventory provided by Brett D. Jones was used.

The questionnaire consists of five components to be considered when designing an instruction, namely: Empowerment, usefulness, success, interest and caring. Each component was derived from educational and psychological research and theory. Take autonomy as an example: Autonomy is a related construct of empowerment, which is why empowerment describes the degree to which a student perceives he/she has control of his/her learning environment in the course.

The same applies for the usefulness of the coursework to students' future (usefulness), the perception to succeed at the coursework (success), whether instructional methods and coursework seem interesting (interest) and how a student receives his or her instructor is caring about his or her success in the coursework and cares about the student's well-being (caring) [21].

If an instructor supports one or more of these components, the students' motivation increases and by that, increased student learning occurs as an outcome.

The items of the questionnaire were rated on a six-point-Likert scale ranging from *strongly disagree* to *strongly agree*.

5.3. Performance

The performance of the students was measured based on the given task. The students could achieve up to 100 points in the task. For scoring, points were awarded based on the modeled classes, the relationships used, the attributes modeled, and methods.

Table 1. Detailed list of awarded points for elements

Issue	Credits
class	5
type of association	2
method	2
attribute	2
multiplicity	1

5.4. Interview

After completion of the task or experiment, we conducted a guideline-based interview with each participant of the experimental group. Usability in handling the Hololens and further self-assessment about the performance in the task were the focus of the interviews. We conducted semi-standardized interviews with three open guiding questions:

- How did you fare with modeling with the Hololens?
- Should there be further toolboxes implemented for this tool? Describe your experience modeling / working with the Hololens
- How many credits would you give yourself in the task? In terms of 100 credits: How would you evaluate yourself?

6. Evaluation

A total of $(N = 14)$ students took part in the experiment. All 14 students have chosen the software engineering course as compulsory elective in the 5th semester. Despite this very small sample size, these are 100% of the students who also attend the lecture (i.e. the population for this study, no sampling has taken place). Accordingly, there were seven participants each for the experimental group and the control group. There are four artefacts (described in the subsections of the method section) per participant that can be evaluated.

For the evaluation we grouped our data two times: First regarding the 2D or AR assignment and second regarding the mean in self-assessment.

The participants answered questions about previous knowledge in the self-assessment questionnaire, so they were separated in two groups, one with means of 2.73 and lower, on with means of 2.87 and higher (students rated a six–point–Likert scale ranging from strongly disagree to strongly agree).

A significant difference was found for the MUSIC® scale *interest*: A T-Test resulted that the group with less previous knowledge had a mean of 3.86 (SD = 0.18), the group with more previous knowledge had a mean of 4.43 (SD = 0.56), t(12) = -2.57, p = .024.

Table 2. *(Cronbach's alpha = .835 without Item 19) Reliability for each scale (Cronbach's alpha) and mean values and standard deviations calculated for experimental group and control group for MUSIC® Inventory

	M 2D group (SD) $N = 7$	M AR group (SD) $N = 7$	Cronbach's alpha $N = 14$
Empowerment	3.89 (.45)	3.94 (.76)	.821
Usefulness	4.40 (.50)	4.51 (.45)	.337 *
Success	4.25 (.92)	4.61 (.57)	.841
Interest	4.07 (.29)	4.21 (.66)	.834
Caring	4.55 (.23)	4.64 (.56)	.749

Apparently, the AR group reached higher values in the MUSIC® scales than the 2D group, but no significant differences or correlations were detected due to the small sample of seven participants each group, but it could be an indicator for higher motivation when using AR (see table 2).

At the task, where 100 credits could have been reached, the 2D group reached a mean of 79.43 (SD = 9.74) points, whereas the AR group reached 68.21 credits (SD = 13.49), but the difference was not significant $t(12) = 1.783$, $p = 0.1$. A possible influencing factor could be the previous knowledge of the control group. In the self-assessment questionnaire the mean of the 2D group was 3.3 (SD = 0.89), the mean of the AR group was 2.78 (SD = 0.64), which could

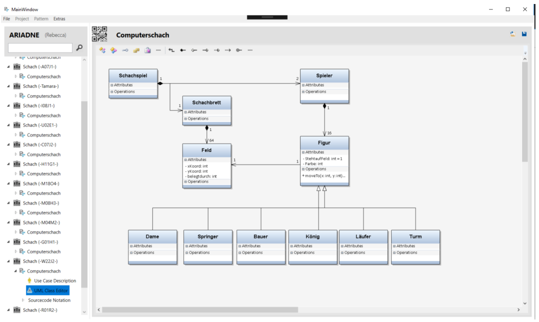

Figure 3. The teacher's perspective in Ariadne

indicate the 2D group had more previous knowledge, but this difference is not significant.

In figure 3 you can see an example of the teachers' perspective in Ariadne 2D; the figure shows the model constructed by a student during the study. Figure 4, in contrast shows a student's model in AR.

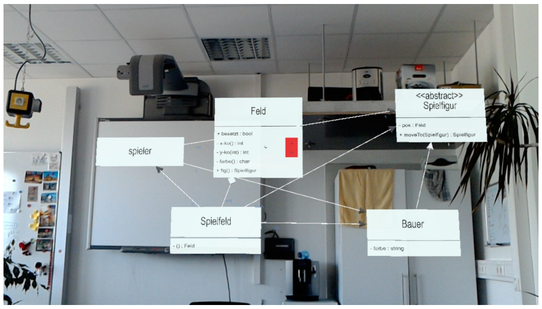

Figure 4. A student delivery during the experiment

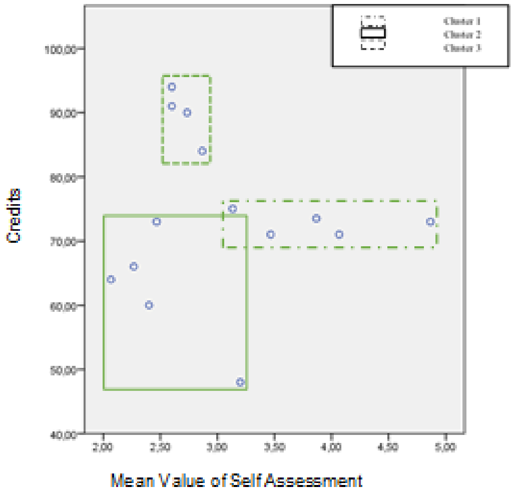

Figure 5. Results of the cluster analysis

Since we are also interested in how well the students perform as novices in the field of modeling, we have conducted a hierarchical cluster analysis.

The cluster analysis is a multivariate procedure which is used to reduce data and to classify subjects to groups or clusters. It is quite similar to the better known principal component analysis, but in contrast to this procedure, it is more focused on the subjects (and not the items). If a cluster analysis is executed, its aim is to form clusters, which are internally very similar but differ from the other clusters as much as possible. For example: Internally, all the created clusters or groups have common attributes, which all the belonging subjects share with the other members of the groups. However, if the groups are compared to each other, they will differ significantly from others [25].

The metric variables of the self-assessment questionnaire in the mean value and the points awarded were used as variables. The cluster analysis produced three clusters (see figure 5) consisting of 2 times 5 and once 4 cases (for 14 participants overall). What is striking about cluster 1 is that the mean value for self-assessment is relatively high (>3.13) in the participants' responses. Furthermore, they scored 72.7 points on average. This cluster consists quite homogeneously of participants of the 2D group and participants of the AR group.

Cluster 2 scored 62.2 points on average with an assessment of 2.4. This cluster consists mainly of the experimental group; only one case comes from the 2D group.

Cluster 3 comprises the cases with the highest scores. Interestingly, it was this cluster that was rated worst (2.6). Here the cluster consists mainly of 2D cases. The correlation between the self-assessment and the credits in the 2D group approves this. There is a negative linear correlation ($r_{\text{bp}} = -.590$) which is not significant ($p = .163$) but it could be an indicator that the participants that do not agree got more credits than the other ones who achieved fewer credits.

We have evaluated the interviews using a qualitative interview analysis method. We could derive three categories depending on the questions, general experience with AR, usability and self-assessment. In the first category, in terms of experience, there is no clear tendency; about half of the participants stated that this was difficult, the other half found it easy. Many reasons could be traced back to the unusual situation. In the usability category mainly problems with the input of the keyboard were stated. Almost all test persons would like a toolbox/toolbar. They rate the category self-assessment rather below average.

7. Limitations & Threats to Validity

Our threats to validity related to this study can be summarized to:

- Our conclusion is based on a small sample. Although all of our current students in that course participated in the study, N = 14 is too small to get significant results and to provide an overall conclusion. We have already planned a follow-up study, which will be evaluated until September 2018.
- Our students were novices in modeling class diagrams. A confrontation with a new tool, a new technology and a new learning content could have led to excessive demands.
- The target group were non-major computer sciences students. The results might be different for students studying computer science.
- We tested with software that was not as complex as the tools we used in the semesters before. Both tools should provide students with the same usability and features in order to receive comparable results regarding performance and motivation. We have not tested against a more complex modeling tool like *Enterprise Architect*.

8. Discussion

Despite the small sample, we can see a tendency in the motivation of students who have modeled with AR compared to those who have modeled with 2D. We could not detect significant correlations and variances between motivation and performance. In general, however, the entire group was very motivated, and the MUSIC® Inventory clearly shows above-average mean values. In addition, no significant difference in the point distribution between the experimental and the control group can be determined. This means that neither group has scored significantly better (or worse) than the other. The aim of this experiment was to find out whether the experimental group *performed equally, was more motivated and performed more successfully in the next tasks using class modeling* than the control group. Regarding the last goal we cannot make an analysis yet, since this experiment was anonymised in order to be fair and to disadvantage none of the students the task was excluded from the rating of the overall course progress. The final exams are in July 2018, and we will ask them to provide their pseudonyms afterwards.

9. Conclusion & Future Works

We present a first approach for realizing a UML modeling tool with AR technology. We did so by rather "naively" mapping desktop user interface concepts to AR. One of our main goals remaining is the user experience of our software. The field of AR is rather young. We expect a multitude of new interaction concepts to be developed. We therefore plan to extend our implementation in respect to interaction concepts suited for AR technology. This also raises the interesting question whether different AR interaction concepts may affect the way the students experience and finally learn the learning contents.

Furthermore, the user interface realized in our software is designed for students with no physical restrictions. Still, we believe that especially the AR settings allows for better accessibility, also for students with restrictions. It is therefore interesting whether there is the possibility to introduce further controls, which allows for using this system even if the students have certain types of impairments.

References

[1] K. Alfert and F. Engelen, "Experiences in 3-dimensional visualization of java class relations," *J. Integr. Des. Process Sci.*, vol. 5, no. 3, pp. 91–106, Aug. 2001, ISSN: 1092-0617.

[2] C. S. C. Rodrigues, C. Ufrj, C. M. L. Werner, C. Ufrj, and C. Universitária, "VisAr3D : an innovative 3D visualization of UML models," *IEEE International Conference on Software Engineering Companion*, pp. 451–460, 2016, ISSN: 02705257. DOI: `10.1145/2889160.2889199`.

[3] L. Feijs and R. De Jong, "3d visualization of software architectures," *Commun. ACM*, vol. 41, no. 12, pp. 73–78, Dec. 1998. DOI: `10.1145/290133.290151`.

[4] H. Sanatnama and F. Brahimi, "Graph Drawing Algorithms: Using in Software Tools," *Journal of Applied Sciences(Faisalabad)*, vol. 10, no. 17, pp. 1894–1901, 2010. DOI: `10.3923/jas.2010.1894.1901`.

[5] J. von Pilgrim and K. Duske, "Gef3D a Framework for Two-, Two-and-a-Half-, and Three-Dimensional Graphical Editors," *Proceedings of the 4th ACM symposium on Software visuallization - SoftVis '08*, p. 95, 2008. DOI: `10.1145/1409720.1409737`.

[6] R. Reuter, F. Hauser, C. Gold-veerkamp, J. Mottok, and J. Abke, "Towards a definition

and identification of learning obstacles in higher software engineering education," in *Proc. 9th International Conference on Education and New Learning Technologies (IATED EDULEARN)*, Barcelona, 2017, pp. 10 259–10 267.

[7] K. Siau and P.-P. Loo, "Identifying difficulties in learning uml," *Information Systems Management*, vol. 43, pp. 43–51, 2006. DOI: `10.1201/1078.10580530/46108.23.3.20060601/93706.5`.

[8] J. Zhang, Y.-T. Sung, H.-T. Hou, and K.-E. Chang, "The development and evaluation of an augmented reality-based armillary sphere for astronomical observation instruction," *Computers & Education*, vol. 73, pp. 178–188, 2014. DOI: `https://doi.org/10.1016/j.compedu.2014.01.003`.

[9] S.-J. Lu and Y.-C. Liu, "Integrating augmented reality technology to enhance children's learning in marine education," *Environmental Education Research*, vol. 21, no. 4, pp. 525–541, 2015. DOI: `10.1080/13504622.2014.911247`.

[10] J. Sweller, "Element Interactivity and Intrinsic , Extraneous , and Germane Cognitive Load," *Educ Psychol Rev*, vol. 22, pp. 123–138, 2010. DOI: `10.1007/s10648-010-9128-5`.

[11] ——, "Cognitive Load During Problem Solving : Effects on Learning," *Cognitive Science*, vol. 285, no. 12, pp. 257–285, 1988.

[12] ——, "Cognitive load theory, learning difficulty, and instructional design," *Learning and instruction*, vol. 4, no. 4, pp. 295–312, 1994.

[13] C. Weinstein, "The teaching of learning strategy," *The handbook of research on teaching*, pp. 315–328, 1986.

[14] W. M. Trochim, "An introduction to concept mapping for planning and evaluation," *Evaluation and Program Planning*, vol. 12, no. 1, pp. 1–16, 1989. DOI: `https://doi.org/10.1016/0149-7189(89)90016-5`.

[15] J. Bacca, S. Baldiris, R. Fabregat, S. Graf, *et al.*, "Augmented reality trends in education: a systematic review of research and applications," *Journal of Educational Technology & Society*, vol. 17, no. 4, p. 133, 2014.

[16] M. Akcayir and G. Akcayir, "Advantages and challenges associated with augmented reality for education: A systematic review of the literature," *Educational Research Review*, vol. 20, pp. 1–11, 2016.

[17] P. McIntosh and M. Hamilton, "X3D-UML: 3D UML mechatronic diagrams," *Proceedings of the Australian Software Engineering Conference, ASWEC*, pp. 85–93, 2010, ISSN: 1530-0803. DOI: `10.1109/ASWEC.2010.14`.

[18] *Nclass - free uml class designer*, `http://nclass.sourceforge.net/`, (Accessed on 03/10/2018).

[19] J. B. Biggs and K. F. Collis, *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome)*. Academic Press, 2014.

[20] C. Brabrand and B. Dahl, "Using the solo taxonomy to analyze competence progression of university science curricula," *Higher Education*, vol. 58, no. 4, pp. 531–549, 2009, ISSN: 00181560, 1573174X.

[21] B. D. Jones and V. Tech, "Motivating Students to Engage in Learning: The MUSIC Model of Academic Motivation," *International Journal of Teaching and Learning in Higher Education*, vol. 21, no. 2, pp. 272–285, 2009, ISSN: 1812-9129.

[22] B. D. Jones and G. Skaggs, "Measuring Students' Motivation: Validity Evidence for the MUSIC Model of Academic Motivation Inventory Measuring Students' Motivation: Validity Evidence for the MUSIC Model of Academic Motivation Inventory," *International Journal for the Scholarship of Teaching and Learning*, vol. 10, no. 1, 2016. DOI: `10.20429/ijsotl.2016.100107`.

[23] B. D. Jones, M. Li, and J. M. Cruz, "A Cross-Cultural Validation of the MUSIC® Model of Academic Motivation Inventory: Evidence from Chinese- and Spanish-Speaking University Students," *International Journal of Educational Psychology*, vol. 6, no. 1, p. 25, 2017. DOI: `10.17583/ijep.2017.2357`.

[24] H. Mohamed, M. Soliman, and B. Jones, "A cross-cultural validation of the music model of academic motivation and its associated inventory among egyptian university students," *Journal of Counseling Quarterly Journal*, vol. 36, pp. 2–14, 2013.

[25] J. Bortz and C. Schuster, *Statistik für Human- und Sozialwissenschaftler*. Springer Berlin Heidelberg, 2010, ISBN: 9783642127694.