A Case Study on the Rapid Development of Natural and Synergistic Multimodal Interfaces for XR Use-Cases

Chris Zimmerer
Julius-Maximilians-Universität
Würzburg, Germany
chris.zimmerer@uni-wuerzburg.de

Martin Fischbach
Julius-Maximilians-Universität
Würzburg, Germany
martin.fischbach@uni-wuerzburg.de

Marc Erich Latoschik
Julius-Maximilians-Universität
Würzburg, Germany
marc.latoschik@uni-wuerzburg.de

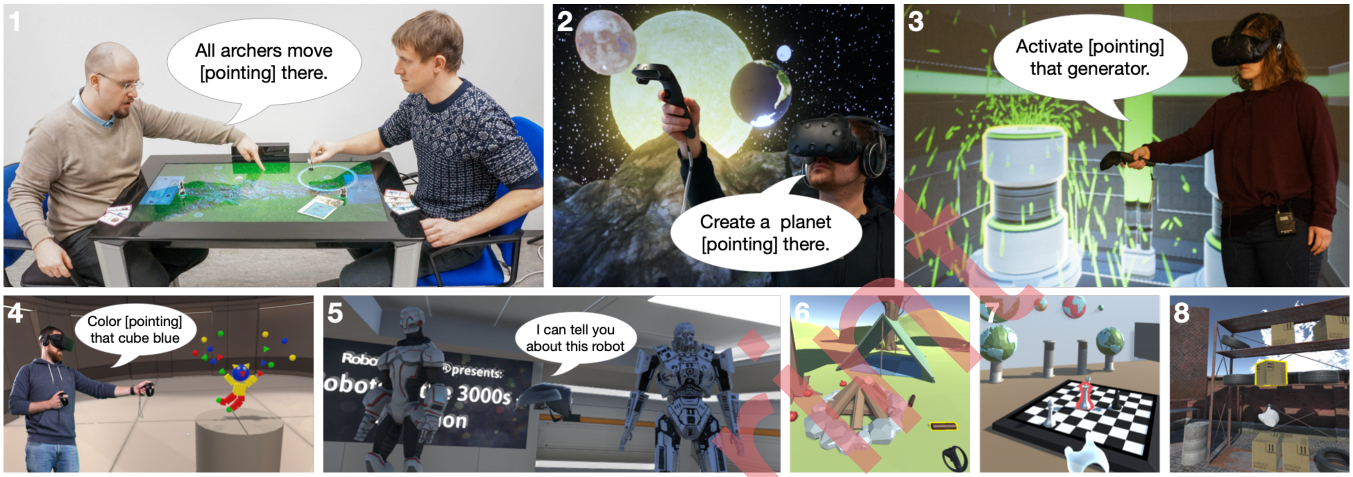

1
All archers move [pointing] there.
2
Create a planet [pointing] there.
3
Activate [pointing] that generator.
4
Color [pointing] that cube blue
5
I can tell you about this robot
presents:
3000s
6
7
8

Figure 1: Applications with synergistic speech and gesture interfaces for an interactive surface (1) and virtual reality (2-8) implemented with our toolchain by us and our students.

preprint

ABSTRACT

Multimodal Interfaces (MMIs) supporting the synergistic use of natural modalities like speech and gesture have been conceived as promising for spatial or 3D interactions, e.g., in Virtual, Augmented, and Mixed Reality (XR for short). Yet, the currently prevailing user interfaces are unimodal. Commercially available software platforms like the Unity or Unreal game engines simplify the complexity of developing XR applications through appropriate tool support. They provide ready-to-use device integration, e.g., for 3D controllers or motion tracking, and according interaction techniques such as menus, (3D) point-and-click, or even simple symbolic gestures to rapidly develop unimodal interfaces. A comparable tool support is yet missing for multimodal solutions in this and similar areas. We believe that this hinders user-centered research based on rapid prototyping of MMIs, the identification and formulation of practical design guidelines, the development of *killer applications* highlighting the power of MMIs, and ultimately a widespread adoption of MMIs. This article investigates potential reasons for the ongoing uncommonness of MMIs. Our case study illustrates and analyzes lessons learned during the development and application of a toolchain that supports rapid development of natural and synergistic MMIs for XR use-cases. We analyze the toolchain in terms of developer usability, development time, and MMI customization. This analysis is based on the knowledge gained in years of research and academic education. Specifically, it reflects on the development of appropriate MMI tools and their application in various demo use-cases, in user-centered research, and in the lab work of a mandatory MMI course of an HCI master's program. The derived insights highlight successful choices made as well as potential areas for improvement.

CCS CONCEPTS

• **Human-centered computing → Interaction paradigms**; **Systems and tools for interaction design**.

KEYWORDS

Multimodal Interface, Speech and Gesture, Software Architecture, Multimodal Fusion

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
CHI '22 Extended Abstracts, April 29-May 5, 2022, New Orleans, LA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9156-6/22/04.
https://doi.org/10.1145/3491101.3503552

ACM Reference Format:
Chris Zimmerer, Martin Fischbach, and Marc Erich Latoschik. 2022. A Case Study on the Rapid Development of Natural and Synergistic Multimodal Interfaces for XR Use-Cases. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI '22 Extended Abstracts), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3491101.3503552

1 INTRODUCTION

Multimodal Interfaces (MMIs) are based on the user's natural communication skills [34] and allow the potential simultaneous use of at least two different input modalities [31]. Over 40 years ago, Bolt pioneered this interaction for graphical user interfaces [3]. In the seminal „Put that there" demonstration users could create, select, modify, and delete 2D objects on a large projection using natural input modalities such as speech and gestures synergistically. Since then, a number of potential benefits of MMIs have been proposed by the research community including an increased expressiveness, flexibility, reliability, and efficiency [33, 36, 37, 39]. MMIs have been conceived as particular promising when users are (physically) situated in the application and share a frame of reference [9], e.g., in smart homes [38] and human-robot interaction [5] as well as in Augmented [15], Mixed [28], and Virtual Reality [18, 21] (AR, MR, and VR; XR for short). Here, speech and gesture are considered to be the most powerful combination of input modalities [35, p. 4] for selection and system control tasks [27] because of their expressive power and complementarity [8]. Users can easily describe semantically rich information such as actions or the visual appearances of objects using speech, while expressing extensive references using gestures, e.g., to positions (deixis), to shapes (iconics), or to movements (kinemimics).

Despite the proposed advantages and the particular suitability for XR, the currently prevailing user interfaces are unimodal. They consist of graphical menus operated by spatial 3D input devices such as physical controllers with 3D position and rotation tracking, push-buttons, and joysticks. The development of such systems has been greatly facilitated by technological advancements. Commercially available software platforms like the Unity [42] or Unreal [14] game engines simplify the complexity of developing XR applications through appropriate tool support: graphical editors support the design of virtual environments through drag-and-drop and node-based visual scripting approaches the implementation of application logic. Plugins like the XR Interaction Toolkit [45] or the Virtual Reality Toolkit [44] provide ready-to-use device integration, e.g., for head-mounted displays, 3D controllers, or motion tracking, and according unimodal interaction techniques such as menus, (3D) point-and-click, or even simple symbolic gestures. Tool support has reached a high level of maturity, enabling rapid development of unimodal interfaces for XR. This supports user-centered research and has led to the identification and formulation of many practical design guidelines and the widespread adoption of unimodal interfaces in this area (see LaViola Jr et al. [27] for a comprehensive overview of 3D user interfaces).

However, comparable tool support is yet missing for natural and synergistic MMIs for XR. There is comparatively less user-centered research with functional MMIs and subsequently less practical guidelines [37, pp. 449–478], as well as a distinct lack of *killer applications* highlighting the power of MMIs. This poses a causality dilemma between tool support, research/guidelines, and *killer applications*: It raises the question whether the lack of research and killer applications is a consequence of comparably poor tool support, or the poor tool support a consequence of the lack of research and killer applications? Lalanne et al. [20] announced the maturity of MMIs and their technology more than a decade ago. To date, this maturity only applies to interfaces that allow sequential or alternative use of natural modalities in less complex interaction environments, e.g., touch and voice input in 2D graphical interfaces of smartphones. It does not extend to synergistic MMIs for spatial or 3D interaction environments. Although the reliability of unimodal recognition systems for speech and gestures has been considerably improved by modern machine learning methods [7, 30], there are currently no ready-to-use solutions that allow the joint analysis of inputs from both modalities to derive a collective meaning, i.e., multimodal fusion, and the integration of the resulting MMI into a concrete application context to trigger functionality, i.e., semantic integration. Such tools must not only overcome the difficulties of performing multimodal fusion [18, 50] and semantic integration [9, 23], but also meet the requirements of the User-Centered Design (UCD) [1, 32] and agile software development [13] philosophies. The UCD proposes an iterative process where the **rapid development of functional prototypes** plays an essential role. Modern agile software development methods like Scrum or Extreme Programming [13] emphasize the requirement to **be flexible to changing requirements** and details the frequency of iterations to: **weeks instead of months**. Both philosophies consequently also contain implicit requirements to the **usability of the toolchain for developers** - to support or at least not to hinder the rapid iterative development of prototypes. We believe that a tool(chain) that enables the rapid development of natural & synergistic MMIs and that satisfies these requirements will resolve the causality dilemma. It will considerably support user-centered research based on rapid prototyping of MMIs, the identification and formulation of practical design guidelines, the development of killer applications highlighting the power of MMIs, and ultimately lead to a more widespread adoption of MMIs.

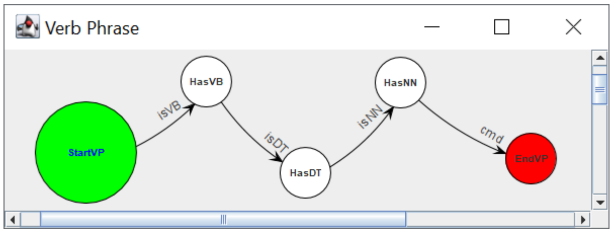
This case study illustrates and analyses lessons learned during the development of a toolchain for rapid development of natural and synergistic MMIs and its application for several XR use-cases. We analyze the toolchain in terms of requirements posed by the UCD and agile philosophy: developer usability, development time, and MMI customization. This analysis is based on the knowledge gained in years of research and academic education. In particular, it reflects on the development of appropriate MMI tools and their utilization (I) in the implementation of demo applications, (II) in conducting user-centered research based on rapid prototyping of MMIs, and (III) in the practical work of the of the mandatory MMI [26] course of the HCI master's program [25] at the University of Würzburg in Germany. We pose the following questions to structure our discussion and to reflect on the progress made as well as highlight potential areas for improvement:

(Q1) Does our toolchain enable the implementation of natural and synergistic MMIs for XR use cases?
(Q2) How well does it support the rapid development of natural and synergistic MMIs in terms of developer usability, development time, and MMI customization?
(Q3) What are the remaining obstacles to make the development of MMIs as easy as their unimodal alternatives?

preprint

```
1  create StartState "StartVP" withArc        "isVB"  toTarget "HasVB"
2  create State      "HasVB"   withArc        "isDT"  toTarget "HasDT"
3  create State      "HasDT"   withArc        "isNN"  toTarget "HasNN"
4  create State      "HasNN"   withEpsilonArc "cmd"   toTarget "EndVP"
5  create EndState   "EndVP"
6
7  create Arc        "isVB" withCondition isVerb andFunction doVerb
8  ...
9  create EpsilonArc "cmd"  withFunction updateApplicationState
```

(a) Defines a simple imperative verb phrase, e.g., *select a ball*.



(b) A visual representation of the cATN's graph.

Figure 2: A sample code excerpt to create a simple interface using the cATN's description language and its visual representation.

2 TOOLS AND TECHNOLOGY

We developed a multimodal fusion method called the concurrent Augmented Transition Network (cATN) [50] for the research software platform Simulator X [24]. The cATN is the result of a comprehensive requirements analysis and the successor of the temporal Augmented Transition Network [22]. It features a code-native description language that allows developers to describe multimodal utterances declaratively. Figure 2a showcases an example code excerpt that defines a simple imperative verb phrase, consisting of a verb, followed by a determiner, and finally a noun (line 1-4). Figure 2b depicts the cATN's visualization tool, which graphically represents the transition network defined in this way. The cATN parser moves concurrent cursors from one state to another by checking the conditions of the arcs against the input it receives from the respective recognizers. For example, line 7 `isVerb` checks whether the input is from a speech recognizer and is of type verb, taking into account both timestamps and confidences. If this check passes, line 7 `doVerb` stores the input in the registers of the transitioning cursor. A dedicated arc automatically performs semantic integration based on the contents of the cursor's registers if a respective end state is reached (line 4 and 9). The cATN is compatible with the probable future rise in machine learning approaches for multimodal fusion, e.g., by natively supporting features to handle probabilistic user input hypotheses.

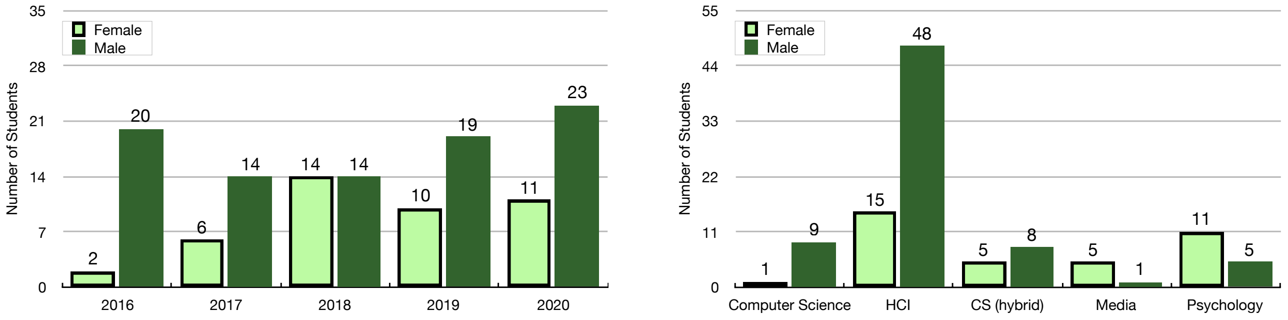
We use the Unity game engine and the XR Interaction Toolkit [45] to implement HMD-based VR applications. Unity's graphical editor and wide range of freely available assets facilitate comparatively easy development of virtual environments, while the scripting approach with C# makes it easy to implement the application's functionality, e.g., to create, change, or delete virtual objects in the environment. To enable semantic integration, we connect Simulator X with Unity via a dedicated transport layer for software platforms [46] based on an entity-event state decoupling and exchange approach. It bidirectionally synchronizes relevant parts of Unity's application state with a dedicated interaction context in Simulator X. The cATN can query this interaction context based on semantics-based software techniques [9] to perform semantic integration directly during multimodal fusion. For example, while parsing the user input: "*color [pointing] that green ball yellow*", the cATN can retrieve an object of type ball that has the color green, check if the user is pointing at it, and instruct Unity to color it yellow. We use the Microsoft Speech SDK as an automatic speech recognizer since it provides n-best guesses, timestamps, and confidences for each input and supports offline processing.

3 DEMONSTRATIONS AND RESEARCH

A summary of our work in the field of MMIs can be found on our webpage [52]. The first application that used an early version of the cATN was developed by two of our master students with Simulator X: The Quest V2 prototype [49] is a digital tabletop game in which players can move physical playing pieces by means of a tangible user interface but also command virtual playing pieces via a synergistic speech and gesture MMI (Figure 1, 1). We published this work as a demonstration at the IEEE Virtual Reality (IEEE VR) conference in 2016 [28]. We then developed two VR demonstrations to show the feasibility of our connection approach between the cATN and commercial game engines: The Big Bang demonstration [48] is an HMD-based VR application implemented with the Unreal Engine that allows the user to create a solar system via a synergistic speech and gesture MMI (Figure 1, 2). The Space Tentacle demonstration is an HMD-based VR adventure game implemented with Unity where the user has to multimodally interact with an artificial intelligence on a crashed space ship to escape (Figure 1, 3). The latter has been published on the IEEE VR conference in 2018 [51]. After refining the toolchain, a master student implemented another demonstration using Unity and the cATN called Robot Museum [16] (Figure 1, 5).

The development of these demonstrations has helped us to continuously evolve our toolchain in terms of supported features, performance, and usability. We published the cATN as a tool for multimodal fusion and semantic integration at the end of 2018 in the Multimodal Technologies and Interaction journal [50]. Afterwards, we applied our toolchain to empirically research MMIs in VR. In 2019 and 2020, we published two user studies at the International Conference on Multimodal Interaction [47, 53] comparing synergistic multimodal –speech & gesture– interfaces against unimodal –menu-based– interfaces for VR design applications and their implications on the users' creative performance (Figure 1, 4). The multimodal VR design application and its MMI are a direct result of a follow-up project conducted by two students after their participation in the MMI course in 2018. These publications received a Best Paper Runner-Up Award and a Best Paper nomination from the leading conference in the field, highlighting the importance of empirical research with functional interfaces. They contribute to the comparatively small body of empirical research in this area by providing concrete insights into synergistic speech and gesture interfaces for a specific task and application domain. It further highlights how better tool support can resolve the causality dilemma and support user-centered research and the identification and formalization of more practical guidelines and *killer applications*.

preprint



(a) Students divided by year and gender.

(b) Students divided by bachelor degree and gender.

Figure 3: Number of students registered for the Multimodal Interface course from 2016 to 2020.

preprint

4 MULTIMODAL INTERFACE COURSE

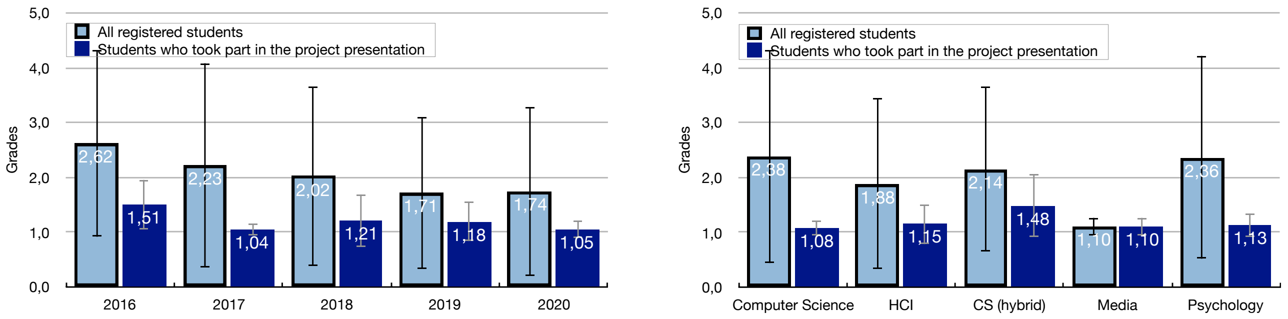
The Multimodal Interface course [26] is part of the mandatory curriculum of the research-oriented master program Human-Computer Interaction [25] at the University of Würzburg in Germany. The program provides an interdisciplinary study of information technology issues in the context of psychological factors. The MMI course concentrates on the analysis of multimodal input, i.e., how to perform multimodal fusion and semantic integration. It implements an active learning approach [4] to facilitate learning. Specifically, the method of Learning-by-Design [19, 29] is used, in which students must practically apply theoretical knowledge to solve a complex task. In the MMI course, this task is to implement a VR application with a functional MMI that supports a set of synergistic speech and gesture commands, similar to the „Put that there" demonstration of Bolt [3] but for VR. Students have to cooperatively solve this task in teams up to three persons since small-group learning proves beneficial regarding learning outcome [41]. The self-driven implementation of such an application aims to provide a deeper understanding of the theory, related technologies, and practice-oriented competencies for applying the theory to real-world problems.

The structure of the MMI course, similar to the Machine Learning and 3D User Interface course of the HCI study program, is divided into three parts: (1) The required theoretical knowledge is taught in weekly two hour lectures. After an introduction to the field, it introduces algorithmic parsers for simple context free languages and progresses to more complex parsers capable of processing context sensitive multimodal languages, i.a., the cATN. It concludes with knowledge representations and software techniques for performing semantic integration. (2) In the two-hour weekly exercises, the supervisor hands out and discusses assignment sheets to familiarize students with the technologies. They start with an introduction to the project management tool GitLab [17] and the versioning control system Git [6] that student groups have to use during the semester. Afterwards, it focuses on the game engine Unity and how to develop simple virtual environments for VR, by leaning on the official documentation and tutorials. The exercise proceeds with an introduction to the cATN by discussing the source code of example applications that become increasingly complex. The final exercise sessions are about putting everything together so that students achieve a first working version of their application. (3) Students have to work on the assignment sheets in additional sessions with their team during the lecture period. For this purpose, we provide students with a computer lab equipped with eight VR capable computers with HMDs. During the lecture-free period, students continue working on their applications in a self-directed manner until the project presentation at the end of the semester. The MMI course is a 5-ECTS module which, in accordance with the European Credit Transfer and Accumulation System, implies a workload of approximately 150 hours, divided into 30 hours each for attending lectures and exercises and 90 hours for project work.

We grade students' projects based on a requirement catalogue published in the first exercise session and an individual questioning about the source code and the theory behind it at the end of the semester during the project presentation. The catalogue defines a series of mandatory requirements that each team has to fulfill in order to pass and a pool of optional requirements from which students can choose to improve their grade. These requirements describe features of the prototype related to the virtual environment and the MMI, but also to the documentation and evaluation of the interface. For example, one requirement states that the application has to support the selection of virtual objects using speech-accompanying pointing gestures and provides an exemplary interaction for clarification: "Select [pointing] that ball." A noteworthy requirement is an *own idea* in which students can propose a feature that they would like to implement, e.g., a more sophisticated application context.

4.1 Students and Prior Knowledge

There were 133 registrations (90 male, 43 female) for the MMI course from 110 individual students (72 male, 38 female) over a 5-year period (see Figure 3a). The MMI course is typically taken by students in their first or second semester, making it one of the first comprehensive software development projects in the HCI master program. Due to the heterogeneous composition of the students, previous experience in computer science and software development range from very good to nonexistent. Figure 3b provides an overview of the students' bachelor degrees divided in five categories (only 108 of the 110 students). The categories reach from pure computer science programs over computer science hybrid courses such as Business Informatics and Human-Computer Interaction to study programs in the domain of media to pure psychology programs.



(a) Average grades subdivided per year.

(b) Average grades subdivided according to students' bachelor degree.

Figure 4: Grades divided in overall average and average without students who did not take part in the project presentation.

4.2 Changes over the years

In the years before 2018, students used an early version of the cATN implemented in Simulator X to perform multimodal fusion and semantic integration. They did not use Unity or HMD-based VR but implemented a simple environment for a large screen or projection wall using Simulator X. Starting 2018, students used the toolchain described in section 2. In 2020, we had to adjust the requirements for the project due to the Corona pandemic and the closure of all university facilities. The exercise was prepared as a block course that took place in the lecture-free period on short notice, after the administration allowed heavily restricted access to the university again. Each student had two days in our computer lab where they were presented with a an already finished implementation of the application. Under the supervision of the exercise leader students had to familiarize with the code, understand the software architecture, and make minor adaptations to it.

4.3 Applications and Grades

As part of the requirements for passing the MMI course, students created short videos showcasing their applications. Since the project allowed the implementation of own environments, each application is different. Especially the results of the MMI course in 2018 [10] and 2019 [11] are most telling in what students achieved in just one semester using our toolchain. In addition, results before 2018 and after 2019 can be found on the HCI chair's Youtube channel [12].

The 110 students participated in 133 project presentations with an overall average grade of 2.01 (1.0 is the best grade and 5.0 means failure). Out of the 110 students, 104 eventually passed the exam while six did not pass the exam yet. Out of the 104 who passed, 86 passed at the first try, 17 after the second try, and one after the third try. Out of the six students who did not pass the exam, four only tried it once, one tried it twice, and one tried it four times. However, of all 29 exams that are marked as failed, no student actually failed during the project presentation, e.g., as a consequence of a poorly implemented application or poor performance in the individual questioning. In all cases, students did not attend the project presentation because they, e.g., lost interest or deferred the course to a later semester, and therefore received a grade of 5.0. Figure 4a shows the average grade for each year while Figure 4b provides an overview of the average grading categorized by prior education.

4.4 Official Course Evaluation

The University of Würzburg conducts an official evaluation where students can rate each course based on three question groups regarding the overall experience, the lecture, and the exercise. Students rate each item on a 5-point scale ranging from 1 to 5 (best to worst). Unfortunately, this assessment takes place during the lecture period when students do not yet have an overall picture of the course. We did not receive results for 2016, 2017, and 2020 due to a lack of participation. However, 11 students in 2018 and 16 students in 2019 rated the overall course as good ($M = 2.4$, $SD = 0.8$ in 2018, $M = 2.0$, $SD = 1.0$ in 2019), but slightly above average in difficulty ($M = 3.7$, $SD = 0.8$ in 2018, $M = 3.4$, $SD = 1.0$ in 2019). Students agreed that the practical work in the exercise helped to better understand the theoretical content of the lecture ($M = 2.1$, $SD = 1.0$ in 2018, $M = 1.8$, $SD = 0.9$ in 2019).

4.5 Semi-Structured Interviews

We conducted 13 semi-structured interviews with eight students (three males, five females) who participated in the MMI course in 2018 and 2019 to gain in depth insights. All participants passed the course and have very different previous experience in the field of computer science. For the analysis of the qualitative data from the interviews, we follow an inductive approach [43]. The goal is to find common, predominant, or significant themes that summarizes the raw data and convey key insights. To this end, we firstly identified relevant text segments, labeled these segments to create categories, further condense these categories by reducing overlap and redundancies, to finally create a model that incorporates the most important categories. For this process we used an affinity diagram. We present a sub set of our results that is relevant for analyzing our toolchain. In total, we were able to form 17 categories that are grouped into 3 main categories. Each category is presented with a label and a short description that also includes quotations from the participants.

4.5.1 Overall Concept. The first category summarizes findings regarding the lecture, exercise, and project.

Lecture: The lecture was perceived as comparatively difficult and overall theoretical: "*The change from introduction to theory was very sudden and intense.*" "*I was not sure if I had understood everything correctly.*"

preprint

Exercise: All participants agreed that the difficulty of the exercises steadily increased. There are different opinions about the entrance difficulty. Students with more previous knowledge felt that an introduction to Git, GitLab, and Unity was not necessary: “*Why did I have to hear about Git for the 50th time?*” “*At the beginning it was very easy because I already had experience with Unity.*” However, students with less or no experience in computer science described the first exercises as quite demanding: “*At the beginning it was very difficult, although it was actually about simple things.*”

Project: All participants liked the project and agreed that it was a lot of fun, but also stressful and work intensive: “*The project was fun and I would have done it even if it hadn’t been mandatory.*” “*I had the impression that it was too much work for the 5 ECTS.*” The space for own ideas was found to be particularly motivating. They found that the large project motivated them more than small, disjointed exercises would have: “*The project has increased my motivation because it was not just a matter of surviving the course but of really getting to grips with its contents.*” Respondents were proud of what they have achieved: “*I have included the video of my project in my application portfolio ... that is really great.*”

Comparison to other project-based HCI courses: All participants agreed that the Machine Learning course project was more difficult and labor intensive than the project in the MMI course, due to the mathematical knowledge required to implement a machine learning algorithm. The project in the 3DUI course was perceived the easiest, as it could be solely implemented with Unity or Unreal.

4.5.2 Project. The second category summarizes the project work.

Working on the Project: All interviewees described a steep learning curve: “*At first it was difficult to understand the whole thing, how things are connected.*” “*Once we understood how everything works, we got into a good flow and were able to implement everything on our own.*” However, students with less prior knowledge felt intimidated in the first couple of weeks: “*Since it was the first time that I had to do such a project, it totally intimidated me in the beginning.*” “*In between I also made a list of pros and cons whether I should drop the course or the whole study.*”

Used Technology: Participants rated implementing a simple VR application with Unity as comparatively easy, due to its graphical editor and the vast availability of documentation, tutorials, and forum posts. Using the cATN’s description language to construct graphs has been described as a bit more difficult, but still comfortably doable, especially with the help of the cATN’s visualization tool: “*The description language is logical and helps to build up the network. When we first saw it, we found it intuitive.*” Understanding how the parser works was not always straight forward: “*Due to concurrent cursors, it was sometimes complex to track what was happening*”. All participants agreed that the most difficult part, however, was to understand the overall system architecture of the project. Especially the synchronization between Unity and Simulator X and the implication this design has on performing semantic integration: “*Working with unity and cATN alone was actually okay, but understanding their combination proved quite difficult*”.

Supporting Materials: Depending on their previous knowledge in software development, student rated the importance of supporting materials differently. While more knowledgeable students found the code of example applications for the cATN to be most valuable, students with less experience (or no experience at all) were unable to use and adapt this code for their own application: “*The code examples were less helpful because I didn’t understand what they said. I didn’t understand how to transfer them to my own specific problems.*” The desire for API documentation for the cATN was expressed by all, but its importance was described mainly by students with less experience: “*Documentation is not a must because we had the code, but probably it would have been faster with it.*” Finally, the students agreed that an automatic log of the entire processing pipeline and especially the cATN would have been very helpful.

4.5.3 Learning Outcome: The last category summarizes findings regarding the students’ learning outcome.

Learning by Design: All interviewees strongly felt that the project helped to better understand the theoretical content of the lecture and better keep the knowledge in the long term: “*If I had only participated in the lecture, I would not have been able to develop such an understanding, I needed the practice.*” Similarly, all participants preferred the project over a written exam: “*Things you learned once for a written exam you forget very quickly, but things you did yourself you can remember much better, the knowledge is somehow more sustainable.*” However, the content that was not part of the practical work was forgotten much faster: “*I can only remember the cATN, but nothing about the other parsers.*”

Soft Skills: Beyond the content of the course, almost all respondents reported that they acquired additional personal, social, and methodical competencies. Especially students with less prior knowledge reported gaining more self esteem in managing and conducting complex projects with a small team: “*MMI was a great entry-level project because you learned a lot and projects after that seemed much easier.*” “*I learned not to despair immediately, but to sit in front of a problem for a week and then actually be able to solve it.*” “*It gave me a lot, not only in terms of content, but also socially and organizationally, e.g., collaborating with others and using project management software like Git and GitLab.*”

Confidence for Future Use: All participants reported that they feel confident to use the introduced tools again for another project, and two of the interviewees actually did use it in a follow up project: “*After the course, I said: yeah cool, now I can do something, and I really learned something! So I decided to do a follow-up project.*”. “*I would be very confident to develop a multimodal interface with this tool as part of another project.*”

5 DISCUSSION AND CONCLUSION

We start the discussion by answering the first of the proposed questions *Q1*: The four proof-of-concept demonstrations and in particular the results of the Multimodal Interfaces course demonstrate that our toolchain enables the implementation of synergistic MMIs for XR. Even students with little or no experience in software development were able to successfully use our toolchain, as evident by the MMI course results. We have also conducted and published two user studies with functional MMIs developed with our toolchain. These not only demonstrate the technical maturity of our toolchain and its suitability for research, but also provide empirical results on the design of MMIs.

To answer *Q2*, we reflect on how well our toolchain supports the rapid development of such interfaces in terms of development time,

preprint

developer usability, and customization of MMIs: (1) The project work of the MMI course is estimated to be approximately 90 hours per student based on the 5 ECTS. Many of the interviewed students felt that the course required an above average amount of work, but only less than half of the work is directly related to the interface development. The rest is for developing the VR environment, writing documentation, recording videos, or conducting a small interface evaluation. Thus, we estimate that the interface development effort is on average around 45 hours. It can be assumed that the development time for future applications will be less, since these numbers are based on students working with this toolchain for the first time. This is supported by the confidence in using the toolchain that each student expressed in the interviews. Such a development time is consistent with the iteration frequency proposed by the agile philosophy, which calls for weeks instead of months. (2) The cATN's description language and its visualization tool were mentioned positively regarding developer usability. The students reported that they had no difficulty understanding and using the description language, characterizing it as intuitive. The associated tool for visualizing graphs was found to be helpful for troubleshooting. (3) The decision to use a description language and a corresponding algorithmic parser approach has proven beneficial in terms of MMI customization. Developers can describe MMIs declaratively and customize them by simply changing fragments in the description language.

Due to its architecture, the cATN is suitable to be extended by machine learning approaches, e.g., recognition of synonyms using word embeddings. While we already benefit from better recognition results for unimodal speech and gesture input through modern machine learning methods [7, 30], we must take care not to restrict interface customization when integrating more methods in our tools. Data-driven approaches tend to be less compatible with rapid development. They require large corpora of training data, the selection of relevant features, as well as the careful tuning of learning parameters and model hyperparameters [2, 37, 40]. These training and optimization phases are time-consuming and require in-depth knowledge, which risks making the development of an interface and its customization comparatively costly. In summary, our toolchain supports the rapid development of synergistic MMIs. The design choices to use a description language, an algorithmic parser approach, and a visualization tool seem to be beneficial in terms of supporting development time, developer usability, and customization of MMIs.

However, there is still room for improvement and obstacles that need to be overcome to close the gap between unimodal and synergistic multimodal interface development (*Q3*). (1) The fact that the cATN is part of the research platform Simulator X increases the complexity of the toolchain substantially. Students reported that they had the most difficulties to understand the inter connection between Unity and Simulator X and its implications on performing semantic integration. Packaging the cATN as a plugin for a commercial platform will drastically reduce this complexity. The downside is that the cATN can then only be used in that platform. However, the upside of making the toolchain more comprehensible and easier to use for developers justifies this downside in our opinion. (2) While the frontend of the cATN, i.e., the description language, has been perceived predominantly positive, some students encouraged us to experiment with graphical editors. This idea is inspired by the visual scripting approaches employed by commercial game engines, e.g., Blueprints in the Unreal engine. In the future, we would like to explore the design space of a graphical editor for the cATN to create multimodal interfaces via drag-and-drop of predefined transitions and investigate its impact on the developer usability. We also want to support the development process with a better tool for troubleshooting that provides more detailed information about each processing step during multimodal fusion and semantic integration. (3) Another important aspect that has been raised by an overwhelming majority of students is the necessity of supplementary materials such as documentation, tutorials, and forum posts. Unity is managed by a large company and has a large and active community that are able to provide ample materials. However, as a university, we simply lack the resources to accomplish this on a similar scale for the rest of the toolchain. This lack of resources is also a limiting factor in ultimately providing a fully implemented graphical front-end, visualization tool, and overall supported toolchain that would be comparable to what large companies can achieve. However, we have to explore different design spaces to develop design proposals and guidelines that can be adopted by industry to develop and maintain commercial products for the future.

In conclusion, we showcased the suitability of our toolchain for rapidly developing natural and synergistic MMIs for three XR use cases: developing demo applications, conducting user-centered research, and its application in teaching. We provided insights in terms of development time, developer usability, and MMI customization. In addition, we pointed out potential areas for improvement to further close the gap regarding development effort between unimodal and natural & synergistic MMIs. We hope that closing this gap will result in better tool support, more empirical research towards practice-oriented guidelines for MMIs and ultimately to overall more usable interfaces.

preprint

REFERENCES

[1] Chadia Abras, Diane Maloney-Krichmar, Jenny Preece, et al. 2004. User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications* 37, 4 (2004), 445–456.

[2] Avrim L Blum and Pat Langley. 1997. Selection of relevant features and examples in machine learning. *Artificial intelligence* 97, 1-2 (1997), 245–271.

[3] R. A. Bolt. 1980. Put-that-There: Voice and Gesture at the Graphics Interface. *Computer Graphics* 14 (1980), 262–270.

[4] Charles C Bonwell and James A Eison. 1991. *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports.* ERIC.

[5] J. Cacace, A. Finzi, and V. Lippiello. 2017. A robust multimodal fusion framework for command interpretation in human-robot cooperation. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN).* 372–377. https://doi.org/10.1109/ROMAN.2017.8172329

[6] Scott Chacon and Ben Straub. 2014. *Pro git.* Springer Nature.

[7] Ming Jin Cheok, Zaid Omar, and Mohamed Hisham Jaward. 2019. A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics* 10, 1 (2019), 131–153.

[8] Philip R Cohen, Mary Dalrymple, Douglas B Moran, FC Pereira, and Joseph W Sullivan. 1989. Synergistic use of direct manipulation and natural language. In *ACM SIGCHI Bulletin*, Vol. 20. ACM, 227–233.

[9] Martin Fischbach, Dennis Wiebusch, and Marc Erich Latoschik. 2017. Semantic Entity-Component State Management Techniques to Enhance Software Quality for Multimodal VR-Systems. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 23, 4 (2017), 1342–1351. DOI: 10.1109/TVCG.2017.2657098.

[10] Chair for Human-Computer Interaction at the University of Würzburg. 2018. *Results of the Multimodal Interface course.* Retrieved September 27, 2021 from http://hci.uni-wuerzburg.de/2018/09/26/mmi-course-results/

[11] Chair for Human-Computer Interaction at the University of Würzburg. 2019. *Results of the Multimodal Interface course.* Retrieved September 27, 2021 from http://hci.uni-wuerzburg.de/2019/09/27/mmi-course-results/

[12] Chair for Human-Computer Interaction at the University of Würzburg. 2021. *Youtube Channel.* Retrieved September 27, 2021 from https://www.youtube.com/user/hciwuerzburg/playlists

[13] Martin Fowler, Jim Highsmith, et al. 2001. The agile manifesto. *Software development* 9, 8 (2001), 28–35.

[14] Epic Games. 2019. *Unreal Engine.* Retrieved September 27, 2021 from https://www.unrealengine.com

[15] Gunther Heidemann, Ingo Bax, and Holger Bekel. 2004. Multimodal Interaction in an Augmented Reality Scenario. In *Proceedings of the 6th International Conference on Multimodal Interfaces* (State College, PA, USA) *(ICMI '04)*. Association for Computing Machinery, New York, NY, USA, 53–60. https://doi.org/10.1145/1027933.1027944

[16] David Heidrich, Chris Zimmerer, Martin Fischbach, and Marc Erich Latoschik. 2021. *Robot Museum.* Retrieved September 27, 2021 from http://hci.uni-wuerzburg.de/2018/06/12/robot-museum-demo/

[17] GitLab Inc. 2012. *GitLab DevOps Platform.* Retrieved September 27, 2021 from https://about.gitlab.com/de-de/

[18] Ed Kaiser, Alex Olwal, David McGee, Hrvoje Benko, Andrea Corradini, Xiaoguang Li, Phil Cohen, and Steven Feiner. 2003. Mutual disambiguation of 3D multimodal interaction in augmented and virtual reality. *Proceedings of the 5th international conference on Multimodal interfaces - ICMI '03* (2003), 12. https://doi.org/10.1145/958436.958438

[19] Matthew J. Koehler and Punya Mishra. 2005. What Happens When Teachers Design Educational Technology? The Development of Technological Pedagogical Content Knowledge. *Journal of Educational Computing Research* 32, 2 (2005), 131–152. https://doi.org/10.2190/0EW7-01WB-BKHL-QDYV arXiv:https://doi.org/10.2190/0EW7-01WB-BKHL-QDYV

[20] Denis Lalanne, Laurence Nigay, philippe Palanque, Peter Robinson, Jean Vanderdonckt, and Jean-François Ladry. 2009. Fusion Engines for Multimodal Input: A Survey. In *Proceedings of the 2009 International Conference on Multimodal Interfaces.* Association for Computing Machinery, New York, NY, USA, 153–160. https://doi.org/10.1145/1647314.1647343

[21] Marc Erich Latoschik. 2001. A general framework for multimodal interaction in virtual reality systems: PrOSA. In *The Future of VR and AR Interfaces-Multimodal, Humanoid, Adaptive and Intelligent. Proceedings of the Workshop at IEEE Virtual Reality.* 21–25.

[22] Marc Erich Latoschik. 2005. A User Interface Framework for Multimodal VR Interactions. In *Proceedings of the IEEE seventh International Conference on Multimodal Interfaces, ICMI 2005.* Trento, Italy, 76–83. https://downloads.hci.informatik.uni-wuerzburg.de/pp217-latoschik.pdf

[23] Marc Erich Latoschik and Henrik Tramberend. 2010. Short Paper: Engineering Realtime Interactive Systems: Coupling & Cohesion of Architecture Mechanisms. In *Proceedings of the 16th Eurographics Conference on Virtual Environments & Second Joint Virtual Reality.* Eurographics Association, 25–28.

[24] Marc Erich Latoschik and Henrik Tramberend. 2011. Simulator X: A scalable and concurrent architecture for intelligent realtime interactive systems. *Proceedings - IEEE Virtual Reality* (2011), 171–174. https://doi.org/10.1109/VR.2011.5759457

[25] Prof. Dr. Marc Erich Latoschik. 2010. *Study Program Human-Computer Interaction.* Retrieved September 27, 2021 from https://mcs.phil2.uni-wuerzburg.de/master/

[26] Prof. Dr. Marc Erich Latoschik. 2021. *Module Catalogue for the Subject Human-Computer-Interaction.* Retrieved September 27, 2021 from https://www2.uni-wuerzburg.de/mhb/MHB1-en-88-g91-H-2021.pdf

[27] Joseph J LaViola Jr, Ernst Kruijff, Ryan P McMahan, Doug Bowman, and Ivan P Poupyrev. 2017. *3D user interfaces: Theory and practice.* Addison-Wesley Professional.

[28] Sascha Link, Berit Barkschat, Chris Zimmerer, Martin Fischbach, Dennis Wiebusch, Jean Luc Lugrin, and Marc Erich Latoschik. 2016. An intelligent multimodal mixed reality real-time strategy game. In *Proceedings of the 23rd IEEE Virtual Reality (IEEE VR) conference.* IEEE, 223–224. https://doi.org/10.1109/VR.2016.7504734

[29] Punyashloke Mishra and Matthew J. Koehler. 2003. Not —what‖ but —how‖: Becoming design-wise about educational technology. In *In Y. Zhao (Ed.), What teachers should know about technology: Perspectives and practices (pp. 99–122). Greenwich, CT: Information Age.*

[30] Ali Bou Nassif, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan. 2019. Speech recognition using deep neural networks: A systematic review. *IEEE Access* 7 (2019), 19143–19165.

[31] Laurence Nigay and Joëlle Coutaz. 1993. A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (Amsterdam, The Netherlands) *(CHI '93)*. ACM, New York, NY, USA, 172–178. https://doi.org/10.1145/169059.169143

[32] Donald A Norman. 1986. *User centered system design: New perspectives on human-computer interaction.* CRC Press.

[33] Sharon Oviatt. 2012. Multimodal interfaces. In *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications, 3rd Edition.* Lawrence Erlbaum Assoc., Mahwah, NJ, 405–430.

[34] Sharon Oviatt and Philip Cohen. 2000. Perceptual user interfaces: multimodal interfaces that process what comes naturally. *Commun. ACM* 43, 3 (2000), 45–53.

[35] Sharon Oviatt and Philip R. Cohen. 2015. *The Paradigm Shift to Multimodality in Contemporary Computer Interfaces.* Morgan & Claypool Publishers.

[36] Sharon Oviatt, Rachel Coulston, and Rebecca Lunsford. 2004. When do we interact multimodally?: cognitive load and multimodal communication patterns. *Proceedings of the 6th international conference on Multimodal interfaces* (2004), 129–136. https://doi.org/10.1145/1027933.1027957

[37] Sharon Oviatt, Björn Schuller, Philip Cohen, Daniel Sonntag, and Gerasimos Potamianos. 2017. *The Handbook Of Multimodal-Multisensor Interfaces, Volume 1: Foundations, User Modeling, and Common Modality Combinations.* Morgan & Claypool.

[38] Sebastian Peters, Jan Ole Johanssen, and Bernd Bruegge. 2016. An IDE for Multimodal Controls in Smart Buildings. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction* (Tokyo, Japan) *(ICMI '16)*. Association for Computing Machinery, New York, NY, USA, 61–65. https://doi.org/10.1145/2993148.2993162

[39] R. Sharma, V. I. Pavlovic, and T. S. Huang. 1998. Toward multimodal human-computer interface. *Proc. IEEE* 86, 5 (May 1998), 853–869. https://doi.org/10.1109/5.664275

[40] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems.* 2951–2959.

[41] Leonard Springer, Mary Elizabeth Stanne, and Samuel S Donovan. 1999. Effects of small-group learning on undergraduates in science, mathematics, engineering, and technology: A meta-analysis. *Review of educational research* 69, 1 (1999), 21–51.

[42] Unity Technologies. 2017. *Unity.* Retrieved September 27, 2021 from https://unity3d.com/

[43] David R Thomas. 2006. A general inductive approach for analyzing qualitative evaluation data. *American journal of evaluation* 27, 2 (2006), 237–246.

[44] Virtual Reality Toolkit. 2020. *Unity.* Retrieved September 27, 2021 from https://vrtoolkit.readme.io

[45] XR Interaction Toolkit. 2020. *Unity.* Retrieved September 27, 2021 from https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@0.9/manual/index.html

[46] Dennis Wiebusch, Chris Zimmerer, and Marc Erich Latoschik. 2017. Cherry-Picking RIS Functionality – Integration of Game and VR Engine Sub-Systems based on Entities and Events. In *10th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS).* IEEE Computer Society.

[47] Erik Wolf, Sara Klüber, Chris Zimmerer, Jean-Luc Lugrin, and Marc Erich Latoschik. 2019. "Paint that object yellow": Multimodal Interaction to Enhance Creativity During Design Tasks in VR. In *Proceedings of the 21st ACM International Conference on Multimodal Interaction (ICMI '19).* New York, NY, USA, 195–204.

[48] Chris Zimmerer. 2016. *Big Bang.* Retrieved September 27, 2021 from http://hci.uni-wuerzburg.de/2016/10/11/planetarium/

[49] Chris Zimmerer, Martin Fischbach, and Marc Erich Latoschik. 2016. *GIB MIR Project page.* Retrieved September 27, 2021 from http://hci.uni-wuerzburg.de/2016/02/01/quest-v2/

[50] Chris Zimmerer, Martin Fischbach, and Marc Erich Latoschik. 2018. Semantic Fusion for Natural Multimodal Interfaces using Concurrent Augmented Transition Networks. *Multimodal Technologies and Interaction* 2, 4 (2018), 81.

[51] Chris Zimmerer, Martin Fischbach, and Marc Erich Latoschik. 2018. Space Tentacles - Integrating Multimodal Input into a VR Adventure Game. In *Proceedings of the 25th IEEE Virtual Reality (VR) conference.* IEEE, 745–746. https://downloads.hci.informatik.uni-wuerzburg.de/2018-ieeevr-space-tentacle-preprint.pdf

[52] Chris Zimmerer, Martin Fischbach, and Marc Erich Latoschik. 2021. *GIB MIR Project page.* Retrieved September 27, 2021 from https://www.hci.uni-wuerzburg.de/projects/mmi/

[53] Chris Zimmerer, Erik Wolf, Sara Wolf, Martin Fischbach, Jean-Luc Lugrin, and Marc Erich Latoschik. 2020. Finally on Par?! Multimodal and Unimodal Interaction for Open Creative Design Tasks in Virtual Reality. In *Proceedings of the 2020 International Conference on Multimodal Interaction (ICMI '20).* Association for Computing Machinery, New York, NY, USA, 222–231.

preprint