Institute of Visualization and Interactive Systems

University of Stuttgart
Pfaffenwaldring 5a
D–70569 Stuttgart

Bachelorarbeit Nr. 119

Exploring the Design Space of Programming by Demonstration with Context Aware Assistive Systems

Matthias Hoppe

Course of Study:	Informatik
Examiner:	Prof. Dr. Albrecht Schmidt
Supervisor:	Dipl.-Inf. Markus Funk, Dipl.-Inf. Sven Mayer, M.A. Oliver Korn
Commenced:	April 1, 2014
Completed:	September 30, 2014
CR-Classification:	H.5.2

Abstract

Due to the recent development in head-mounted displays, projection systems, rising capabilities of Augmented Reality and availability of Kinect for depth images, the advance of assistive systems in industrial context began. We propose a system that not only provides methods to assist industrial workers in their everyday tasks by assisting and providing instructions and giving feedback about the executed tasks. But although present a system that provides the capability to record and therefore create instructions by demonstration. This tackles the problem of current assistive systems and the complexity editors that are used for creating these instructions. We conducted a study that involved the creating of instruction via different conditions that were performed by experts in the field of manual assembly. Furthermore we verified these instruction with 51 industrial workers that completed assembly tasks guided by the instruction. Our results indicated that interactive instructions created through Programming by Demonstration are equal to existing approaches. Additional qualitative feedback showed that instructions through Programming by Demonstration are generally well perceived.

Kurzfassung

Durch die jüngsten Entwicklungen im Feld der Head-Mounted Displays, Projektions Systeme, wachsende Möglichkeiten bezüglich augmentierter Realität und der Verfügbarkeit von Tiefendaten durch die Kinect Kamera, finden Assistenzsysteme immer mehr Anwendung in industriellen Bereichen.

In dieser Arbeit wird ein System vorgestellt, welches nicht nur die Möglichkeit bietet, Arbeiter in der Industrie bei ihrer täglichen Arbeit, durch Anzeigen von Anleitungen und Feedback über ausgeführte Arbeitsschritte, zu unterstützen. Des Weiteren wird ein System vorgestellt das ermöglicht, Arbeitsanleitungen durch Demonstration zu erstellen. Hierdurch wird das Problem aktueller Assistenzsysteme beseitigt, Anleitungen mit komplizierten Editoren erstellen zu müssen.

Mit einer Studie wurde das Erstellen von Anleitungen über verschiedene Ansätze überprüft. Die Studie wurde mit Experten der manuellen Montage durchgeführt. Weiterhin wurden die erstellten Anleitungen von 51 Arbeitern aus der Industrie getestet, welche die Anleitungen zum erfüllen eines Arbeitsvorganges genutzt haben.

Die Ergebnisse weisen darauf hin, dass interaktive Anleitungen, die per Programmierung durch Demonstration erstellt wurden, gleichwertig gegenüber herkömmlichen Anleitungen sind. Die Auswertung des qualitatives Feedbacks der Studie hat zusätzlich gezeigt, dass Anleitungen die per Programmierung durch Demonstration erstellt wurden, generell als positiv wahrgenommen werden.

Acknowledgements

A big thank-you goes out to Klaus Klein, Michael Spreng and the Audi AG for giving us the possibility to conduct our study with professional workers from the field, offering great assistance and a perfect organisation in every way, to enable us a trouble-free research study!

Also we would like to thank Schnaithmann Maschinenbau GmbH for building the experimental setup and Andreas Bächler from Hochschule Esslingen for the great cooperation.

Contents

List of Figures

1 Introduction

With the wide spread popularity of computers, tablets, smart phones and other technical devices and increasing inclusion in our everyday life. Programming and creating digital content that ranges from simple Internet sites over customising applications appearance of a smart phone to complex industrial installations, get more important. More and more computers in all different kinds of shapes and applications come into our lives, most of them are designed to make our lives easier. However, through the complexity and volume of different systems and their diversity in functions and structure. Knowledge, experience and training is needed in order to operate or even customise these applications and computers.

Writing code by using programming languages like Java, C or C++, are the conventional way to create a piece of software. Even though these are "simplified" and more comprehensible ways than writing pure machine code, they are still complex and hard to comprehend for a inexperienced user and need quiet some time to get into and being able to work with. Therefore simpler alternatives, with better usability are needed. In Chapter 2 we will present different approaches of already existing projects and new concepts, that could be an alternative to conventional programming.

Especially in the industry many manufacturers are interested in using new technology to improve their product quality and production process, by enhancing training or aiding worker with their everyday tasks. Light Guide Systems uses projections on the working area, to instruct the user by presenting him information about his current working task. This information could be, what part needs to be picked from boxes or how the user has to assemble objects. [ops]

This is a great way to support the user with presenting information about his work tasks. However the system is not able to detect if the user is performing the right tasks that are needed. The ASLM project [1] provides similar functions as the Light Guide System, but also showed, by proof of concept of a workflow, that it is possible to supervise the worker and detect if a wrong assembly was executed or a part was picked from the wrong box.

We want to improve this concept and propose a system that makes the supervision highly adaptable to all kinds of workflows, by using object recognition and change of depth data to detect an assembly or withdrawal from boxes.

[1] http://www.hs-esslingen.de/de/hochschule/fakultaeten/maschinenbau/forschung-und-transfer/forschungsprojekte/forschungsprojekt-aslm.html

The current used and researched systems do not only need to be created and tested by professionals, but also processes that are in use need to be customised and updated to represent the latest processes or modified models. These tasks need do be manageable by workers of the appropriate areas, that possibly don't have training in programming or creating these systems. Because of this the systems have to be easily accessible and usable by inexperienced users without extensive training in the field of programming.

There are plenty of efforts to make these interactions simple and easily accessible for a broad audience. In this paper we will present different approaches to improve the usability and enable an easier access by inexperienced users. This does not only mean making writing code simpler but also trying to discard the need to write code but rather "make" code by demonstration or create code automatically by using user behaviour to detect the users needs and intentions. We focus on the approach to achieve this is programming by demonstration (also called programming by example). This is probably the approach that feels closest to the goal of being able to show the computer what to do. (Chapter 1 [you01])

Three of the basic ideas behind these systems, that realise Programming by Demonstration are inferencing systems, that are able to "guess" what the users intentions are, systems that uses heuristics in order to determine what the user wants to do or defined rules, that are created when the application is being programmed. Predefined rules are used in most demonstrational systems that are often used for special areas of application. (Chapter 3.1 [you01])

We will use Programming by Demonstration to create instructions as a fast process, that is not a stressful and exhausting task for the user. We want to lighten the burden of creating instructions with highly complex and time consuming editors, that are state of the art with current systems. [ops][KSHK12]

Our concept of using Programming by Demonstration to create instructions automatically, is one of the developments that are part of the motionEAP project. motionEAP is a research project and is government-funded by the German Federal Ministry of Economics and Technology, in cooperation of several big industrial companies and research parters [2].

1.1 Current Technical Development

In this section we will present a short overview of the current technical development of interaction methods that led us to the setup we use in our project (for example Kinect). As well as presenting alternative interaction methods that could be used in private or industrial cases of applications in similar systems.

[2] http://www.motioneap.de/partner/

With the increasing popularity of smart phones and several head mounted displays, like Google Glass, Oculus Rift, and Sony's Project Morpheus, that are about to be released, Augmented Reality and Virtual Reality also got a growth in popularity and potential applications. Even though Augmented Reality and Virtual Reality, and other varieties from the field of mixed realities, are nothing new for people that are interested in technology. More and more people get in touch with these technologies [3]. This is probably caused by the rise of smart phones, that have a strong processor, a screen and a camera. Additionally, GPS (Global Positioning System) and accelerometers enable extra functions for orientation and location detection. Because of this availability more developers had the chance to create application that are accessible by everybody that owns a smart phone, without investing extra money in new additional devices.

Popular applications for smart phones are navigation systems that not only allow the usual usage of a car navigation system, but some also allow to use the camera, accelerometers and the GPS of the smart phone to point to a building or a street and get on-screen information about restaurants or street names. The information that is given to the pedestrians about their surroundings are displayed on the location via Augmented Reality. This allows a direct, immersive and easier navigation than a top down map where the users location and the target location are marked with dots. [1] [2]

New interaction and input methods got popular and got more public attention with the rise of camera systems for consoles like Sony's Eye Toy for Playstation 2 [3]. Even more gamers got in touch with these systems when the Kinect for Xbox360 got released [4]. The Kinect 3.1b is a camera modul by Microsoft that first got released in November 2010 for the XBox 360 gaming console, and is used as a motion sensing input devices in order to play and control games with the users full body. Many hobby developers had a big interest in using the Kinect for other purposes and explore new ways of interaction and input method. Microsoft later released the Kinect software development kit for Windows 7 on June 16, 2011 [5]. On February 1, 2012 a PC version got released [6].

It offers a RGB camera that stores three channel data in a 1280x960 resolution (see 3.2a). This can be used to store colour pictures and detect objects. We use the infra-red (IR) emitter and IR depth sensor to scan the structure of objects and detect movement. "The emitter emits infra-red light beams and the depth sensor reads the IR beams reflected back to the sensor. The reflected beams are converted into depth information measuring the distance between an object and the

[3] http://www.technologyreview.com/news/428654/augmented-reality-is-finally-getting-real/
[1] http://www.windowsphone.com/de-de/store/app/here-city-lens/b0a0ac22-cf9e-45ba-8120-815450e2fd71
[2] https://play.google.com/store/apps/details?id=com.trackyapps.street_lens
[3] http://de.playstation.com/ps2/accessories/detail/item51693/EyeToy-USB-Kamera/
[4] http://www.xbox.com/en-US/Kinect
[5] http://research.microsoft.com/en-us/news/features/kinectforwindowssdk-022111.aspx)
[6] http://blogs.msdn.com/b/kinectforwindows/archive/2012/01/09/&kinect-for-&windows-commercial-&program-announced.aspx

sensor. This makes capturing a depth image possible." The integrated multi-array microphone and 3-axis accelerometer is not in use in this current setup. The microphone can be used to record sound or detect the location of a sound source. It is possible to detect the current orientation of the Kinect via the 3-axis accelerometer. [7] We use the functions that the Kinect offers to realise object recognition and enabling touch gestures that get detected by the depth camera. [HA12]

After many failed attempts, dreams of 80s and 90s gamers and technical enthusiastic people, to create a cyberspace with virtual reality head-mounted displays got possible with devices like Oculus Rift [8] or Project Morpheus [9] on the rise. These enable immersive projections by using head-mounted displays as well as natural feeling interactions by registration of head movements and angles.

New input and interaction methods like these are not only interesting for gaming, but also for other aspects of our daily life's, for private use or even industrial systems that can support us in our working tasks. Many projects research new ideas to make our everyday lives easier.

Wear Ur World[MMC09] is a camera-projector system that allows the user to use a projector that is mounted on the chest, to view information. The wearer is also able to control and interact with the projection by using gestures or pointing with a finger. These can be recognised by the camera that is also mounted on the chest.

These projector system help to view information and allow a hands free way to view information without having to take a device like a smart phone out of a pocket. In some cases, like entering a password or working with other sensible data, this still leaves the problem that a new input method is needed to interact with the device in a more private way. Many different research projects develop and test hands-free input systems for mobile devices that don't use voice input, pointing or making gestures in a public space.

Skinput is an approach to use body parts (for example an arm) to control devices like smart phones via bio-acoustic. This has the advantage that a person always knows where his arms, elbow and fingers are, and therefore can accurately touch them without having to look where they are located [HTM10].

There are not only new input ideas for projector systems but also for devices like smart watches. Smart watches are a good representation of how it is possible to improve input on a devices that offers little space for interaction, like touch gestures or on-screen keyboards. Adding elements like strings that are retractable from the device can be used to increase the interaction space without increasing the actual size of the device. [BNF06]

[7] http://msdn.microsoft.com/en-us/library/jj131033.aspx
[8] http://www.oculus.com/
[9] http://www.officialplaystationmagazine.co.uk/tag/ps4-oculus-rift-vr-headset/

All these research projects show the current thrive for new interaction methods via input and presentation. These interesting new ways to interact with systems and also offer a wide variety of possibilities that can be used in wide areas of industrial processes that are currently limited.

1.2 Structure of the Thesis

The structure of this thesis is described in the following paragraph. We start by showing different approaches and systems in the related work section. The related work covers different concepts of trying to making programming easier by introducing new approaches like, block-based programming with Scratch or Lego Mindstorm, optical tracking of objects or persons. We also examine difficulties of defining conditions and show different approaches and applications of Programming by Demonstration. In the last part of the related work, we discuss how Augmented Reality can be used in industrial processes.

In the concept section we will present the current state and the basic setup of the motionEAP project and explain the used components. The theoretical model depicts how the structure of a work process has to be defined in order to create an instruction by demonstration. We will discuss how we define a work step, what we use as indicators for these single steps and what we can use to trigger these steps in our system. We also explain how feedback is used and how instructions can be created by demonstration and present how an instruction is used to teach and show the recorded working steps. In the final section we will summarise the treated subjects and will give a outlook on the future work.

Outline

This thesis is structured as follows:

Chapter 2 – Related Work: In this chapter we present concepts that influenced us.

Chapter 3 – Concept: The current setup of the motionEAP project and its components are presented.

Chapter 4 – Theoretical Model: In this chapter we present how we define working steps and create instructions.

Chapter 5 – Study: In this chapter we present the research study and results.

Chapter 6 – Conclusion and Future Work concludes the results of the thesis and will present future work.

2 Related Work

The commonality of all the approaches that we present is that they try to make it easier, for end users with little programming experience, to automate tasks or set up actions in case of a special condition.

We classify programming in four major categories. Writing Code is the usual approach that takes practices and experience and is not suited for the most users due to its complexity and high learning effort. To simplify programming, predefined items like events, conditions and actions, can be used to allow the user to select and set up his program from these items, that can be connected to each other (e.g. Scratch). Context monitoring is used to automate tasks by letting a program record events and actions. By analysing these data, the program can create automations or give the user options from which he can select and configure automations afterwards (see "Keep Doing what i just did" [MB13]). Additionally we see Programming by Demonstration as an option to let inexperienced users create code by not only selecting from options that are given but rather show the system what he wants to create and let the system thereby automatically create the code for him.

Making programming more accessible is an important factor to aid users in private and industrial tasks. In order to integrate our concept of creating instruction by Programming by Demonstration we need ways to detect and record the instructions that the user gives the system. For replaying these instructions we also need a way to present the recorded information and supervise the executed tasks of the user.

In the next sections we will look into systems that approach ways to detect and record activities and in what ways additional data in the context of his current task can be presented to the user with the use of augmented reality. We saw, head-mounted display or projection systems that could be used to provide instructions at already existing working stations, as the most suitable options for industrial applications.

2.1 Programming with Predefined Items

One of the hardest things for new and inexperienced users to do, while programming by writing code, is probably to visualise what the written code will perform. Even the simplest Java code that records an input of the user (entering a phrase) and shows this input on the screen, consists of a considerable amount of lines, that can be really confusing and unclear to a new user. [ML07]

An alternative to writing code the traditional way, is using predefined items (e.g. graphical elements, like blocks) that represent implemented code elements. These blocks can be connected to each other.

This representation of code makes it easier to visualise what the code will produce and therefore teaches new users the basic structures of programming without an deterrent overload of syntax.

Scratch [scr] is a graphical programming language developed by MIT's Media Lab and is a way to create simple animations of characters, that serve as an demonstration. These consist of different modules (that are represented by coloured blocks) that control the movement of the character, control structures and so on. The module "move X steps" (where X can be replaced by a given number) can be placed under a module like "Say [enter phrase] for Z seconds". [ML07] If executed, this sequence then shows an animation of the character saying something (with a speech bubble appearing) and then walks away X steps.

A study [SD03] with context-aware applications showed that the most common used structual idea for programming are "if-then"-statements. "Practical trigger-action programming in the smart home" [UMPYHL14] introduces the concept of trigger-action programming, which enables users of a smart home to program simple "if [TRIGGER] then [ACTION]" automations, by selecting different triggers and actions, which "... can express most desired behaviours..." [UMPYHL14]. An example of this is "If it is 6pm, then turn the lights on."

A similar concept is provided by Lego Mindstorms. There are many different systems and modifications to select from. Each system has a different difficulty. It ranges from drag and drop different blocks (similar to Scratch), writing simple code like bob.moveForward(5) (which moves the robot named bob 5 cm forward) up to systems that provide the possibility to control the rotation of each single motor. [SM02]

We used this idea of simplifying programming structures to implement predefine working steps or actions like, picking a part from a box, to create code structures that later can be easier selected with an editor or automatically with Programming by Demonstration.

2.2 Context Monitoring

Aside from possible privacy issues due to the recording of everything that happens, context monitoring provides an easy way with little work for the user to generate tasks that the user wants to be automated.

Smart phones are a good and by now widespread tool that makes all kind of context monitoring fairly accessible. Apart from the obvious sensors like a microphone, accelerometer to be able to sensing the position of the phone and GPS (Global Positioning System) to locate the phone, they provide the ability to connect new sensors and tools via Bluetooth or WiFi. The operation system can also be modified with different applications to record whatever data is needed. [LJM+12]

"Keep Doing What I Just Did: Automating Smartphones by Demonstration" [MB13] is an approach that helps the user to set up automation by discovering usage patterns. For this actions (e.g. applications that the user starts), conditions (e.g. time of the day, GPS location) are recorded. From this recorded data and behaviour "Keep Doing It" generates a recommended task automation. There are even more possible automations generated, from which the user can select, in case the recommended automation is not what the user was looking for (see Figure 2.1).

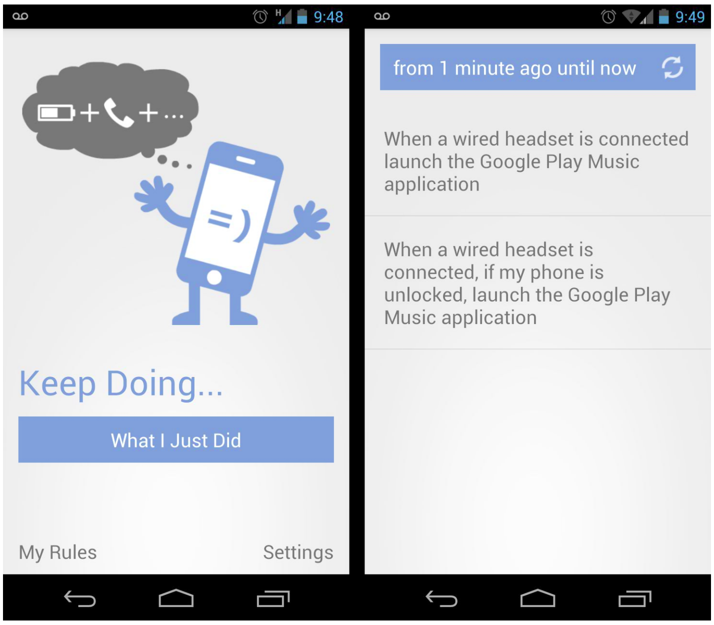


Figure 2.1: Interface and recommendation list of Keep Doing It (image taken from [MB13]).

One example is that the user leaves his home everyday at a 8:00 AM, connects his headphones and starts his music player. The algorithm then can generate a possible rule, when the headset is connected at 8:00 AM, the music player is started and it could give the alternative that the change of location (detected via GPS) and plug in in the headphones are the trigger for launching the music player.

A study [KVVM04] that tested different interactions with a smart homes (via mobile phone, computer, media terminal) showed that the participants "...ranked automation as the most wanted interaction technique." On the other hand it shows that "...even though automation could be context-aware, subjects felt they are not ready for it, for they fear that technology does not understand the various functions in the home."

"Ambient Intelligence" [GRSP+12] provides a system that is aimed to support users with their activities by reacting to the location of the user by recognising objects and user behaviour with visual sensors. It is possible to locate persons with such systems, but recognising activities is a bigger problem. Especially with a group of people identifying certain situations can be challenging.

One of the biggest problems with context monitoring is to decide what context is being monitored at the moment and to identify situations. "aCAPella" [DHB+04] uses a business meeting as the situation that is being taken as an example to show the difficulties of define a situation. The first problem is, to identify how various people define as an meeting. Most of the time it is clear that 5 people in a room that are talking to each other, are in a meeting.

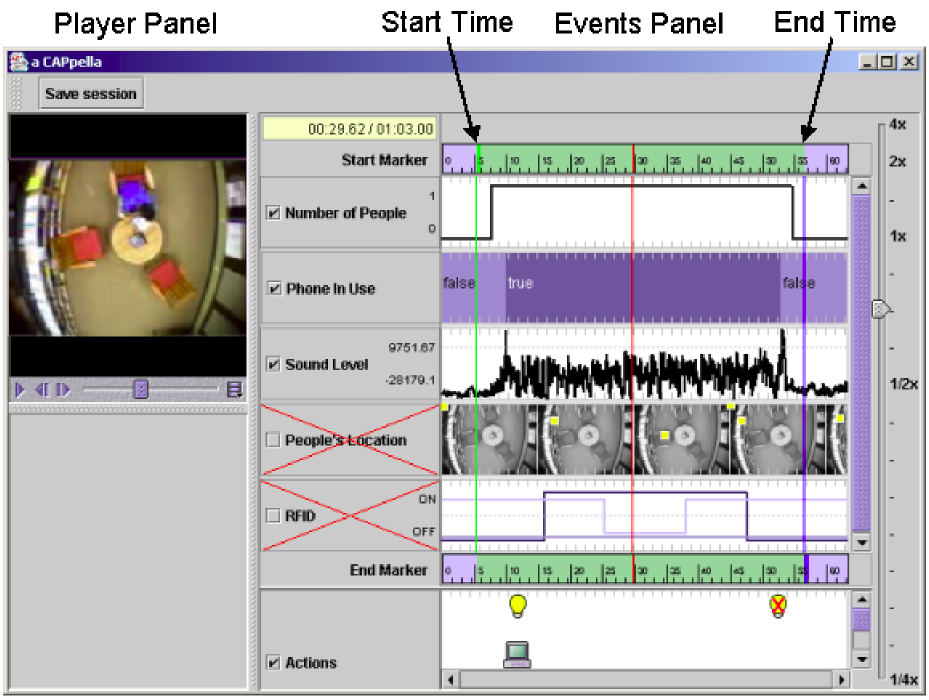


Figure 2.2: aCAPpella interface that shows different recordings of the meeting scenario and lets the user, for example, select start and end time (image taken from [DHB+04]).

But it is not that clear in all the cases. Some would understand two talking person in a room as a meeting, but maybe they are both on the phone and each talking to a different person. And for some it would be a meeting if one person and is talking to somebody else by using a laptop. Thus a system that is aimed at automating a setup for meetings (dimming the light, turning on the projector, etc.) should be able to detect all possible scenes. It is hard to program every possible scenario that could happen.

This is why aCAPella's system, records everything that happens in the meeting room (number of persons, time, possible talking, number of phones used). The user then can select from multiple examples and indicate the given time and the important events that are important and are indicating a meeting (see Figure 2.2). This proceeding is used to improve the system and help learning and improving the definition of given proceedings (e.g. a meeting). The user has to repeat this process several times, in order to improve aCAPella's ability to recognise this kind of event.

In order to realise such a context-aware system, it has to have multimodal sensing capability and a background model (of the room) needs to be created, to differentiate a scenario where people and movements are in the same room, later on. This includes recordings of different light settings and exposures of various sunlight. The aCAPpella system can be very flexible in its selection features and possibilities, but apparently a informal test showed that to much flexibility may be too complicated.

Aude Billard et al. [BCDS07] presented different methods of machine learning for robots and distinguish low and high level representations. This led us to the conclusion that recording the users action over a certain time, or tracking and evaluate every action the user performs, is not suitable for what the want to achieve and that detecting high-level actions are sufficient to define the tasks and working steps that the user will perform during an assembly.

2.3 Programming by Demonstration

Another approach to help a technical inexperienced user to work with a system is programming by demonstration. This basically means that the user is able to just show, and hereby teach the system, what he or she wants to do and the system can transform this demonstration in executable code. There are a variety of approaches to achieve this difficult goal. But if the system is reliable and able to interpret what the user wants to achieve, then it would probably be the easiest and most simple way for a technical inexperienced user to program.

The requirement for writing code is widespread in all kinds of areas, and programming by demonstration is therefore not only used in processing and creating text documents. "Gesture coder: a tool for programming multi-touch gestures by demonstration" [LL12] presents a way to create new multi-touch gestures, not by writing the complex code, but by demonstrating the gesture a few times. The application then automatically generates the code. The user has the possibility to test the gesture and where necessary add additional examples. The produced code then can be added to the users application.

In order to making the creation of instructions easier, we looked into macros. Macros are a way to lighten the workload of users regarding repetitive tasks. To create these, the user needs to have some sort of programming skills and therefore macros would demand time of the user, by first learning to create and program macros. Even when using Programming by Demonstration to create macros, the user is often required to think ahead and anticipate what tasks he will have to repeat in the future, and therefore start and stop some kind of recording. Due to this increased mental task load the user likely just repeats the task rather than creating macros [SK96].

"Simplifying macro definition in programming by demonstration" [SK96] shows two approaches to reduce the users burden of macro creation with the application "DemoOffice". This is done by recording all of the users actions and then using "action slicing" or "macro auto-definition". "Action slicing" creates macros by extracting the users activities that had an influence on created data. The user can simply select data (by drag and drop). The "macro auto-definition" detects the users actions which probably will be performed again. The system automatically creates new macros due to this data. The user can select from the given macros, and is shown an explanation of what the macro will do, as well as an example of the actions the macro will execute. If the macro will perform the right task, the user then can activate it by simply clicking on it.

Using macros to execute a particular routine or code are not the only possibility to simplify repetitive tasks. Templates are used to aid inexperienced user and lighten their workload by supplying them with a basic structure that is usually created by experts and contain the logic, and semantic of applications or simple document forms. Later on inexperienced users can fill out blanks on these templates to accomplish their task. These can be Microsoft Excel templates that already contain the logic to calculate desired results, and only the input data has to be entered. Likewise templates for Power Point or other presentation software, reduce the workload for an inexperienced user by offering him a complete presentation where only the text and pictures need to be added at the corresponding blank.

Such templates are not only used in offices that process insurances but also for astons on TV or other multimedia areas. And for these areas end-user development is required to ensure that technical inexperienced users can author content on different multimedia fields. [ASA+13]

2.3.1 Creating Documentations and Instructions

It is a time consuming task to create documentations or instructions and Microsoft Word is still one of the most commonly used text editing tools, because of this Madhu Prabaker and Lawrence Bergman proposed a program called DocWizards that uses Programming by Demonstration to author documentations.

DocWizard [BCLO05] is able to capture and replay events from a GUI that the user executes. By doing so, it supports to automatically and incrementally create documentation through multiple demonstrations. Doc Wizards records any interaction with the GUI of a given application and captures changes that are caused by those interactions. The system is also able to incrementally

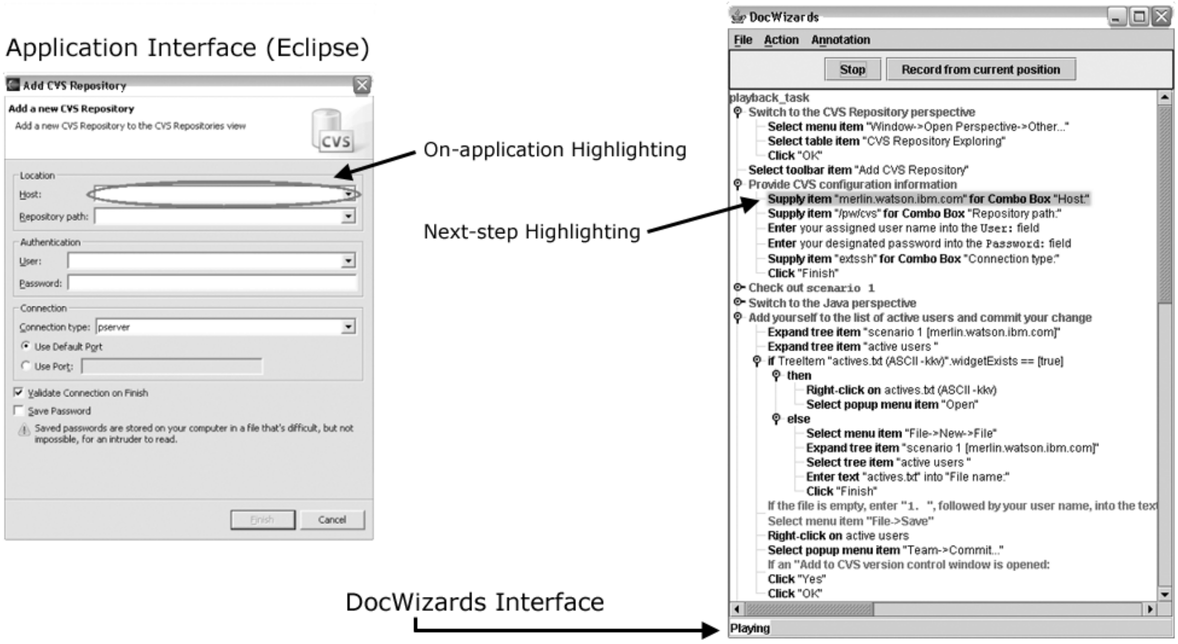


Figure 2.3: DocWizard Interface that shows the highlighting of the next step that is prompted to the user, both on the GUI (left side) and in the text (image taken from [PBC06]).

update by these observations. To process multiple demonstrations DocWizards needs to identify and differentiate these and then introduces if-then-else statements, that are included in the documentation, to explain possible differences. The author also has the option to provide and include additional information by adding comments to a desired step.

DocWizards supports the user by highlighting the next step in the document text and the GUI parts that are included in the next step (see figure 2.3). The authors note that there a two even groups of users that create a documentation. One group is called "immediate annotators", which prefer to add comments and create hierarchical groupings of the steps, immediately after demonstration the corresponding steps. The other group is called "delayed annotators", this group rather stays in one mode (recording or annotating) for a longer time, to record all the steps first and annotate afterwards. Furthermore it is noted that users of the second group sometimes made a vocal mentioning of wanting to annotate something afterwards, but forgot to perform this action in the end. This shows that there should be support for a variety of working and annotation styles for users. For example it would be a good idea to provide a mechanism for reminding delayed annotators to perform important annotations that they otherwise possibly could forget. DocWizards puts attention of immediate feedback for the authors in record mode. The documentation gets updated immediately after recording each task and hereby provides the author the ability to observe the recordings and reflect upon the tasks in real time.

The concept of listing instruction while they are being recorded live on the GUI and marking steps in the instruction list while highlighting corresponding tasks on the GUI are a suitable way to record and present instruction in an assembly process. Due to this we decided to separate the feedback in working instructions of the current step that needs to be executed, are highlighted on the working surface. An overview of whole process is provided on the GUI on an external monitor, in case additional feedback is needed.

In most areas of production in the industry or for assembly in private aspects of furniture, even for disassembly due to cleaning or maintenance purposes for example in a medical field [RHM$^+$13], instructions are an essential part of a process. For these tasks, instructions need to be created. In many fields instructions are manually written down or pictures of single steps are taken. Depending on the field, instruction videos of the assembly are shot.

The state of the art concept for creating digital instruction are graphical editors [ops][KSHK12]. These are very complex and time consuming ways of creating instructions that need plenty of training and are error-prone.

Therefore we present a system that provides an easy way of creating instructions for assembly processes in an industrial application.

2.4 Augmented Reality

Augmented Reality allows to extend the reality with virtual elements in order to present additional information, blended in with the real world. This can be achieved in many different ways through various senses, but is often used as a term of visual augmentation. Augmented Reality is not only used in TV broadcasting of sporting events (showing distance lines blended in with the field) but also in other areas [VKP]. The application of Augmented Reality in the automotive industry is various. It can be used to check the physical produced parts with their virtual counterparts. In order to achieve this the virtual part has to be located at the same position as the real part and the overlay of the virtual part of the construction data has to be transparent. [NK06] As stated by Caudell and Mizell [CM92], tasks in the fabrication and assembly in aerospace industry are mostly complex processes that need involvement of human workers. Concerning this matter head-mounted display could improve the efficiency of human workers. A collaboration of Metaio and Volkswagen was established to improve industrial processes by including and enhancing new Augmented Reality aspects. This ranges from concept planning, arranging production lines, variance comparison to part verification. [PBDM07] Augmented Reality can not only be integrated in the industrial production process but also used as a training device or for assistance in maintenance [RHM$^+$13]. Especially in areas where good training and large amounts of data and documentation are essential.

With augmented reality the user could have easy and fast access to the documentation that is needed in the current working step. This is even more helpful when the data is widespread on different mediums (internet, CD-Roms, books, etc.) or when it comes to tasks that are performed infrequent. [SLSG03]

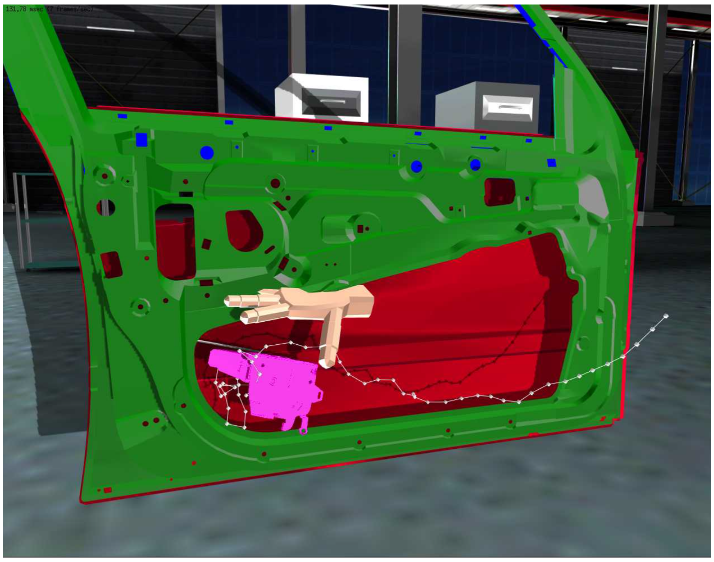


Figure 2.4: Virtual Demonstration of the door lock insertion (image taken from:[RSKM98])

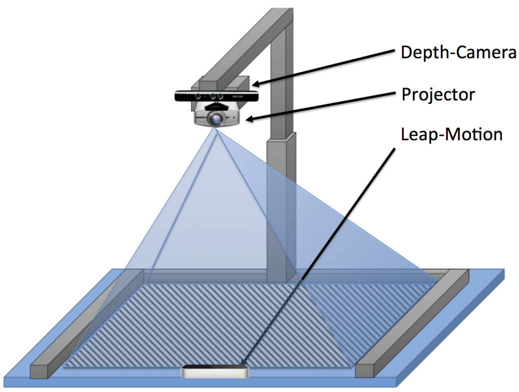
Digital version of most parts are already existent in the aerospace and automotive industry and therefore three-dimensional models are accessible at a low cost. These models can be used to give instructions to a worker that is wearing, for example, a head-mounted display that shows a three-dimensional version of a door lock that has to be inserted into the car door (see Figure 2.4). For tracking simple markers can be placed on the door. Animations of the insertion of the door lock, then show the worker where the part has to be placed and where he has to hold it in order to be able to place it in the tight space without getting stuck. [RSKM98] This is especially helpful with fast changing products or a production line with high range of variations. At an augmented workplace, augmented elements can not only help by presenting the user additional information via a head-mounted display but also by giving him the opportunity to implement objects in his work task. With an camera projector system the user has the possibility to assign different functions to his everyday objects. The user can scan an physical object that then can be assigned to a function. For example, a bottle or a bottle cap can be used as an rotary knob, or an pen can be used as the control module of an slider, in order to change the volume or brightness. [FKS14b] We took these systems as an inspiration to improve production processes by providing feedback via Augmented Reality to a worker that uses our projector based assistive system. This provides the worker with the advantages of instruction via Augmented Reality, by removing the need for wearing possible disruptive devices like a head-mounted display.

3 Concept

There are many concepts and systems that incorporate Augmented Reality into production or industrial processes, but Augmented Reality is often perceived as visual concept that is implemented with head-mounted displays or mobile phones.

Funk et al [FKS14a] present a different concept to use Augmented Reality. Therefore an ordinary working place, like a table or a industrial assembly station is used and extended with a camera projection system that is fitted to identify objects and work processes, but also has the capability to present the user with in-situ feedback and project additional information.

The camera can detect movement and variations in the depth image (see Figure 3.2a) as well as colour images (see Figure 3.2b. The projector can prompt information on the surface (e.g. table or more complex structures) and on the boxes on the backside of the table. If wanted the worker can interact with the information projection on the surface by multi-touch gestures, due to the capability of detecting fingers and touch events (small distance to the surface). It is possible to recognise and save objects with their size, shape (top down view) and position (see Figure 3.2c) and therefore save different working steps and their variations in depth data (see Figure 3.2d).



(a)

(b)

Figure 3.1: (a) Basic setup of the motionEAP table.[FKS14a] (b) Kinect verison for XBox 360[a]

[a] http://commons.wikimedia.org/wiki/File:Xbox-360-Kinect-Standalone.png

This setup is well suited for typical workstations in the manual assembly. Such stations or a tables are often equipped with top mounted tools (hanging from the top and can be pulled down) and several rows of boxes on the backside of the workstation, where different parts are stored. Due to the projection possibilities of the system the worker can be instructed and supervised by the system. The system will project a field on the box from which parts have to be picked. If the worker reaches into the box the Kinect camera can detect due to the depth image if parts got picked from the right box and can prompt the worker with a corresponding feedback. The amount of information of each working step can be varied regarding the skill, error rate and preference of the worker by detecting the time that is elapsed during a working step and the amount of errors (e.g. picked part from the wrong box).

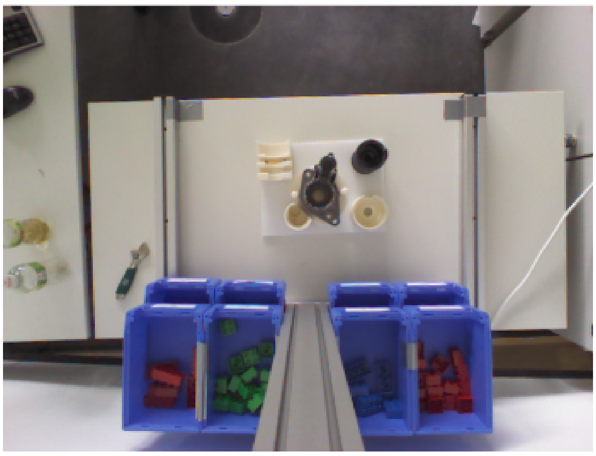
The concept that we suggest is creating instructions by demonstration. This is realised by using the components and capabilities of the assistive system. This is a simple way of creating instructions, that needs little training and is relatively safe when it comes to errors.

3.1 Setup

The camera projector system 3.1a of the motionEAP project is highly portable, lightweight and cheap. It consists of a Kinect-sensor, that provides colour and depth image that are used for object recognition, interaction and movement detection and touch detection with the surface. A conventional projector is used to project information like instructions or feedback, highlighting of the next working step or pictures. Additionally a Leap Motion can be added to the setup to enable interaction with three-dimensional gestures.

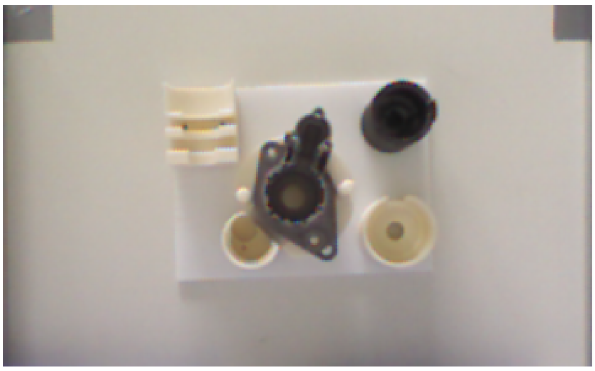
On the research-prototype a metal frame is used to hold the components in a adjustable height above the table. The frame was usually placed on top of a ordinary table. The Kinect is mounted next to the projector on top of the rig. If desired the position and angle can be easily adjusted.

Due to this simple setup, the system is not only variable in its prototype setup but can also be modified for different needs and field of application. The head of the frame, that contains the Kinect and the projector, can be easily mounted on an already existing assembly station, and thereby be adjusted to most working places and demands.

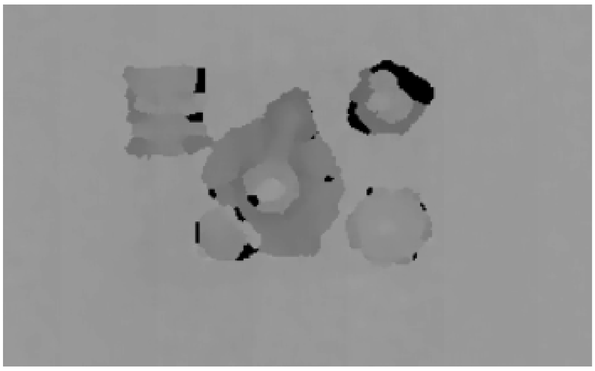


(a)

(b)



(c)



(d)

Figure 3.2: Views of Kinect: (a)RGB Image (b)Depth Image (c)Cropped RGB Image that shows the assembly area (d)Cropped and smoothed depth image of the assembly area

4 Theoretical Model

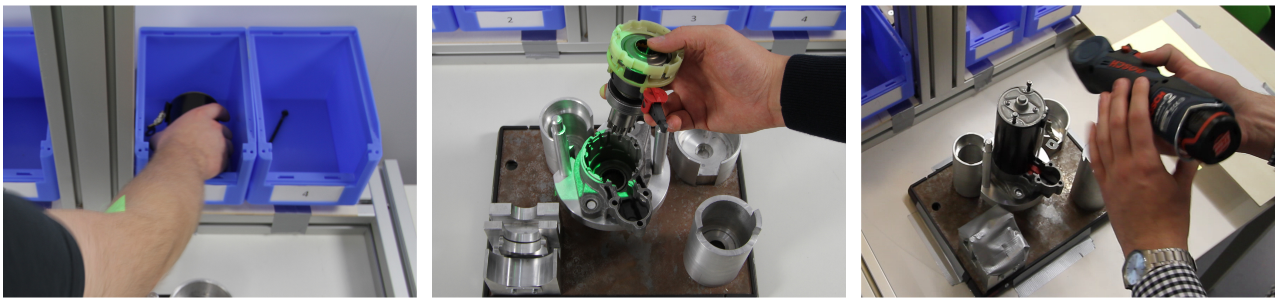
Our Theoretical Model consists of multiple components that are required to record and to display interactions and therefore automatically create a tutorial by demonstrating the different steps of an assembly.

Instructions consist of multiple single steps that are made up of a description of a part, that is needed for the next working step. This description can contain additional information like the place where the part can be picked up, and description of how the existing parts have to be assembled and if tools are needed for the construction.

4.1 Distinction

As discussed in [DHB+04], it is difficult to define situations in a way that automation systems can distinguish them.

In order to divide the assembly process into single working steps, we need to clarify what we can define as a working step. What are the indicators we can look for to separate the current step from the next working step. What can be used to mark down single steps and how can we use the system to detect the beginning, the end of a working step and therefore switch to the next working step.



(a) **(b)** **(c)**

Figure 4.1: Indicators for steps: (a)Picking of a part from a box (b)Asembly of a part (c)Usage of a tool

For our model we defined three varying kinds of working steps:

- Picking up parts: This is the withdrawal of parts from the corresponding box that contains this item.
- Assembly: Placement of obtained part at specific position or assembly with other parts.
- Usage of tools: Picking up corresponding tool and use at specific part.

For picking a part, the user needs to know what part he or she needs to pick and in which box the part is stored. This is usually the working step that indicates the beginning of a workflow. After the user picked a part he or she has to place it on the working area or assemble it with a already placed part. For the placement or assembly the user needs information where he has to place or how he has to assemble the part. During some assemblies, usually after picking a part like a screw, the user needs to use tools (screwdriver or hammer) to compound the parts. If there are various tools available at the work station, the user needs to know what kind of tool, what size he needs and where he can find it.

4.2 Trigger

Triggers are an important part of our model that we need to detect the occurrence of events. By these we can identify an executed working step, when a step is finished or a new one begins and switch to the next working step. To achieve this we need to define how we can bind the conditions of the indicators to something we can detect with our sensors. We decided that the detection of high level actions are enough to identify our defined conditions.

To detect the picking of a part from a box we found out that it is enough to recognise that a hand is entering a box. Therefore a frame that can be placed at each box, will trigger if a withdrawal is executed (see Figure 4.2 A). The withdrawal is defined by the change of a

Figure 4.2: Triggers: A) Frame that monitors box depth data B) Depth image difference for assembly C) Image recognition for usage of tools

certain percentage of the depth data in the according frame. If needed, this percentage can be adjusted.

The Kinect depth image (see Figure 4.1 B) is used to detect a change in the working area. In case and assembly is executed the system will identify the assembly by comparing the current depth image with a previous depth image.

For detecting the usage of tools, activity recognition could be used to record the performed task, but for this the user would need to wear a wristband or other devices that could affect the work. When top mounted tools or tools that are place in a specific holder are used, a option would be to install sensor that can detect when a tool is being pulled down or picked up from its holder. For better scaling and to get rid of the need for additional installations, that would add complexity and cost to the workstation, we decided to use image recognition [BTVG06]. To detect the usage of tools we use a high level detection via the RGB image of the Kinect, that triggers if the tool is picked up and put down again in its designated zone (see Figure 4.1 C). These triggers define the end-conditions of the current step and as soon as one of the define conditions are triggered, the system will register these and automatically switch to the next working step.

These triggers are highly adaptable and adjustable to allow to use the systems in variable workflows. The frames of the withdrawal trigger can be adjusted if the workflow demands a different source of parts other than boxes and the backside of the table. Alls kinds of tools can be used with the trigger for tool usage, the tool only needs to be added to the available tools in the system by taking a picture.

Our defined workflow starts with the picking of the first part. For this the user gets a feedback light, from which box he has to take the part. During a withdrawal the indicator and hence the end of the current step will be the users hand that grabs the parts and hereby enters the box. This will trigger the next working step, that is usually an assembly process (see 4.2 A)). While an assembly is in progress we use the change in the field of depth to determine if there is still movement (the users hand is still operating). If the field of depth is not changing, it indicates that the assembly is executed and the next working step is in place (see 4.2 B)). In order to save the assembly step the depth image of the assembled part gets compared to the depth image that represents the workplace before the assembly. The difference of the picture then gets saved as the assembly that just occurred. In case a tool is need to perform an assembly, the capability of the system to recognise predefined objects is being used. The tool gets compared via image recognition to the saved image of the available tools. If a tool is being used and returned to its designated place, the system then gets triggered by the recognised image of the tool and uses this as an indicator that the usage of the tool is completed and the next working step is due (see 4.2 C)).

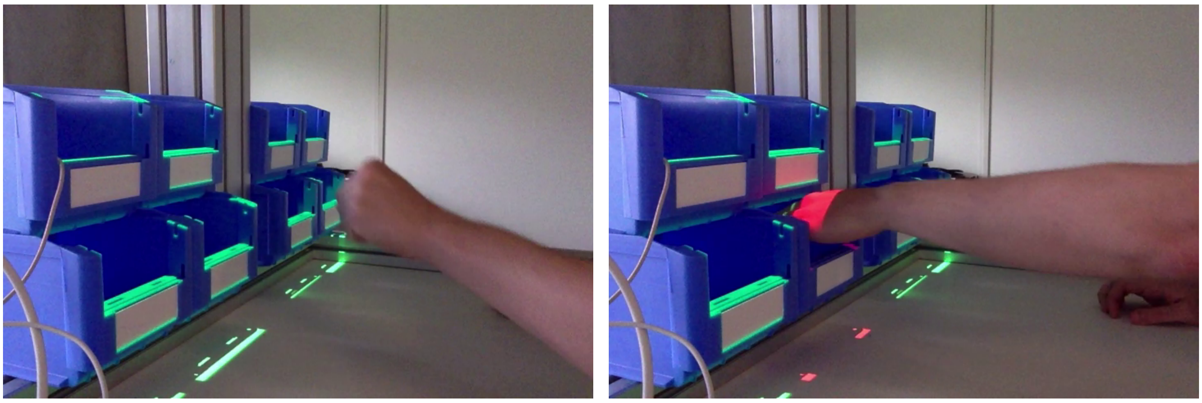
4.3 Feedback

We use two different areas for feedback. Working area feedback that will show information on the actual working surface and boxes. On-screen feedback, that will show additional informations like a list of working steps, that are either marked as completed or unmarked. The on-screen feedback will be shown on an extra screen, that provides the GUI of the system and serves as a control screen to monitor the process and gives a overview of the whole workflow (see Figure 4.5).

The feedback gets adjusted to the mode the user is using. Selectable modes are, the editor mode where instructions are created via the editor, Programming by Demonstration where the user demonstrates the assembly and the replay mode where the user can use the instructions to a perform a workflow. The given feedback and its behaviour will be explained in the corresponding following sections.

4.4 Editor

To enable the recording and give feedback for the withdrawal from boxes or assembly, the user has to create these for every box and assembly zone. If not already created, the user has to switch to the "Boxes"-tab and for each box, click on the video window and adjust the frames to the demanded size. To create the feedback, the user has to switch to the "Scenes"-tab and create a "New Scene". Then switch to the "Editor"-tab and drag and drop the desired shape (e.g. rectangle, circle,...) from the editor window (see Figure 4.4) to the second screen (onto the working surface of the projector), place and resize it to the desired position on a box or a place that marks the assembly step (see Figure 4.7a). Then switch back to the "Scenes"-tab, save the



(a) **(b)**

Figure 4.3: (a)Indicated boxes in recording mode (b)Feedback for activated boxes

scene for every box that is in use and save it with an appropriate name like "Withdrawal Box X". This has to be done for every assembly part as well and saved with an appropriate name like "Placement Part Y".

After this, the user has to create the assembly zones to enable the detection of each assembly step. For this the user has to switch to the "Assembly Zones"-tab. To record the current depth image the user first has to click on the "Snapshot"-button then place the part and click on detect zone. By doing so, the system will record the depth data of the assembly step. This has to be repeated for every assembly step and saved by clicking on "Save Assembly Zones". It is important to note the numbers of each zone for following tasks (for example the names for the zones could be z0-z3).

After this the user has to connect the created zones to one single workflow. This is done by clicking on "File -> New Workflow", click and select "Load boxes","Load Assemblyzones" and "Load Object Zones". For every working step, either withdrawal or assembly, the user has to click on the "+"-button to add an step and enter a name (e.g. "Withdrawal" or "Assembly" and an end condition that describes from which box a part has to be picked or what assembly zone is being used (e.g the noted z0-z3). In case a tool is being used the user has to create the feedback the same way as for an box or assembly. These steps define the workflow, now the user has to connect the before created zones by opening the menu of each step in the "Workflow Steps"-list, by clicking on the small arrow and right clicking the element and click and select "Add Scene". After this the workflow is created and can be saved.

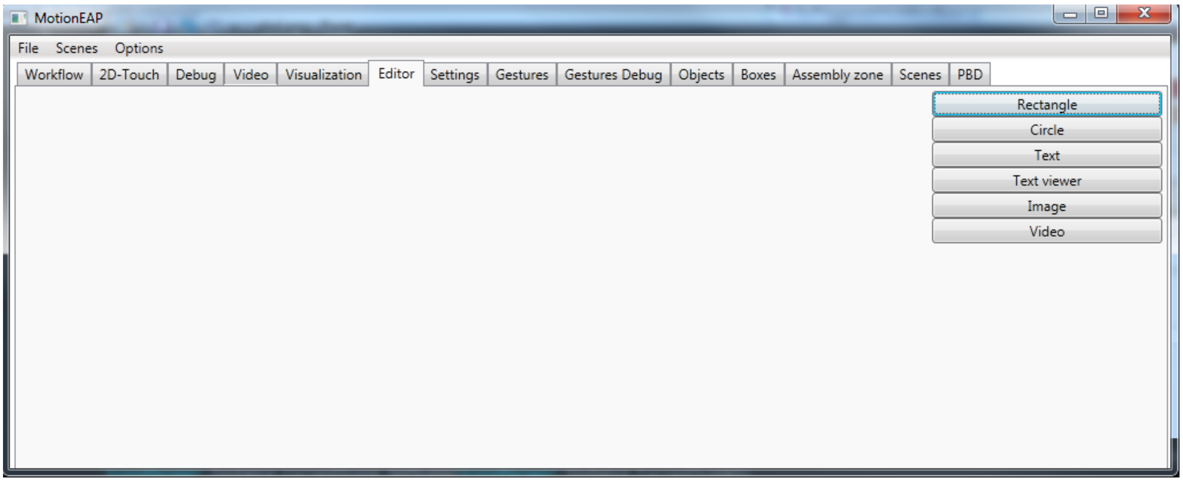


Figure 4.4: Editor screen that provides shapes for feedback.

4.5 Recording of Instruction

If the user wants to create a new instruction by demonstration, he or she simply needs to start the application. In case a new layout or table is used, the user needs to load the box layout and object zone layout that are needed at the specific setup, otherwise the last used layouts automatically get loaded. These layouts make it possible to enable a fast selection and customisation of different setups on various tables or work stations. The layouts need to be created only once for each setup of a workstation on the contrary to the editor process. By switching to the "Boxes"-tab and open the box layout by clicking the "Load Box Layout"-button and select the desired layout. These layouts are customisable and can be fitted to each individual setup and defines which boxes can be used to detect the withdrawal of parts. The loaded layout can be seen and tested in the "Boxes"-tab. For the case that the usage of tools are necessary, the user has to load a "Object Zone" by switching to the "Objects"-tab and then select a zone by clicking on "Load Detection Zone Layout"-button. This procedure is necessary in the current version of the system and only once when the application is started. It is not necessary in case the user wants to record another instruction.

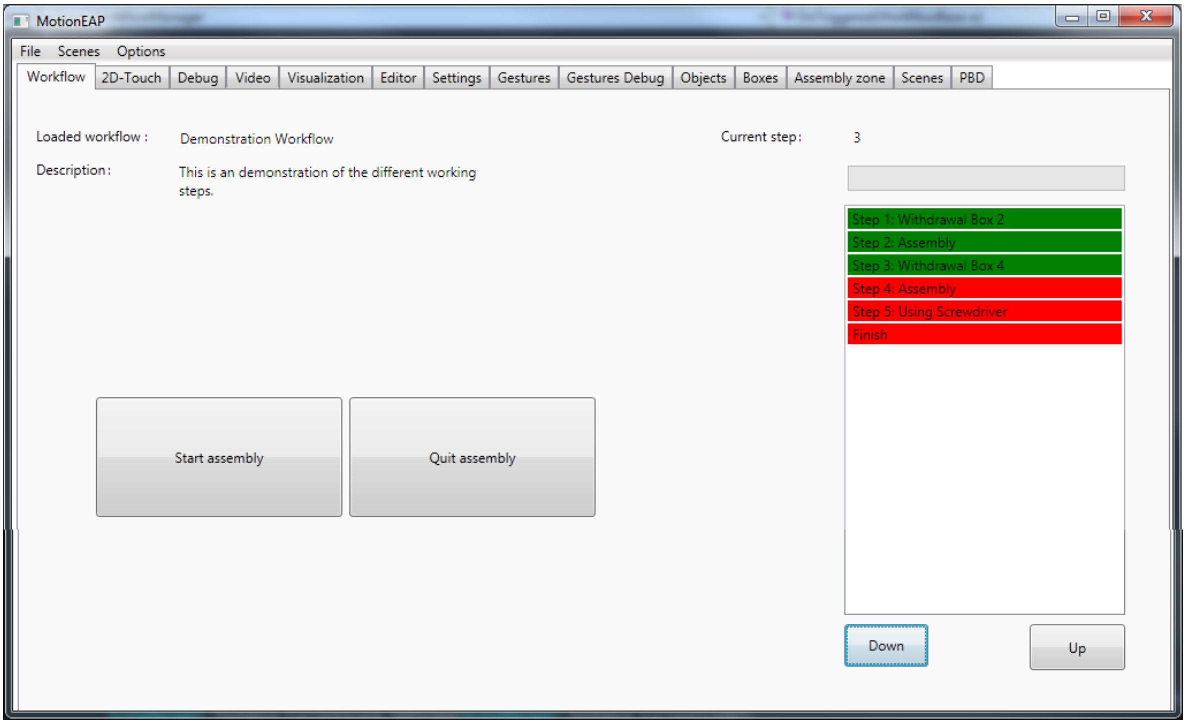


Figure 4.5: GUI overview feedback for an assembly process.

In the recording mode, in-situ projection on the working area is used as direct feedback. All boxes got a green field (see Figure 4.3a) that will turn red when the user reaches into the box (see Figure 4.3b).

If the user wants to start the recording, he or she has to switch to the "PBD"-tab (Programming by Demonstration). It is possible to label the workflow and give a description of the task that is going to be recorded. By clicking on the "Start Recording"-button the system now tries to detect possible working steps. Usually a workflow starts with the withdrawal of a part from a specific box. By reaching into a box the system will now record the working step and will create a new item in the Working Steps List on the external screen with the Name "Step X: Withdrawal from Box y".

If the user wants to record a placement of a part or an assembly, he or she simply has to perform the task. By taking the hands out of the workfield the user signals the system that the assembly is executed. The system detects that there is no movement in the working area and therefore will begin to analyse the changes of the assembly. After a short moment the executed assembly is detected and a new step "Step X: Assembly" will be added to the working step list that is shown as on-screen feedback in the "Workflow"-tab. Additionally a feedback massage "Zone created" will be projected on the working area. The on-screen feedback in the "Assembly Zones"-tab will show a frame around the area where the depth data has changed and a percentage will indicate how much the part assorts with its recorded position (see Figure 4.6).

If the usage of a tool is needed, the user has to pick up the tool, and after using it, lay it back to its designated place or holder. After a short moment a item with the label "Step X: Usage of tool Z" will be added to the working step list.

(a) **(b)**

Figure 4.6: (a)Consistency feedback of correct assembled part. (b)Consistency feedback of incorrect assembled part.

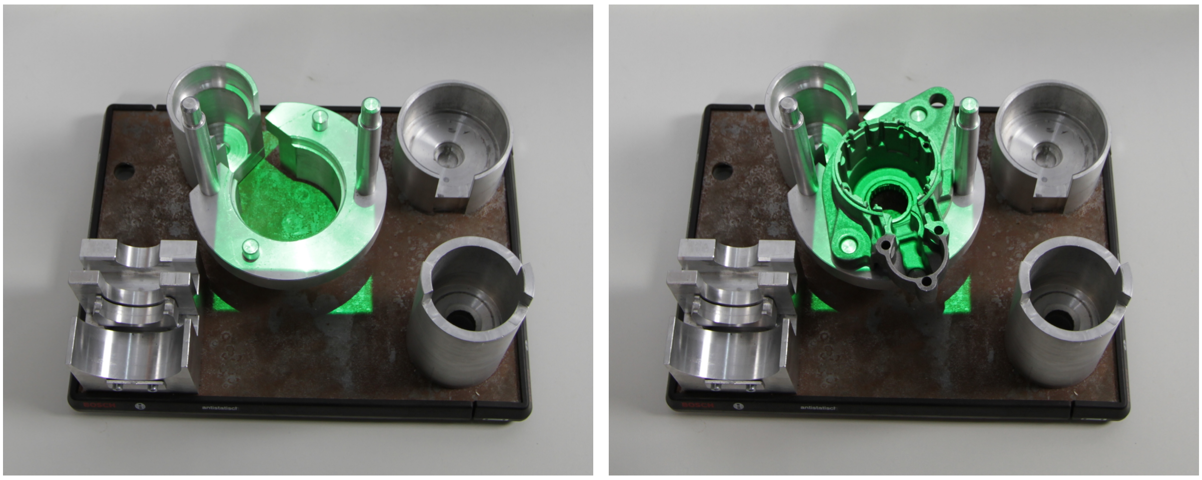
These steps can be repeated until the complete process is recorded. If this is the case, the user has to click on "Stop Recording" and "Save" in order to save to recorded process. While recording the user has the possibility to pause the recording by clicking on "Stop Recording", while "Status:" shows "paused" the user can test the detection of the assembled object and can inspect the correct placement of the objects without the system interfering and possible false recordings. If desired the user can continue the recording by clicking "Continue Recording" or save it.

4.6 Assembly with Instruction

In order to test the recorded instruction or for actual work application the workflow needs to be loaded by clicking on "File->Load Workflow" and then select the desired available workflow. The "Workflow"-tab will now show information about the loaded workflow, like name and description. Additionally the list of all the working steps are visible in the working step list on the right side (see Figure 4.5).

To start the instruction the user needs to click on the "Start Assembly"-button. The first instruction then will be shown on the work surface. Usually this will be the picking of a part from the box that is highlighted with a green field. When picked the feedback will switch from the box where the part was picked, to a green field that shows the area where the part has to be placed or an assembly needs to be performed.

When a part needs to be placed on the working surface or needs to be assembled with another part. The respective area will be highlighted with a green field (see Figure 4.7a) that signals



(a) (b)

Figure 4.7: (a)Indicated area for placement (b)Placed part

the user where he has to place the part 4.7b. The system will detect a correct assembly as soon as the assembled part is not covered up by the users hands. As soon as the interaction from the user with the object stops and the assembly is performed, the system will check if the assembly is correct by comparing the depth image of the assembly area (the green field) with the depth image of the corresponding assembly task from the instruction.

If a tool is needed to perform an assembly it will be highlighted. The working step will be marked as executed as soon as the user puts the tool back to the highlighted area. The system detects the tool via image comparison with the database of available tools. This high level detection is chosen due to the problem that it is not possible to check the firmness of a tightened screw or similar tasks.

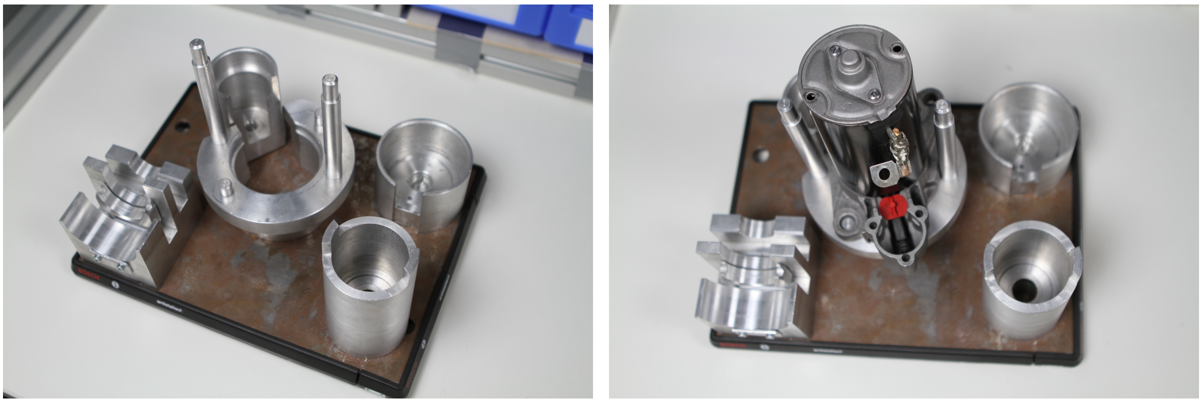
After each successful performed step, the step will be marked green on the working step list, that is shown as on-screen feedback to give the user a feeling for his or her progress in the construction process and grant additional information about the working steps. If the complete construction is done, the user has to click the "Finish Assembly"-button to terminate the current workflow.

5 Study

In order to prove our assertion tha Programming by Demonstration is an easier way to create instructions for assistive systems than conventional ways of writing or programming instructions, we conducted a study where three conditions were to be tested. We decided to split the study in two parts: In the first part we examined how the participants will perform using the three provided conditions to create instructions. For the second part we tested the created constructions from part one and how the conditions affect the assembly process and outcome.

5.1 Study 1: Creating Instructions

In the first part of the study we conducted a repeated measures study where the participants had to create a instruction of the assembly process with each of the three conditions. The participants were workers that are experienced in manual assembly. The parts that were used were know to the participants, but we chose a new assembly so that the participants didn't know the process.



(a) **(b)**

Figure 5.1: (a)Work piece carrier for the starter parts that was used in the study. (b)Starter for the engines.

The independent variable for the study were the three conditions (editor, demonstration, video). For the dependent variables we chose factors that are vital for a successful and effective industrial process. We measured the task completion time so that we were able to compare the effort of the different methods. The task load [1] was important to track down if a method stresses out the participant and therefore would probably not be viable on a long-term basis.

Using videos of the assembly as an instruction option was one of our choices that represent using digital media as an way of manuals for assistive systems. And are still a current method in the industry to teach workers new assembly tasks. We decided to record a single video that shows the complete assembly process and runs on repeat. We tested conventional editors as one of our conditions. Editors are the state of the art systems to create instructions for assistive systems, these are operated manually and it is a time consuming and stressful process where the worker has to by highly concentrated in order to make no mistakes. Due to their complexity the user needs an special amount of training to create instructions by hand [ops][KSHK12].

The last condition in our study was Programming by Demonstration, in order to be able to compare our proposed new way of creating instructions in an more natural and easier way.

We decided to conduct the study with starter parts for Audi engines, that are already known to the participants in part one of the study5.1a. The work piece carrier that was used is specially designed to improve the assembly process with assistive systems and will be used (possibly in different forms) in industrial production.

At the beginning of the study all participants got a presentation about the system of what assistive systems are and in what way they can improve the everyday working process of assembly workers. Before each round the participants got taught the assembly process by the supervisor. The participants could practice the process and ask question till they were confident with the given task. After each condition the participants filled out the questionary.

In the video condition the participant only had to assemble the object, the supervisor started and stopped the recording. The participants were told to take care that they don't cover up the parts and assemble in a speed that seemed appropriate for an instruction.

Before using the editor, the participants got an introduction about the system, the editor and what they will have to do. Because of the high complexity and step training curve the participants were guided with a detailed step by step instruction by the supervisor.

For the Programming by Demonstration condition, the participants got a short introduction of how the system works, what the components like Kinect and projector are, how we use the depth data to detect and determine objects and working steps and that they will have to wait for the feedback of the system after a placement or assembly is executed. After this they had time to try out the detection and feedback of the boxes and get a feeling for it.

[1] http://humansystems.arc.nasa.gov/groups/TLX/

The study was conducted in cooperation with Audi AG in Ingolstadt. This was a very vital advantage to be able to test our system with the target audience of industry workers that are experienced in the area (for the study part one) and inexperienced workers that had to learn new assembly processes (vital for part two).

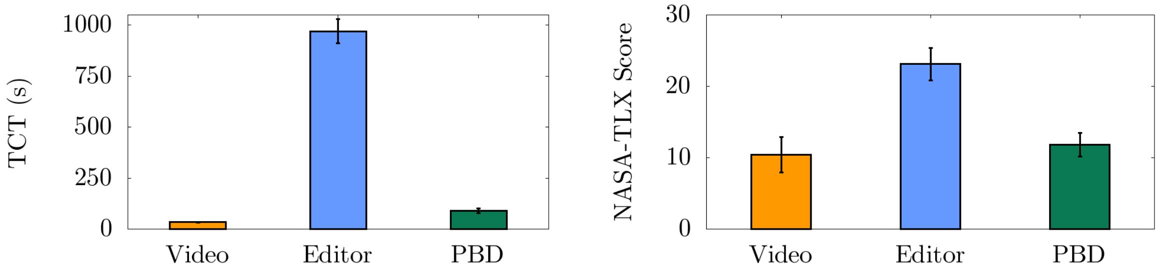
The participants were assembly workers at Audi AG and therefore experts in the field of the assembly with the used parts. The corresponding assembly process was prior unknown. The average age of the participants was 31.1 years, two of them were female and eight male.

5.1.1 Results

The results show that creating a video instruction is the fastest condition with an average task completion time of 34.6 seconds. Shortly behind is the Programming by Demonstration condition, that took the users an average of 91.3 seconds to complete the task. The by far longest task completion time is caused by the editor condition with 970.1 seconds.

The task load index for creating instructions with a video is 10, together with the Programming by Demonstration condition, that has a slightly higher task load index of 11.8. The task load index of the editor is over double the value of video and programming by demonstration, with 23.1. The creation of instructions is the editor still confusing and stressful as shown by commentary of participants and high TLX, Even tough an exact step by step instructions from the supervisor was given and highly repetitive tasks had to be completed.

Participants commented that Programming by Demonstration or recording a video are the easiest ways to create an instruction and they like the low amount of time it takes to use these conditions and that they would like to work with these in their everyday work. Other



(a) **(b)**

Figure 5.2: (a)Task Completion Time shows how long the participants needed to create an instruction with the corresponding condition (b)Taskload Index shows how demanding the performance of a condition is

participants commented that they would like to use Programming by Demonstration in their everyday work because of the easy and straightforward operation.

Seven participants commented that the editor was the condition that they liked the least. One participant added that the editor is too complex and time consuming for the task that needs to be completed.

5.2 Study 2: Assembling Based on Instructions

In the second part of the study we decided to conduct a in-between subjects study and verify previously created instructions to tutor participants with the assembly. The participants used one of the three conditions created instruction from part one, in order to assemble the starter. The participants were workers that are inexperienced in the assembly process of the starters and therefore the target audience for the assistive system.

The independent variable for the study were the three conditions (editor, demonstration, video). For the dependent variables we chose factors that are vital for a successful and effective industrial process. We measured the task completion time so that we were able to compare the effort of the different methods, the error rate to track down possible problems that the participants can have with the system or errors, if the system fail to process the input correctly. The task load [2] was important to track down if a method stresses out the participant and therefore would probably not be viable on a long-term basis.

Using videos of the assembly as an instruction option was one of our choices that represent using digital media as an way of manuals for assistive systems. And are the current method in the industry to teach workers new assembly tasks. We presented the video on an external screen that was placed at the side of the working surface. The video that shows the complete assembly process and runs on repeat, so that the participant that uses this instruction video has the possibility to watch it again if he made a mistake or is not sure how he or she has to assemble a part.

The instructions created by the editor and Programming by Demonstration, represented the state of the art and our proposed way of creating instructions. They were tested as two separate conditions, in order to verify the integrity of each instruction, even though they are similar when it comes to using these instruction.

Before the study all participants got informations about the system, what assistive systems are and that we test if the instructions that got created by other workers are good enough to teach inexperienced workers the assembly process.

[2] http://humansystems.arc.nasa.gov/groups/TLX/

We gave each participant a short introduction of the system and explanation of the task they have to accomplish. Participants that used the video instruction, that played on repeat, had the possibility to watch the video as many times as they wanted while they assembled the starter.

In case of the editor and programming by demonstration where the system will guide them through a assembly process, the participants simply had to follow the instructions given by the system. The only instruction given by the supervisor was that after a assembly step is executed, the participant has to take their hands away from the object and let the system check if the assembly is correct. If the assembly was performed correctly, the instructions for the next working step appeared automatically.

The participants were assembly workers that were inexperience with the used parts and similar assembly processes. The average age of the participants was 47.8 years 12 of the participants were female and 39 male. By a total of 51 participants, each condition got tested by 17 worker.

5.2.1 Results

The task completion time of the video and Programming by Demonstration condition were 2.13 minutes. The editor had a slightly higher task completion time of 2.31 minutes. The task load index of editor and programming by demonstration were 28 and 27.53. While the video condition has an task load index of 20.59. The lowest error rate is caused by programming by demonstration with 1.11. Slightly higher is the error rate of the editor with 1.23. The highest error rate has the video condition with 1.52.

Frequent comments regarding the video instructions were that the steps shown in the video were too fast. A participant commented that the video is only suited for training new workers but not as an everyday application and criticized that there is no feedback regarding wrong executed assembly. Another participant criticized that the view of the assembly was not 1:1.

The results are approximately equal, when it comes to using these instructions created by Editor and Programming by Demonstration. The only real difference lies in the way they are created. Sometimes it was unclear to the participants how parts have to be placed, and placed them upside down, because of the feedback shape and the suggestive shape of the holder for the parts.

Participants commented that the instructions created by the editor would be useful in everyday tasks and that "the system would be a lifesaver for handicapped people". Participant added that the step by step instruction of the workflow and the easy handling with a positive remark.

One participant commented that he or she "thinks that the system is great and a good idea for sheltered workshops" but wouldn't like to use it on a daily basis because he or she "wants to be challenged." A participant commented that a improvement would be to project how many parts have to be picked from the box and to project the exact position where the part has to be assembled.

Participants commented that a system that uses instructions created by Programming by Demonstration would be really useful and they would enjoy working with such a system everyday. Several participants commented that they would use it everyday because of the capability to verify correct executed tasks. Another participant commented that the feedback marking is show till the assembly is correct.

A participant commented that "the instructions and the order of the working steps were easy to understand". Others added that sometimes the instructions of how to place a part was not detailed enough and that "one or two prominent portions of the part should be more highlighted" and the demanded rotation should be shown. Some of the participants that could imagine to use such a system everyday added that "the system should once be explained by an expert" A frequent comment was that the system would especially be suited "to train new workers or for the assembly of rarely executed tasks".

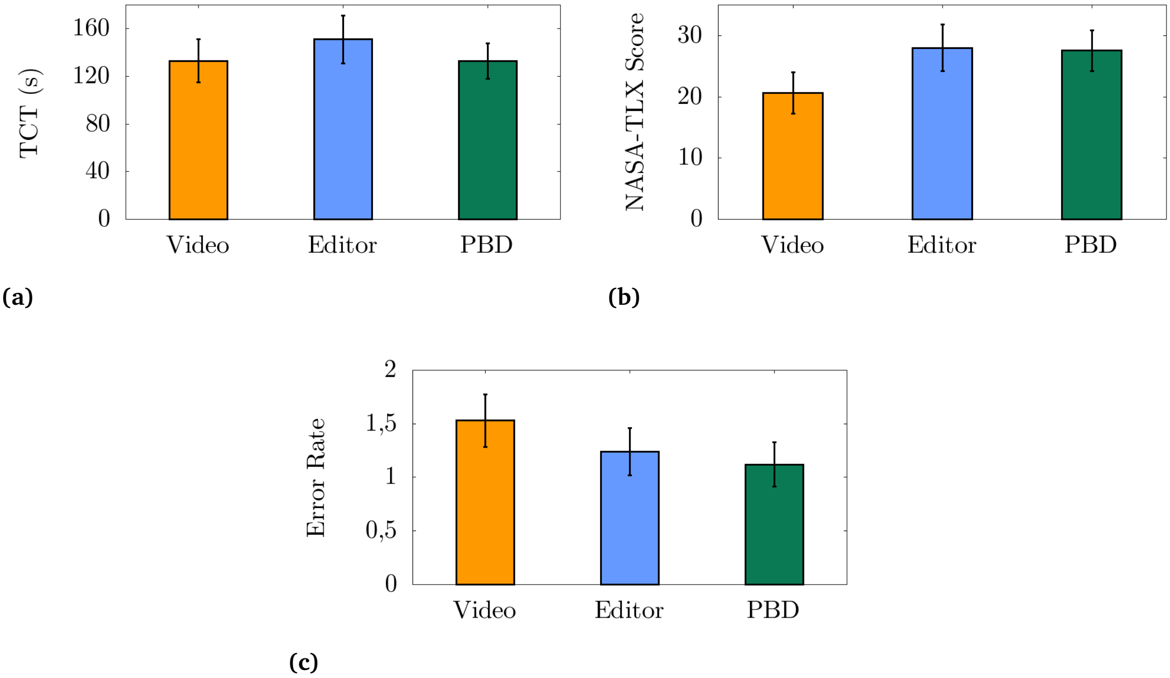


Figure 5.3: (a)Task Completion Time shows how long the participants needed to create an instruction with the corresponding condition (b)Task load Index shows how demanding the performance of a condition is (c)The Error Rate shows the number of errors during a task

5.2.2 Discussion

The favoured methods to create instructions were the video and Programming by Demonstration. As the results of the task completion time, task load index and comments show, are these the most pleasant and fastest ways to create instructions.

The high task completion time and task load index, show that the editor is the slowest and most stressful way to create instructions. Participants stressed this with their comments about the discomfort of using the editor.

The results of the assembly with instructions showed no significant difference in the task completion time and task load of the instructions created with the editor compared to Programming by Demonstration. Therefore we can assume that the only difference lies in the process of creation. Participants expressed that more detailed instructions would be helpful in cases where the part can be placed and assembled in different ways (for example upside down) and still fit in the work piece carrier and projected box.

The feedback regarding the video on the creation aspect was, that is a fast an simple way to create instruction, but using the video for assembly got negative feedback regarding the missing assistance and controlling aspect and confusing instructions regarding the angle of the video. Additionally the video had the problem that it couldn't provide a step by step instruction, as positively remarked on the Programming by Demonstration and editor instruction, but rather only show the complete assembly process. A further problem is that the video demands to switch the focus between the assembly task and the on-screen feedback.

Therefore it should be tested to use the advantage of the video, to give highly detailed instructions, to improve the problem of the instructions created by Programming by Demonstration or editor. This can be done by record the assembly process via the Kinect and cut the video automatically according to the single steps and show the current step on repeat, while still giving the control possibility of the system and eliminating the missing function of the video to give only provide needed information of the current step. The video could be show as the on-screen feedback or projected as extra information on the working surface next to the assembly zone. This additionally could remove the need to switch the focus between the on-screen feedback and the assembly task.

The study showed that Programming by Demonstration is an improvement to creating instructions with an editor. But further research is necessary to explore the significance between instruction created by video and Programming by Demonstration.

6 Conclusion and Future Work

This thesis presented the exploration of creating instructions by programming by demonstration with an assistive system in the context of industrial assembly and therefore investigated the integration of different recording and feedback possibilities. Furthermore the presented system has the ability to control the assembly that is performed by the user and give feedback to correct performed assembly tasks.

We focused on the possibilities that such systems provide for industrial applications and explored the creation and usage of instructions. The instructions were created via an state of the art editor or instructional videos. Further we presented our hypotheses that creating instructions by demonstration is as alternative and improvement to the state of the art methods in an industrial context. We presented the possibility to create instructions by simply demonstrating the assembly process. The system then automatically records all working steps and provides and instruction that furthermore has the ability to give feedback to the user and control the working steps by detecting correctly executed assembly tasks. This is done by scanning the depth image of the working surface and therefore safe the changes of objects.

The hypotheses was not fully confirmed even though the findings support the hypotheses. Our proposed thesis to create instructions by demonstration were 3 times slower than comparable video instructions, but faster than creating instruction with an editor by a factor of 10. However the comments and feedback of the participants regarding the everyday use and controlling possibilities of Programming by Demonstration and such assistive systems further support the hypotheses. However, exploration between instructional videos and instructions created by Programming by Demonstration are necessary.

The study showed that the difference between instructions created by the state of the art editor and Programming by Demonstration only are significant in the way creation and not in the terms of using these instructions. Therefore Programming by Demonstration is an less complicated way of creating instructions that is less stressful for the user. Furthermore a follow up study, with work flows that have a different level of complexity, needs to be performed.

Future Work

There are still plenty of possibilities to improve the system and application. For the recording of different tasks and usage of tools, activity recognition could be used to detect the current performed task. Simple feedback options like showing the number of parts that have to been picked could be useful additions to already existing feedback options.

A interesting field of research is the function of demonstrating the same workflow several times, possible with single step in different order and let the system then create instructions that allow different paths of assembly and support the worker by detecting the current state of the assembly and showing him possible paths of how the assembly continues. This could be used to determine what paths would be the most performant routes of assembly by detecting the current state of assembly and the position of the worker and improve the picking from wide selection of parts that are spread out over several shelves.

Research in the field of feedback could address the ability to support partially blind worker that need additional or different indicators for working steps. These could by any form of support like different colours or a colour blind mode, giving auditory feedback and instructions or using movement as indicators.

For improved instructions when it comes to placement or assembly, showing the shape via projection rather than only showing the position could be helpful. Even with a holder for exact placement that supports the user, sometimes different placements can be possible in the indicated area. Additionally the drawbacks of programming by demonstration could be improved by showing videos of the current working step on the additional screen or at the side of the working surface. This could diminish the problems of unclear instructions by still offering the monitoring and still offer the advantages like automatically checking and controlling the executed tasks, that assistive systems provide. These videos can recorded while creating the instruction by demonstration and automatically be cut according to the corresponding working steps that are indicated by the triggers.

It would be interesting to see the implementation of a system that detects the users working speed and error rate, probably with recording the data over a longer time, and provide him with context aware information and feedback that adjusts itself to the performance of the user. If a new workflow is performed all information should be provided and exact instructions should be given. If the users completion time of the task improves, the system could only provide him with the basics information and feedback. If the user has a high error rate in a particular step, the information amount and instruction density could be automatically adjusted in order to improve the users production quality.

Bibliography

[ASA+13] R. G. d. A. Azevedo, R. C. M. Santos, E. C. Araújo, L. F. G. Soares, C. d. S. Soares Neto. Multimedia Authoring Based on Templates and Semi-automatic Generated Wizards. In *Proceedings of the 2013 ACM Symposium on Document Engineering*, DocEng '13, pp. 205–214. ACM, New York, NY, USA, 2013. doi:10.1145/2494266.2494283. URL http://doi.acm.org/10.1145/2494266.2494283. (Cited on page 20)

[BCDS07] A. Billard, S. Calinon, R. Dillmann, S. Schaal. Robot Programming by Demonstration. In *Handbook of Robotics*. MIT Press., 2007. URL http://lasa.epfl.ch/people/member.php?SCIPER=115671. (Cited on page 19)

[BCLO05] L. Bergman, V. Castelli, T. Lau, D. Oblinger. DocWizards: A System for Authoring Follow-me Documentation Wizards. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pp. 191–200. ACM, New York, NY, USA, 2005. doi:10.1145/1095034.1095067. URL http://doi.acm.org/10.1145/1095034.1095067. (Cited on page 20)

[BNF06] G. Blasko, C. Narayanaswami, S. Feiner. Prototyping Retractable String-based Interaction Techniques for Dual-display Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pp. 369–372. ACM, New York, NY, USA, 2006. doi:10.1145/1124772.1124827. URL http://doi.acm.org/10.1145/1124772.1124827. (Cited on page 12)

[BTVG06] H. Bay, T. Tuytelaars, L. Van Gool. SURF: Speeded Up Robust Features. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, pp. 404–417. Springer-Verlag, Berlin, Heidelberg, 2006. doi:10.1007/11744023_32. URL http://dx.doi.org/10.1007/11744023_32. (Cited on page 31)

[CM92] T. Caudell, D. Mizell. Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 1992. (Cited on page 22)

[DHB+04] A. K. Dey, R. Hamid, C. Beckmann, I. Li, D. Hsu. A CAPpella: Programming by Demonstration of Context-aware Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pp. 33–40. ACM, New York, NY, USA, 2004. doi:10.1145/985692.985697. URL http://doi.acm.org/10.1145/985692.985697. (Cited on pages 18 and 29)

[FKS14a] M. Funk, O. Korn, A. Schmidt. Assisitive Augmentation at the Manual Assembly Workplace using In-Situ Projection. In *Proceeding of the CHI Workshop on Assistive Augmentation*. ACM, New York, NY, USA, 2014. (Cited on page 25)

[FKS14b] M. Funk, O. Korn, A. Schmidt. An Augmented Workplace for Enabling User-defined Tangibles. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pp. 1285–1290. ACM, New York, NY, USA, 2014. doi:10.1145/2559206.2581142. URL http://doi.acm.org/10.1145/2559206.2581142. (Cited on page 23)

[GRSP+12] J. Gómez-Romero, M. A. Serrano, M. A. Patricio, J. García, J. M. Molina. Context-based Scene Recognition from Visual Data in Smart Homes: An Information Fusion Approach. *Personal Ubiquitous Comput.*, 16(7):835–857, 2012. doi:10.1007/s00779-011-0450-9. URL http://dx.doi.org/10.1007/s00779-011-0450-9. (Cited on page 18)

[HA12] J. Hardy, J. Alexander. Toolkit Support for Interactive Projected Displays. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, pp. 42:1–42:10. ACM, New York, NY, USA, 2012. doi:10.1145/2406367.2406419. URL http://doi.acm.org/10.1145/2406367.2406419. (Cited on page 12)

[HTM10] C. Harrison, D. Tan, D. Morris. Skinput: Appropriating the Body As an Input Surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 453–462. ACM, New York, NY, USA, 2010. doi:10.1145/1753326.1753394. URL http://doi.acm.org/10.1145/1753326.1753394. (Cited on page 12)

[KSHK12] O. Korn, A. Schmidt, T. Hörz, D. Kaupp. Assistive System Experiment Designer ASED: A Toolkit for the Quantitative Evaluation of Enhanced Assistive Systems for Impaired Persons in Production. In *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '12, pp. 259–260. ACM, New York, NY, USA, 2012. doi:10.1145/2384916.2384982. URL http://doi.acm.org/10.1145/2384916.2384982. (Cited on pages 10, 22 and 40)

[KVVM04] T. Koskela, K. Väänänen-Vainio-Mattila. Evolution Towards Smart Home Environments: Empirical Evaluation of Three User Interfaces. *Personal Ubiquitous Comput.*, 8(3-4):234–240, 2004. doi:10.1007/s00779-004-0283-x. URL http://dx.doi.org/10.1007/s00779-004-0283-x. (Cited on page 17)

[LJM+12] Y. Lee, Y. Ju, C. Min, S. Kang, I. Hwang, J. Song. CoMon: Cooperative Ambience Monitoring Platform with Continuity and Benefit Awareness. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pp. 43–56. ACM, New York, NY, USA, 2012. doi:10.1145/2307636.2307641. URL http://doi.acm.org/10.1145/2307636.2307641. (Cited on page 17)

[LL12] H. Lü, Y. Li. Gesture Coder: A Tool for Programming Multi-touch Gestures by Demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 2875–2884. ACM, New York, NY, USA, 2012. doi:10.1145/2207676.2208693. URL http://doi.acm.org/10.1145/2207676.2208693. (Cited on page 19)

[MB13] R. d. A. Maués, S. D. J. Barbosa. Keep Doing What I Just Did: Automating Smartphones by Demonstration. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pp. 295–303. ACM, New York, NY, USA, 2013. doi:10.1145/2493190.2493216. URL http://doi.acm.org/10.1145/2493190.2493216. (Cited on pages 15 and 17)

[ML07] D. J. Malan, H. H. Leitner. Scratch for Budding Computer Scientists. *SIGCSE Bull.*, 39(1):223–227, 2007. doi:10.1145/1227504.1227388. URL http://doi.acm.org/10.1145/1227504.1227388. (Cited on page 16)

[MMC09] P. Mistry, P. Maes, L. Chang. WUW - Wear Ur World: A Wearable Gestural Interface. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pp. 4111–4116. ACM, New York, NY, USA, 2009. doi:10.1145/1520340.1520626. URL http://doi.acm.org/10.1145/1520340.1520626. (Cited on page 12)

[NK06] S. Nolle, G. Klinker. Augmented Reality As a Comparison Tool in Automotive Industry. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '06, pp. 249–250. IEEE Computer Society, Washington, DC, USA, 2006. doi:10.1109/ISMAR.2006.297829. URL http://dx.doi.org/10.1109/ISMAR.2006.297829. (Cited on page 22)

[ops] Light Guide Systems Ops-Solutions. http://www.ops-solutions.com/. Accessed: 2014-09-09. (Cited on pages 9, 10, 22 and 40)

[PBC06] M. Prabaker, L. Bergman, V. Castelli. An Evaluation of Using Programming by Demonstration and Guided Walkthrough Techniques for Authoring and Utilizing Documentation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pp. 241–250. ACM, New York, NY, USA, 2006. doi:10.1145/1124772.1124809. URL http://doi.acm.org/10.1145/1124772.1124809. (Cited on page 21)

[PBDM07] K. Pentenrieder, C. Bade, F. Doil, P. Meier. Augmented Reality-based Factory Planning - an Application Tailored to Industrial Needs. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '07, pp. 1–9. IEEE Computer Society, Washington, DC, USA, 2007. doi:10.1109/ISMAR.2007.4538822. URL http://dx.doi.org/10.1109/ISMAR.2007.4538822. (Cited on page 22)

[RHM+13] S. Rüther, T. Hermann, M. Mracek, S. Kopp, J. Steil. An Assistance System for Guiding Workers in Central Sterilization Supply Departments. In *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '13, pp. 3:1–3:8. ACM, New York, NY, USA, 2013. doi:10.1145/2504335.2504338. URL http://doi.acm.org/10.1145/2504335.2504338. (Cited on page 22)

[RSKM98] D. Reiners, D. Stricker, G. Klinker, S. Müller. Augmented Reality for Construction Tasks: Doorlock Assembly. In *IN PROC. IEEE AND ACM IWAR'98 (1. INTERNATIONAL WORKSHOP ON AUGMENTED REALITY*, pp. 31–46. AK Peters, 1998. (Cited on page 23)

[scr] Scratch MIT. http://scratch.mit.edu/. Accessed: 2014-09-09. (Cited on page 16)

[SD03] T. Sohn, A. Dey. iCAP: An Informal Tool for Interactive Prototyping of Context-aware Applications. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pp. 974–975. ACM, New York, NY, USA, 2003. doi:10.1145/765891.766102. URL http://doi.acm.org/10.1145/765891.766102. (Cited on page 16)

[SK96] A. Sugiura, Y. Koseki. Simplifying Macro Definition in Programming by Demonstration. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, pp. 173–182. ACM, New York, NY, USA, 1996. doi:10.1145/237091.237118. URL http://doi.acm.org/10.1145/237091.237118. (Cited on page 20)

[SLSG03] B. Schwald, B. D. Laval, T. O. Sa, R. Guynemer. An Augmented Reality System for Training and Assistance to Maintenance in the Industrial Context. In *In The 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2003, Plzen, Czech Republic*, pp. 425–432. 2003. (Cited on page 23)

[SM02] M. Schep, N. McNulty. Use of Lego Mindstorm Kits in Introductory Programming Classes: A Tutorial. *J. Comput. Sci. Coll.*, 18(2):323–327, 2002. URL http://dl.acm.org/citation.cfm?id=771322.771366. (Cited on page 16)

[UMPYHL14] B. Ur, E. McManus, M. Pak Yong Ho, M. L. Littman. Practical Trigger-action Programming in the Smart Home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pp. 803–812. ACM, New York, NY, USA, 2014. doi:10.1145/2556288.2557420. URL http://doi.acm.org/10.1145/2556288.2557420. (Cited on page 16)

[VKP] D. W. F. Van Krevelen, R. Poelman. A survey of augmented reality technologies, applications and limitations. (Cited on page 22)

[you01] *Your Wish is My Command: Programming by Example*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001. (Cited on page 10)

All links were last followed on September 30, 2014.

Appendix

The appendix contains the forms, that we handed out to the participants in the conducted study. The forms are provided in an German version, as the study was an cooperation with Audi AG and therefore was conducted with German experts of the field. The first appendix is the consent form that was used. Followed by the form that was used for the study of creating the instructions (the form was provided in different version regarding the order of conditions). We used an translation of the NASA-TLX questionnaire to determine the task load of each condition. The last questionnaire was a form to enquire the overall tendencies about the used conditions.

The presented forms are the version provided for the survey of creating the instructions. Adjusted forms were used for the survey of testing the created instructions.

University of Stuttgart
Germany

Human Computer Interaction Group (MCI), VIS

Prof. Dr. Albrecht Schmidt

Einverständniserklärung

BESCHREIBUNG: Sie sind hiermit dazu eingeladen an der **Studie** über die **Erstellung interaktiver Anleitungen am Beispiel eines Anlassers für ein Assistenzsystem** in der manuellen Montage teilzunehmen.

ZEITAUFWAND: Ihre Teilnahme dauert ungefähr **60 Minuten**.

DATENERFASSUNG: Für die Evaluation des Systems werden Zeiten und die Fehlerrate gemessen. Zusätzlich werden während der Studie Fragebögen ausgefüllt. In dieser Studie wird das zu testende System geprüft - nicht die Teilnehmer!
Bilder:
Ich bin damit einverstanden, dass Bilder von mir während der Studie gemacht werden.
Ich bin **nicht** einverstanden, dass Bilder von mir während der Studie gemacht werden.
Videos:
Ich bin damit einverstanden, dass Videoaufnahmen von dem Arbeitsprozess während der Studie gemacht werden.
Ich bin **nicht** einverstanden, dass Videoaufnahmen von dem Arbeitsprozess während der Studie gemacht werden.

RISIKEN UND NUTZEN: Mit dieser Studie sind keine Risiken verbunden. Die gesammelten Daten werden sicher und anonym gespeichert. Die gesammelten Daten werden aggregiert und anonymisiert in einem wissenschaftlichen Bericht veröffentlicht. Ihre Privatsphäre bleibt erhalten. Die Teilnahme an der Studie hat keinen Einfluss auf Ihr Arbeitsverhältnis. Die Daten werden nur in aggregierter Form und anonymisiert an Ihren Arbeitgeber weiter gegeben.

RECHTE DER TEILNEHMER: Wenn Sie dieses Formular gelesen und sich dazu entschieden haben an dieser Studie teilzunehmen, ist diese Teilnahme weiterhin **freiwillig** und Sie haben das Recht, jederzeit Ihre Zustimmung zurückzuziehen und Ihre Teilnahme jederzeit abzubrechen. Sie haben das Recht spezifische Fragen nicht zu beantworten. Die Ergebnisse dieser Forschungsstudie werden möglicherweise bei wissenschaftlichen Konferenzen oder Expertentreffen präsentiert oder in wissenschaftlichen Zeitschriften veröffentlicht.

KONTAKT INFORMATIONEN: Bei Fragen, Bedenken oder Beschwerden über diese Forschung, die Abläufe, Risiken und Nutzen, kontaktieren Sie bitte folgende Personen:
Markus Funk (markus.funk@vis.uni-stuttgart.de)
Albrecht Schmidt (albrecht.schmidt@vis.uni-stuttgart.de)

Mit der Unterzeichnung dieses Dokuments stimme ich den oben genannten Bedingungen zu.

Name: ____________________________ *Unterschrift, Datum:* _________________________

Eingangsfragebogen: Studie 1: Erstellen von Anleitungen

Teilnehmer Nummer: ____________________________

Alter: ____

Geschlecht: _______________________________________

Beruf: ____

Reihenfolge: A / B / C

NASA- TLX Fragebogen
Studie: Programmierung durch Demonstration – Teil 1

Proband ID: ______________ Bedingung: ______________ Datum: ___________________

Mentaler Aufwand

Wie geistig anspruchsvoll war die Aufgabe?

Sehr niedrig — Sehr hoch

Körperlicher Aufwand

Wie anstrengend war die Aufgabe?

Sehr niedrig — Sehr hoch

Zeitlicher Aufwand

Wie hastig oder gehetzt war das gesetzte Tempo der Aufgabe?

Sehr niedrig — Sehr hoch

Performance

Wie erfolgreich waren Sie im Lösen der Aufgabe?

Perfekt

Gescheitert

Aufwand

Wie sehr mussten sie sich anstrengen, um Ihre Leistung zu erreichen?

Sehr niedrig

Sehr hoch

Frustration

Wie unsicher, entmutigt, irritiert, gestresst, und verärgert waren Sie?

Sehr niedrig

Sehr hoch

Abschlussfragebogen: Erstellen der Anleitungen

Teilnehmer Nummer:___________

Wie haben Ihnen die Arten der Erstellung der Anleitungen gefallen?

	Sehr schlecht	Eher schlecht	neutral	Eher gut	Sehr gut
Video					
Programmierung Durch Demonstration					
Editor					

Welche Art Anleitungen zu erstellen hat Ihnen am besten gefallen?

Was hat Ihnen daran am meisten gefallen?

Welche Anleitung hat ihnen am wenigsten gefallen und warum?

Welche Anleitungsart würden Sie im täglichen Beruf verwenden wollen?

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

---

place, date, signature