

2013

Evaluation of graphical user interfaces for augmented reality based manual assembly support

Jordan Scott Herrema
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/etd>

 Part of the [Computer Engineering Commons](#), [Computer Sciences Commons](#), and the [Mechanical Engineering Commons](#)

Recommended Citation

Herrema, Jordan Scott, "Evaluation of graphical user interfaces for augmented reality based manual assembly support" (2013).
Graduate Theses and Dissertations. 13269.
<https://lib.dr.iastate.edu/etd/13269>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Evaluation of graphical user interfaces for augmented reality based manual assembly support

by

Jordan Scott Herrema

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Co-majors: Mechanical Engineering; Human Computer Interaction

Program of Study Committee:
James Oliver, Major Professor
Eliot Winer
Judy Vance

Iowa State University

Ames, Iowa

2013

Copyright © Jordan Scott Herrema, 2013. All rights reserved.

TABLE OF CONTENTS

LIST OF FIGURES	iv
LIST OF TABLES	vi
ACKNOWLEDGEMENTS	vii
ABSTRACT	viii
CHAPTER 1 – INTRODUCTION	1
Motivation	1
Research Goals	1
Thesis Outline	3
CHAPTER 2 – LITERATURE REVIEW	5
Augmented Reality	5
Assembly Tasks	12
AR UI Elements	19
CHAPTER 3 – A TAXONOMY OF UI ELEMENTS FOR AR	25
Assembly Task Breakdown	25
UI Element Hierarchy	29
Interface Evaluation	37
CHAPTER 4 – EXPERIMENTAL DESIGN AND SETUP	43
Apparatus	43
Hypotheses and Interface Design	46
User Study Structure	54
CHAPTER 5 – EXPERIMENTAL RESULTS AND DISCUSSION	58
Hypothesis (1)	61
Hypothesis (2)	62
Hypothesis (3)	65
Hypothesis (4)	68
CHAPTER 6 – CONCLUSIONS AND FUTURE WORK	72
Research Conclusions	72
Future Work	73
BIBLIOGRAPHY	75
APPENDIX A – GABBARD’S DESIGN PRINCIPLES	79
APPENDIX B – PAPER INSTRUCTION MANUAL	85

APPENDIX C – CONCRETE AR SCREENSHOTS	101
APPENDIX D – ABSTRACT AR SCREENSHOTS	111
APPENDIX E – XML SCRIPTS	120
APPENDIX F – PYTHON SCRIPTS	152
APPENDIX G – INFORMED CONSENT DOCUMENT	156
APPENDIX H – STUDY QUESTIONNAIRES	159

LIST OF FIGURES

Figure 1.	An example screenshot of an AR scene.	5
Figure 2.	Examples of (a) an optical see-through HMD and (b) a video see-through HMD.	6
Figure 3.	An example of a fiducial marker used for vision-based tracking.	8
Figure 4.	Examples of using AR to specify parts; including (a) a square frame, (b) a 3D arrow, and (c) an attention funnel.	16
Figure 5.	Using AR 3D models and arrows to indicate part manipulation.	17
Figure 6.	Examples of using AR to overlay textual information.	18
Figure 7.	Information presentation specifically associated with real-world locations.	19
Figure 8.	Using AR with semi-realistic 3D models.	20
Figure 9.	AR assembly utilizing 2D text, 3D arrows, and wireframes.	21
Figure 10.	Examples of (a) the impression of object levitation thanks to improper occlusion and (b) the use of edges to restore context.	22
Figure 11.	The four subtasks of the assembly task breakdown.	26
Figure 12.	A hierarchy outlining the breakdown of GUI elements.	30
Figure 13.	Example of using a 3D model to indicate part manipulation.	31
Figure 14.	The workstation used to conduct the user study.	44
Figure 15.	The axial piston motor assembly provided by Sauer Danfoss.	44
Figure 16.	Screenshot of the semitransparent servo piston.	50
Figure 17.	Screenshot of the semitransparent toothed shaft.	51
Figure 18.	Screenshot of additional textual instructions.	52
Figure 19.	Screenshot of an additional 2D sketch.	53
Figure 20.	Average completion times with maximum and minimum values.	58

Figure 21.	Average total error rates with maximum and minimum values.	59
Figure 22.	Average completion times for correct assemblies only with maximum and minimum values.	60
Figure 23.	Average orientation error and part error rates for correct assemblies only with maximum and minimum values shown for total errors.	60
Figure 24.	Errors of orientation.	63
Figure 25.	Average completion times for steps two and six.	63
Figure 26.	Alternate presentation of completion times for correct assemblies only.	66
Figure 27.	Confidence increases for performing the same task.	69
Figure 28.	Confidence increases for performing the same task <i>and</i> other, similar tasks.	70

LIST OF TABLES

Table 1.	Five of the design principles proposed by Gabbard.	23
Table 2.	Pairwise matchups of hierarchized GUI elements and assembly subtasks.	33
Table 3.	Source list for assembly breakdown task A—Identify Part.	35
Table 4.	Source list for assembly breakdown task B—Indicate Part Manipulation.	35
Table 5.	Source list for assembly breakdown task C—Identify Tool.	36
Table 6.	Source list for assembly breakdown task D—Indicate Tool Usage.	36

ACKNOWLEDGEMENTS

I would like to express my deep appreciation to all those who have played a role in helping me to perform my research work, to produce this thesis, and to thrive during this stage of life. I am sincerely thankful to my advisor, Jim Oliver, for his tremendous mentorship on every aspect of my graduate studies—not least his principal role in introducing me to the field of human computer interaction and having faith in my ability to make a successful transition into an exciting but entirely new and unfamiliar area of study. I would also like to extend my thanks to Rafael Radkowski, who has been instrumental in every aspect of my thesis work. He has been indispensable to the development of crucial software, hardware, and my user study itself, as well as being enormously insightful in refining the structure and content of this thesis. I am grateful to Josh Appleby and Sauer Danfoss for the provision of an ideal test assembly as well as incredibly detailed assistance with any related questions that I had along the way.

Furthermore I am forever indebted to my family and friends, who have not only shaped me into who I am, but who have also continued to provide unwavering support for me in all of my endeavors. I cannot possibly list by name everyone who has impacted me in this way, but I pray that I will be able to express my gratitude to each of you by extending to you in the years to come even a fraction of the same love and support that you have so relentlessly shown to me.

Finally, I must thank and honor my Savior and my Lord, Jesus Christ, by whom I exist, through whom I am saved, and for whom I live. Soli Deo Gloria.

ABSTRACT

Augmented reality (AR) technology is advancing rapidly and promises benefits to a wide variety of applications—including manual assembly and maintenance tasks. This thesis addresses the design of user interfaces for AR applications, focusing specifically on information presentation interface elements for assembly tasks. A framework was developed and utilized to understand and classify these elements, as well as to evaluate numerous existing AR assembly interfaces from literature. Furthermore, a user study was conducted to investigate the strengths and weaknesses of concrete and abstract AR interface elements in an assembly scenario, as well as to compare AR assembly instructions against common paper-based assembly instructions. The results of this study supported, at least partially, the three hypotheses that concrete AR elements are more suitable to convey part manipulation information than abstract AR elements, that concrete AR and paper-based instructions lead to faster assembly times than abstract AR instructions alone, and that concrete AR instructions lead to greater increases in user confidence than paper-based instructions. The study failed to support the hypothesis that abstract AR elements are more suitable for part identification than concrete AR elements. Finally, the study results and hypothesis conclusions are used to suggest future work regarding interface element design for AR assembly applications.

CHAPTER 1 – INTRODUCTION

MOTIVATION

The motivation for this research stems from the rising popularity of augmented reality (AR) applications designed to support manual assembly and maintenance tasks. The focus of such systems has been primarily on both the new hardware and software technologies used to implement them [1]. While advances in these AR technologies have been, and continue to be, crucial to the development of the field, further maturation will require significantly more attention to the design of graphical user interface (GUI) elements. It would be imprudent to assume that the GUI practices used by traditional computer-aided design (CAD)-based assembly documentation—or any type of assembly or maintenance documentation for that matter—will apply equally well to a mixed reality application. This thesis seeks to establish practical guidance for the design of AR interfaces in the context of assembly and maintenance support applications.

RESEARCH GOALS

The goal of this research is to investigate, in the context of AR applications for the support of manual assembly tasks, which types of GUI elements can typically be expected to maximize efficiency in terms of both total assembly time and errors committed. Additionally, this research will use the same criteria to demonstrate the benefits of AR assembly support over and above traditional instructional methods. These goals will be met via a user study comparing the effectiveness of three instruction methods—two of which will be AR-based. The first AR-based instructional set will leverage “concrete” GUI elements—that is, interface elements that

are intended to visually resemble the real-world objects or actions that the user is meant to manipulate or perform. The second will consist of “abstract” GUI elements—that is, interface elements which are intended to symbolize or figuratively represent relevant parts or the actions which the user is meant to take. These two sets of interface elements are intended to shed light on how interfaces designed for AR assembly applications should be set up in order to maximize overall efficiency, and will be described in greater detail in subsequent sections of this thesis. The third instructional medium—a traditional printed instruction manual—will also be assessed relative to the AR-based approaches in order to add weight to the argument that AR offers benefits to assembly and maintenance situations that traditional instructional techniques do not.

Each of these instructional methods will be evaluated with a between-subjects study, in which subjects assemble a small motor (approximately 9” x 7”) by hand and with the assistance of ordinary handheld tools. One primary evaluative measure will be the total time that each subject spends in completing the assembly. Additionally, subjects will be observed in order to allow for the written recording of assembly mistakes. These mistakes can take the form of an attempt to conduct an assembly step out of order or in an incorrect configuration, and may even involve succeeding in taking such actions. The latter may be recognized by the subject during later steps in some cases—and therefore corrected—but might not be in others. Such events will also be recorded.

The results of this study will indicate the benefits of AR over traditional instructional methods for assembly applications, as well as demonstrating the types of graphical interface elements that can be expected to maximize such benefits. The hope is that these results will enable designers of future AR applications, particularly those intended for assembly and maintenance situations, to select the most effective types of GUI elements with confidence.

THESIS OUTLINE

To close this introductory chapter, the organization of the remaining chapters of this thesis is outlined below. Chapter 2 provides a literature review consisting of three primary sections. The first describes the technology typically associated with AR in order to give the reader context for both the GUI elements being studied and the hardware implemented in this particular user study. The second reviews assembly tasks—first covering the basics of such tasks in general and narrowing in on those which are relevant to this research, and then addressing issues specifically related to AR-supported assembly. The third section then discusses user interfaces—first in general, and then as pertinent to AR systems—noting, as mentioned previously, that the vast majority of existing AR research focuses on physical technology and software rather than interface design.

Chapter 3 presents a taxonomy of user interface elements for use with AR assembly applications. The taxonomy is developed first by breaking down the presentation of information for assembly tasks into four subtasks. Next, a hierarchy is outlined which can be used to classify GUI elements—based primarily on whether the elements are concrete or abstract in nature. Pairwise comparisons are made between each assembly subtask and each GUI element classification to round out the framework. For illustrative purposes, a number of existing AR research systems are evaluated using the framework as well. Finally, each of the various interface elements are evaluated from a theoretical point of view and a concrete and abstract solution is chosen regarding information presentation for each of the assembly subtasks.

Chapter 4 covers the remaining design considerations for the user study. First, the apparatus is described including the physical workstation, the software utilized, and the physical device that the study participants assembled. Second, the actual interfaces used in the study are

described in conjunction with several hypotheses tied to those interfaces. Finally, the logistics of the study itself are addressed including participant recruitment, test case assignment, data recording methods, etc.

Chapter 5 presents the raw results of the study and then explores the significance of those results to each of the hypotheses proposed in Chapter 4. Additional charts are included to illustrate particularly relevant data, and statistical analyses are carried out in refutation or support of each hypothesis. Finally, explanations are offered for cases where experimental results do not concur with the rationale behind each hypothesis.

Lastly, Chapter 6 reviews both the methods by which user interfaces for AR assembly applications have been investigated in this thesis as well as their overall implications. To close, future work is suggested which holds the most promise for advancing the field of AR assembly in regards to user interfaces.

CHAPTER 2 – LITERATURE REVIEW

AUGMENTED REALITY

Augmented reality (AR) technology is a human computer interaction (HCI) technology in which a user perceives both the real world and virtual objects simultaneously. The goal is to integrate the virtual objects—such as 3D models or overlaid text for example—into real-world space as seamlessly as possible, creating the illusion that all the objects coexist in the same space (see Figure 1) [2]. AR is widely held as a promising technology for many fields, but it is still relatively young and thus its real-world practicality remains largely untested.



Figure 1. An example screenshot of an AR scene [2].

Process and Technology Overview

There are several possible configurations for an AR system, the most popular of which utilize see-through head-mounted displays (HMDs), closed-view HMDs, or monitor-based visualizations. When used for an AR application, an HMD is a head-worn device comprised of

either one or two forward-facing cameras, depending on whether or not the system utilizes stereo (3D) visualization, and small displays in front of the user's eyes such as miniature LCD displays or see-through reflective screens. Two examples of HMDs are shown in Figure 2 [2]. Monitor-based visualizations typically utilize monitors no different from those paired with household desktop computers, and are usually accompanied by one or two cameras oriented over the desired scene in such a location as to approximate the user's point of view without obstructing his movement.

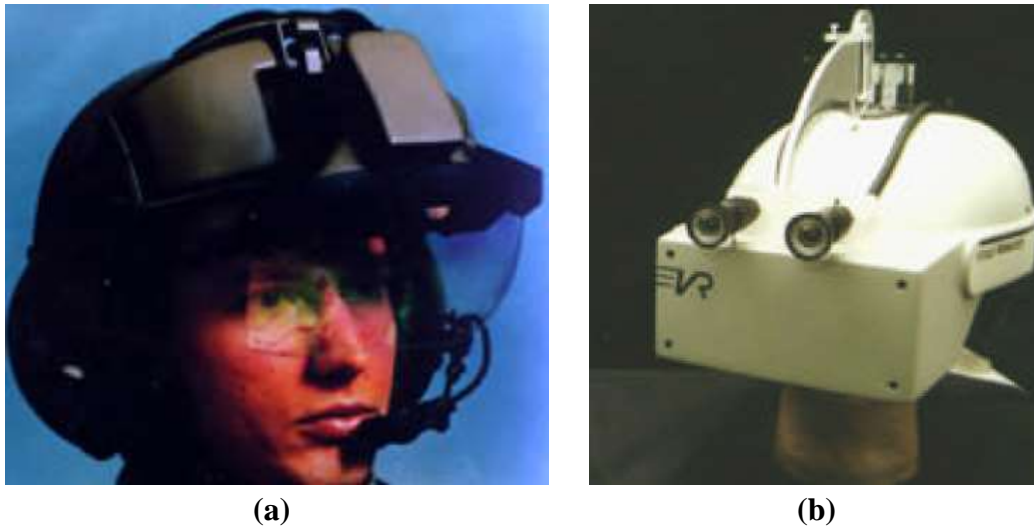


Figure 2. Examples of (a) an optical see-through HMD and (b) a video see-through HMD [2].

There are both advantages and disadvantages inherent to each of these display techniques. Provided that their location is being tracked, HMDs are advantageous in that they offer a completely egocentric augmentation of the scene that the user is viewing. However, wearing such a display can be an unwelcome encumbrance. On the other hand monitor-based systems, while typically unable to provide augmentation as realistically as HMD approaches, are beneficial in that they allow the user complete freedom from physical restraint [2].

Regardless of the display technique being employed, another integral component of any AR system is the tracking system. The most common form of tracking is vision-based tracking, wherein the cameras feed images to a computer which must process the frames in order to determine where in the scene, if anywhere, an augmentation should take place. Making this determination is contingent on the identification of key objects known as tracking targets. When these targets are identified within a scene, augmentations will take place relative to the position of the targets.

One way that tracking targets are identified and tracked is based on features intrinsic to the tracking targets themselves which the image processing software is able to recognize. This particular tracking technique is often labeled natural feature tracking. Another common tracking technique, frequently termed artificial feature tracking, involves the addition of special markers to the tracking targets, allowing the system to track them more easily. These markers can be passive, such as colored dots or unique patterns, or active such as LEDs, lasers or even common light bulbs [2][3]. These markers are typically referred to as fiducials. An example of a passive, patterned fiducial marker can be seen in Figure 3 [4]. Once the tracking targets have been identified by the image processor, they must also be analyzed to determine both their position in three-dimensional space as well as their orientation. Depending on the type of tracking used, this can be done by recognizing surfaces and edges, compiling information from multiple markers, or a variety of other techniques [3].

Depending on the particular AR application, the camera coordinate system may also need to be transferred from local to world coordinates before the final augmentation of the video stream can take place. The position and orientation of the tracking targets is determined relative to the camera that gathered the images.



Figure 3. An example of a fiducial marker used for vision-based tracking [4].

If the camera is fixed in a single location, translating these relative positions into absolute positions is straightforward. However, if the video cameras are mounted on an HMD or other mobile system, the cameras themselves must be tracked as well in order to provide global coordinates for the tracking targets [2]. Tracking the cameras can take advantage of any of the technologies previously listed, as well as GPS, inertial tracking, magnetic tracking, or a host of other technologies [3]. Finally, with the camera position(s) known and the absolute positions of the objects in view calculated, the scene can be augmented with 3D models, text, arrows, simulated tools, and a variety of other virtual entities appearing to interact with the physical world.

Advantages of AR Technology

Augmented reality offers several advantages in a variety of usage scenarios such as medical applications, manufacturing, repair, robot path planning, entertainment, vehicle heads-up displays (HUDs), and any other environment where annotation or additional visualization within a physical environment will be valuable [2]. One widely cited advantage stems from the fact that many of the above applications often involve tasks which are meant to be completed quickly—in

order to protect the health of medical patients, maximize factory profitability, minimize equipment downtime, and so forth. AR can reduced the time required to complete many such tasks by reducing head movement [5][6] as well as reducing eye movement and, correspondingly, reducing attention switching [6][7]. These benefits come about thanks to the fact that almost all of the information that a user could need to reference can be presented to him within the working area, allowing him to remain focused on the physical task being performed.

AR is also useful due to the fact that many of the above-mentioned applications require the worker to mentally transfer information from a set of instructions or other reference material into the real-world environment where the exact context, component orientation, or other details may differ. By collocating new information with the objects to which that information applies, AR systems can reduce the amount of mental transformation that a user is required to perform [6]. Similar to reduced attention switching, this frees the user to focus on the actual performance of his task, rather than devoting mental resources to making sometimes difficult spatial connections between the physical workspace and supplementary instructions or other reference information.

Finally, AR can help alleviate the performance decreases that many users encounter as a result of limited memory. Memory augmentation is very useful in training applications, since the enhancement of long-term memory aids in the understanding and retention of assembly sequences, medical diagnoses and procedures, and other information. The supplementation of working memory is also beneficial because it allows users to perform very complex operations without being required to store every detail of the task in working memory for the full duration of that task. AR aids memorization by leveraging the association of the presented information with a physical location in real-world space [6][7][8].

Challenges for AR Technology

Despite the numerous advantages promised by AR systems, AR is still a relatively immature technology overall and a great deal of progress needs to be made on a number of fronts before it will have wide scale appeal. This can be illustrated by comparing AR technology to the far more mature world of VR, which is already making significant inroads into the commercial realm [2]. Note that this subsection addresses AR challenges in general, reserving a more detailed discussion of user interface challenges for the last section of this chapter.

Currently, one of the most widely reported challenges is accurate object tracking. This challenge comes in several forms, one of which is registration errors. If an AR system misrepresents the location of a virtual object relative to an accompanying physical object by even a very small margin, the visual mismatch between the physical and virtual objects is very easily noticed by the human eye [2]. Consider vision-based tracking as an example of the technical challenge posed by registration errors. Attaching fiducial markers to objects within the workspace can be considered intrusive and limiting to real-world practicality; and yet such tracking techniques often remain insufficient to combat registration errors. Removing such markers in favor of natural feature tracking only exacerbates registration problems. Improvement in this area will come with improvement in tracking technologies.

Another tracking challenge, which is especially pertinent to vision-based tracking, arises from fast movement of objects within the scene. The blur resulting from rapid motion can lead to severe lag in tracking, or even temporary loss of tracking altogether [2][9][10]. One potential solution is attempting to predict motion, but this is extremely difficult, if not impossible, for many applications. Many other approaches are also being pursued (see [2] for more information)

and will hopefully aid in minimizing the barrier that tracking poses to the maturity of AR in the near future.

Aside from the common vision-based tracking approaches, other novel techniques are also being pursued. Powered devices can serve as effective vision-based trackers, but the circumstances where these can be used successfully are extremely limited due to the necessity of preparing the environment and powering the trackers [11]. Alternatively, inertial trackers, which compute position via relative motion, are prone to tracking drift which eventually renders the system unusable if not corrected by an external global source [11]. In an attempt to mitigate such weaknesses, the utilization of multiple tracking methods simultaneously has also been pursued. Non-powered vision-based tracking and inertial tracking, for instance, were tested in conjunction within multiple environments with a fair degree of success by You et al. [11]. But naturally, such hybrid systems come with the cost of added complexity.

Some further challenges depend more directly on the particular type of AR application. For example, it can be difficult in many situations to sense the state of the physical objects in the working area. This would apply, for instance, when attempting to allow an AR system to detect the degree to which an assembly has been completed at any given moment [12]. Another application-dependent challenge pertains to restrictions on range—a particular problem when working outdoors or in and around a very large product—as well as to the inability to prepare an environment in advance—such as when an application is intended to have the capability to recognize landmarks anywhere within an entire city [9]. To put the overall tracking challenge in perspective, Welch and Foxlin give ten characteristics of the so-called perfect tracker and go on to assert that *all* current tracking technologies fail on at least seven of those ten characteristics [3].

Finally, AR systems lack maturity when it comes to the methods by which users can interact with them. Azuma et al. write, “Until recently, most AR prototypes concentrated on displaying information that was registered with the world and didn’t significantly concern themselves with how potential users would interact with these systems” [9]. Interacting successfully with AR systems requires convenient and efficient interface mechanisms. This includes GUIs, but extends to other interaction methods such as haptics [12]. Ultimately, no matter how well an AR system tracks an object or senses an assembly state, the ability of a user to quickly understand what is being displayed and to intuitively input new information into the system is crucial to the overall progression of AR technology. Because this specific challenge is of distinct concern to this research, the third section of this chapter has been devoted to discussing it in more detail.

ASSEMBLY TASKS

Webster’s New World Dictionary defines the word *assemble* as “To fit or put together the parts of (a machine, etc.)” [13]. Alternatively, Ikeuchi and Suehiro designate an assembly task as the process of achieving a particular set of face contacts between manipulated objects and environmental objects [14]. Homem de Mello and Sanderson similarly describe assembly as pertaining to the achievement of surface contacts [15] or nonspecific connections [16][17] between pairs of components. These descriptions, while quite straightforward in themselves, allow for the possibility of a very wide range of necessary actions depending upon the nature of the objects and the exact details of the face contacts or connections desired. Furthermore, the methods used to achieve the desired outcome can vary substantially as well. In the simplest form,

objects can merely be arranged in three-dimensional space by hand. On the other end of the spectrum, parts can be assembled via tools or automated equipment, and involve such additional processes as part deformation, welding, heat treatments, etc.

As the above paragraph briefly illustrates the topic of assembly is incredibly broad; thus it is important to specify the types of assembly processes that this research will address. This research is limited to manual assembly processes in which an assembler organizes and attaches components by hand or with common handheld tools. Such assemblies can involve inserting one component into another, fastening multiple components together with bolts and a wrench, aligning the corresponding faces of two components, or many other similar operations. Because automated assembly stands relatively little to gain from augmented instructions due to its inherent lack of significant human involvement, it will be excluded from consideration here. Complex assembly processes such as welding will also be excluded. The complexity of such processes offers no advantage over simpler and less experience-dependent assembly tasks in terms of experimental insight about GUI effectiveness.

This research also addresses a limited scope of products. Objects of a very large or very small scale could conceivably be produced via manual assembly processes (as defined above) with the aid of AR—however the most effective user interface elements would almost certainly differ in such situations. Furthermore, products requiring particularly unique handling procedures such as those involving radioactive, biological, or other hazardous materials would likely require special safety considerations in the presentation of the assembly instructions. Analyzing GUI elements for such a wide range of assembly scenarios is outside the scope of this research, though it is plausible that the insight gleaned here may be generalizable to such applications in the future.

Evaluating Assembly Effectiveness

One important consideration for assembly tasks deals with the methods or metrics by which overall assembly efficiency can be evaluated. The two most important metrics for overall efficiency are the assembler's error rate (measured via a variety of approaches) and the total time required to complete the assembly; e.g. [6][18][19]. Note that these two metrics are hardly independent, since errors will increase assembly time *if they are corrected*. A number of studies also elect to focus on total assembly time alone; e.g. [20][21].

An increase in efficiency is most significant when a process is executed repeatedly, which is the case for assembly within an industrial setting. Businesses are typically concerned principally with profit, which, all else equal, can be dramatically increased if the same number of sellable products can be manufactured by fewer paid employees. Such a shift can be realized if each employee can assemble more products in a given time period—thus the significance of total assembly time—but will be hampered if those products begin to exhibit larger numbers of defects—thus the significance of error rate.

Total assembly time and error rate are only useful as indicators of an assembly task's viability over long periods of time if the study in which they are measured involves each participant working for extended durations themselves. Since this is typically considered impractical, other metrics are occasionally employed. For instance, the vast majority of assembly studies conclude with the participants responding to questions pertaining to stress levels, task intuitiveness, discomfort, and many other similar concerns which are difficult or impossible to measure directly; e.g. [6][18][19][20].

Seeking less subjective metrics, attempts have also been made to quantify mental fatigue. The NASA task load index is perhaps the most prominent example of a system designed to give

an objective indication of mental workload, and has been utilized by several studies; e.g. [6]. Measuring the index involves users taking two steps. The first consists of users giving a relative weighting of how much they feel each of six factors contributed to overall mental workload. The six factors are mental demand, physical demand, temporal demand, performance, effort, and frustration. After this, the users are presented with cards on which each of the 15 possible pairwise comparisons of the same six factors is written. Users select the factor that they feel was a greater contributor to workload from that pair. Finally, the weightings given in the first step and the ratings given in the second are fed into an averaging formula to yield the final workload index. For more detailed information on the NASA task load index, see [22].

There is no reason to believe that AR-supported assembly should be evaluated any differently than non-AR cases. Such non-AR assembly instruction methods typically include paper instruction manuals or computer-aided instructions (CAI) presented on a computer monitor or similar display. These instruction methods aim for an end result which is no different from that sought by an AR-supported assembly task, and the efficiency metrics should reflect this fact. For an example of a study in which multiple, non-AR instructional methods are compared based on every one of the metrics mentioned in this section, see Tang et al. [6].

Applicability of AR to Assembly

AR has been applied to a variety of assembly operations by numerous researchers, and is used to fulfill the same roles as paper instruction manuals and other instructional methods in order to guide users through an assembly sequence. One of these roles is identifying the assembly components and tools that a user should be interacting with at a given time. An example of this functionality is shown in Figure 4 [23]. Another role is indicating where

components and tools should be placed and how they should be oriented or manipulated, as illustrated in Figure 5 [24]. Finally, AR can fill a variety of instructional roles by displaying textual information overlaid on the assembly workspace itself, as depicted in Figure 6 [7].

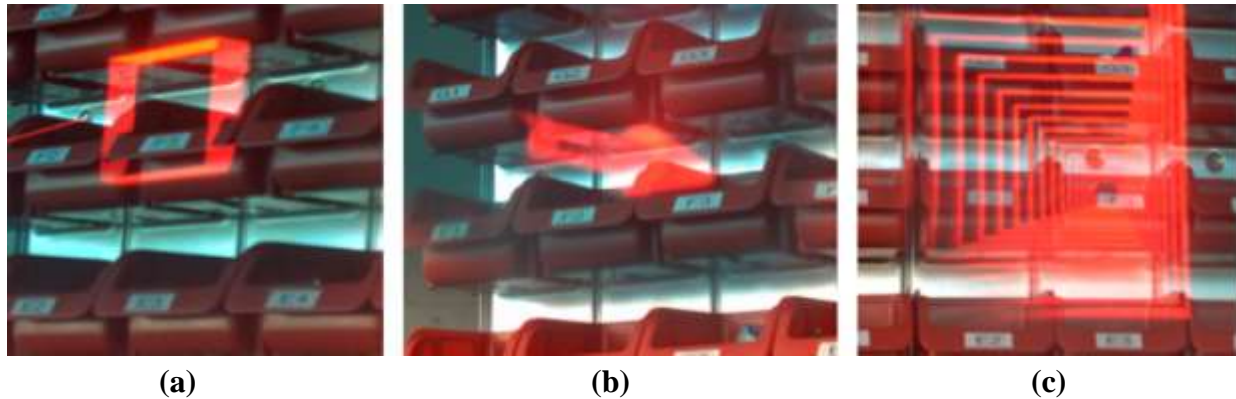


Figure 4. Examples of using AR to specify parts; including (a) a square frame, (b) a 3D arrow, and (c) an attention funnel [23].

Tang et al. explain that assembly tasks are particularly well-suited to augmented reality support. One reason arises from the process of making mental transformations between traditional instructional media and the physical workspace, which is subject to errors in the assembler's understanding of proper part location or orientation. Because it visually presents such information within the workspace itself "AR can reduce errors by eliminating locational ambiguity and explicitly indicating the orientation" [6]. The assistance with mental transformation yields additional benefits by reducing mental workload [6]. Excessive mental workload has the propensity to result in mental fatigue, which can have the undesirable effects of increasing the frequency of assembly mistakes and/or decreasing the speed at which the assembler is willing to work. Finally, reducing the severity of attention switching demands—that

is, the need to switch concentration from the workspace to supplementary information and back—also reduces mental workload and thus further bolsters the same benefits [6].

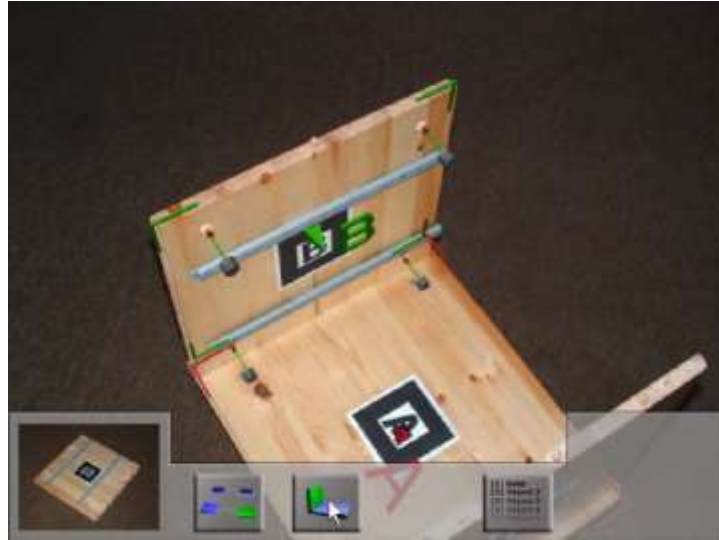


Figure 5. Using AR 3D models and arrows to indicate part manipulation [24].

Further capitalizing on the need to reduce assembly errors, Tang et al. have also shown that the usage of AR for assembly tasks has a particularly strong tendency to reduce dependent error rates (versus independent error rates) [6]; that is, errors which are caused by other errors made in previous assembly steps as opposed to isolated errors, respectively. Once again, this is due to the fact that assembly instructions are presented within the workspace itself. This form of presentation allows for discrepancies between the instruction set and the physical assembly to be detected via direct visual comparison as opposed to less instinctive comparison methods such as the visual-verbal comparison necessitated by text instructions, for instance.



Figure 6. Examples of using AR to overlay textual information [7].

Assembly applications can also benefit substantially from the improved memorization attainable via AR. Improved memorization can shorten the training time of new employees, allowing them to begin genuinely productive work on the floor sooner while simultaneously incurring fewer training expenses. AR can aid these trainees by associating information with locations and/or objects in real-world space [6][7]. See Figure 7 for an example of this type of information presentation, wherein tool and part orientations are indicated along with the textual identification of a key component [18]. As mentioned in previous sections, this association leverages spatial cognition in a way that is extremely beneficial to memory [8]. A related benefit stems from the increasing popularity of agile manufacturing and similar philosophies in industry. These processes often require assembly workers to handle continuously fluctuating assembly procedures. In this scenario, improved memorization cuts down on both the time and money that must be allocated to *retraining*. For these reasons, the enhancement by AR of both training and regular assembly work has particularly great commercial potential [6].



Figure 7. Information presentation specifically associated with real-world locations [18].

AR UI ELEMENTS

The definition of graphical user interface (GUI) given by the *Business Dictionary* reads, in part, “Software that works at the point of contact (interface) between a computer and its user, and which employs graphic elements ... instead of text characters to let the user give commands to the computer or to manipulate what is on the screen” [25]. Numerous other definitions exist but, despite the pervasive usage of the term GUI, many of them are inappropriately restrictive. A number of sources define GUI in such a way as to limit its use to circumstances involving a traditional computer mouse. Even if such conceptualizations were adequate in years past, today’s technology has given rise to an explosion of interaction techniques wherein a traditional mouse, or indeed any physical device at all, need not be present.

AR applications are very versatile in that they can utilize a wide variety of graphic elements in order to facilitate information flow between the system and the user. In some cases, the graphics are designed to mimic real-world objects—such as when 3D models are displayed. An assembly operation utilization this type of visualization is shown in Figure 8 wherein a virtual door lock and human hand are shown being maneuvered into a physical car door [10]. Other AR graphic elements can be abstract in nature. Examples of such elements include arrows, lines, and textual information—each of which are depicted in Figure 9 [4].

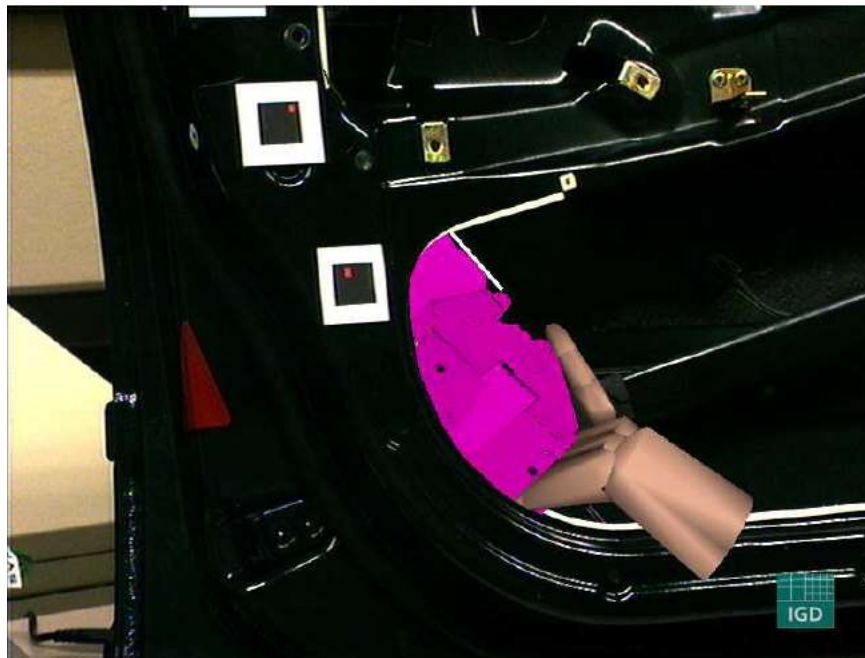


Figure 8. Using AR with semi-realistic 3D models [10].



Figure 9. AR assembly utilizing 2D text, 3D arrows, and wireframes [4].

Any of the above-mentioned graphics can be further enriched via the use of animations or other context enhancement techniques. Translational and rotational animations of graphics relative to the physical world allow for those graphics to convey additional meaning in an intuitive way—such as the insertion of a one part into another, or the twisting of a tool, etc. Additionally, enhancements such as proper occlusion further improve the intuitiveness of a graphical augmentation. Without proper occlusion graphics tend to appear as though they are floating in front of physical objects which they are actually supposed to be behind or inside, such as in Figure 10(a) [26]. In cases where proper occlusion would prevent the user from being able to view the appropriate information, cues such as partial transparency or the rendering of only object edges can be utilized to preserve accurate context without obstructing the user's view. The edge rendering technique is displayed in Figure 10(b) [26].

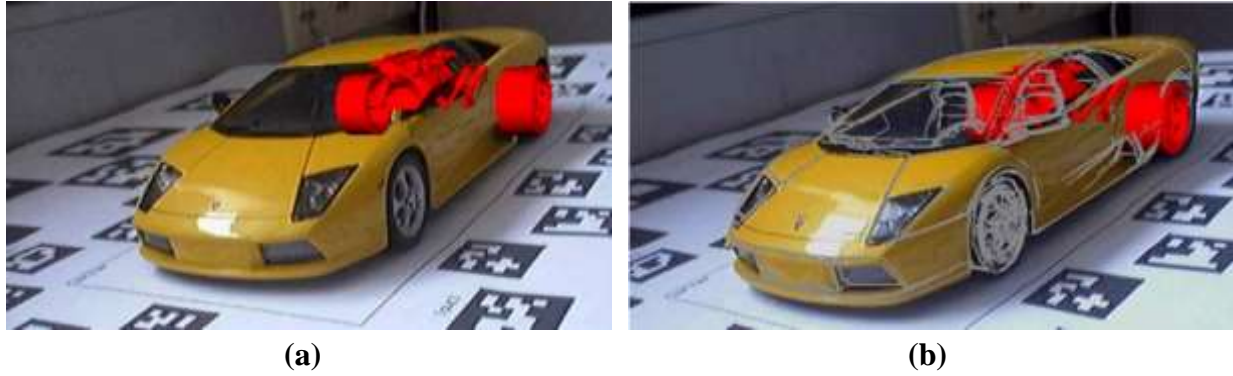


Figure 10. Examples of (a) the impression of object levitation thanks to improper occlusion and (b) the use of edges to restore context [26].

When it comes to deciding which of these graphic elements to employ in various circumstances, only a few researchers have proposed generalizable principles for the design of AR interfaces. Gabbard has assembled a list of basic guidelines for interface design within VR and AR applications, which he has synthesized based on 46 references covering subjects ranging from AR-specific research to general interface design theory [27]. All 53 of his guidelines are listed in Appendix A.

Five of Gabbard's guidelines are particularly relevant to AR GUI design and are listed in Table 1 [27]. The first and fourth—pertaining to the support of multiprocessing and the emulation of common window interfaces, respectively—are effectively inherited from ubiquitous 2D interface design, but are not always given credence for AR interfaces. The second and third—dealing with accurate location depiction and proper occlusion, respectively—are steps that must be taken if virtual objects and information are to interact with the physical environment in a way that is convincing and not distracting. Finally, the fifth guideline—which discusses progressive disclosure—is especially relevant to AR interfaces thanks to the propensity with which a combination of physical and virtual objects can quickly become confusing for a user.

Progressive disclosure can help to guide the rate and sequence in which the user processes visual information.

Table 1. Five of the design principles proposed by Gabbard [27].

Guideline	Reference
Support concurrent task execution and user multiprocessing.	Experience & Observation
Provide accurate depiction of location and orientation of graphics and text.	[28]
Support significant occlusion-based visual cues to the user, by maintaining proper occlusion between real and virtual objects.	[29]
When presenting inherently 2D information, consider employing 2D text and graphics of the sort supported by current window systems	[30]
Use progressive disclosure for information-rich interfaces.	[31]

However, Dünser et al. argue that Gabbard’s guidelines are primarily useful for the research systems from which they were drawn, and are not necessarily transferable to other AR systems [1]; thus they opt to take an approach relying heavily on HCI theory to guide their development of interface design principles. The results of this effort, like Gabbard’s guidelines, remain broad. However, several of the recommendations lend themselves relatively easily to direct implementation. For instance, interactions should be “akin to real world behavior or similar to what the users already are used to” [1]. This principle is especially sound in cases where AR technology is being leveraged in such a way as to mimic physical, real-world interactions, but simultaneously leaves the door open for interfaces which rely on users’ previous

experiences with ubiquitous 2D interfaces. Both interpretations echo the underlying principles of a few of Gabbard's guidelines.

Dünser et al. also advise consistency within and between interaction methods, so as to minimize confusion and expedite user learning in novel environments [1]. It is also advised that feedback should be given to clarify when controls have been used [1]. The latter would seem particularly applicable when an interaction method lacks physical devices that would otherwise have given some level of haptic feedback—even feedback as simple as the clicking of a physical button.

While all of the above principles and advice are useful in their own right, the implications of each to interface design are not nearly as far-reaching as will be necessary if the use of AR technology is to become widespread. The overall state of affairs is well summarized by Rizzo et al., who are referring to what they term 3DUI when writing, “With the absence of [a mature interface design methodology], we are still limited to a trial-and-error exploratory approach” [32].

CHAPTER 3 – A TAXONOMY OF UI ELEMENTS FOR AR

This chapter is dedicated to developing a classification system for information presentation UI elements within AR assembly applications. The first section addresses the decomposition of assembly task steps into their basic subtasks. The second section develops a hierarchy of UI elements which can be used in conjunction with the assembly task breakdown to yield a classification framework. This framework is then utilized to review the most common UI practices in AR literature. Finally, the third section assesses various interface elements from a theoretical point of view to suggest which are best suited to each of the assembly subtasks.

ASSEMBLY TASK BREAKDOWN

Decomposing the steps of the assembly tasks described in Chapter 2 was done using a user centered design approach. This approach involves first outlining an application scenario and a typical user, and subsequently establishing use cases. The physical assembly employed in this research is an axial piston motor provided by Sauer Danfoss, so the application scenario and user typical to this particular industrial context was used. Ultimately, the establishment of use cases will yield four assembly-step subtasks, each of which is addressed in its own subsection below.

The description of a typical assembly scenario and user is based primarily on info provided by Sauer Danfoss. The typical assembly environment involves the presentation of numerous parts in bins which are often lighted using LED's to aid in part selection. Sauer Danfoss cites confusion between similar parts as a common problem. Thus, many parts are designed such that they cannot be used in an incorrect configuration. Additionally, such features

often aid in understanding where to place the part. The typical assembler is estimated to have a high school education, though slightly higher education levels are not out of the question. Barring fresh recruits, all assemblers have some mechanical experience by virtue of working on Sauer Danfoss production lines.

With the above application scenario and typical user in mind, the cognitive walkthrough method was employed to establish use cases. In abbreviated form, the cognitive walkthrough of an assembly task asks two basic questions. First, will the user recognize that the correct action is available? And second, does the user understand that the correct action is necessary to achieve the overall goal? These two questions are used to develop four subtasks—part identification, indication of part manipulations, tool identification, and indication of tool usage. These subtasks typically occur in the sequence depicted in Figure 11. Note that a full cognitive walkthrough also concerns itself with feedback—that is, asking whether the user will know that he or she has taken the right action after performing it. However, because this research is interested in studying errors, including *dependent* errors, that question is disregarded in this investigation.

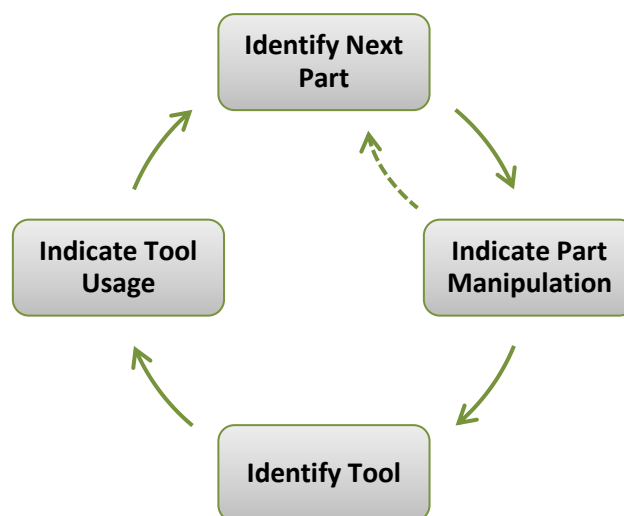


Figure 11. The four subtasks of the assembly task breakdown.

Part Identification

When performing an assembly operation, recognizing that an action is available to take (the first of the cognitive walkthrough questions) necessitates identifying the part or parts which the assembler should be interacting with. Making this identification is the first subtask, and deals *only* with the mental recognition of the relevant part or parts. As mentioned, Sauer Danfoss often utilizes lighted bins to this end. In some cases this subtask is very straightforward—consisting of the identification of a single, clearly unique part. In other cases several distinct parts could be very similar in appearance, making the identification process significantly more prone to error. Additionally, assembly steps can be grouped in such a way that multiple parts must be identified by the assembler at the same time. This would most often be the case if the same action is to be repeated several times using several individual but identical parts. Finally, subassemblies, which the assembler has already constructed, should be identified in the same manner as individual parts when they need to be interacted with.

Indication of Part Manipulation

Understanding how to manipulate a part (usually, where to place it) is an application of the second cognitive walkthrough question—that of understanding that the correct action must be taken to achieve the overall assembly goal. The simplest form of this subtask consists of moving the relevant part from its current location into a new location relative to one other part within the overall assembly. Note that even this basic manipulation necessitates that the assembler relocate the part to the correct location as well as rotate it into the correct orientation. More complex versions of this subtask can involve orienting the part relative to multiple other parts, manipulating more than one part at once, applying additional pressure or torque, and so on.

Tool Identification

The first cognitive walkthrough question leads to another subtask very similar to the first—the identification of a tool or tools. Like the first subtask, this subtask also consists only of the assembler’s mental recognition of which tool he or she will grasp. Again, this process could be straightforward, as in the case of identifying a single, highly-unique tool, but could also be more involved, as in the case of distinguishing between slightly differently-sized but otherwise identical tools. Extra care must be taken here to ensure that the assembler realizes that a tool must be used at all. The assembler will almost certainly expect that he or she must identify at least one *part* in every assembly step, but will not necessarily always be expecting to need to identify a *tool* as well.

Note that this subtask will be irrelevant to any assembly steps in which the usage of a tool is not required. In such cases, the second (previous) subtask of the current assembly step would be followed immediately by the first subtask of the subsequent assembly step as shown by the dotted arrow in Figure 11. It is also possible that in special cases more than one tool will need to be identified in a single step.

Indication of Tool Usage

The second cognitive walkthrough question also has a second application—that is, indicating what action the assembler should take using the previously identified tool or tools. Given that Chapter 2 restricted assembly types to those requiring only common handheld tools, this subtask will typically need to communicate only basic information such as the rotation of a wrench or the squeezing of pliers. However, it is conceivable that certain tools will be unfamiliar to some assemblers—particularly new trainees—and thus require thorough clarification to avoid

confusion. Note that the applicability of this subtask is contingent upon the applicability of the third subtask (tool identification).

If the information provided about all four subtasks is correctly understood and acted upon by the assembler, an entire assembly step will have been properly executed. As illustrated in Figure 11, the fourth subtask is then followed by the first subtask of the subsequent step.

UI ELEMENT HIERARCHY

To further develop a taxonomy of UI elements for AR assembly applications, a hierarchy has been developed to classify graphical information presentation elements into several categories. This hierarchy is combined with the assembly subtasks designated in the previous section in order to develop a framework for further classifying AR assembly interface elements. Additionally, the AR systems designed by a number of researchers—particularly systems which pertain to assembly and maintenance applications—are classified according to this framework in this section in order to characterize and classify the most common UI elements found in literature.

Hierarchy Structure

The primary classification in this hierarchy, illustrated in Figure 12, is the top-level distinction between *concrete* and *abstract* interface elements—which are further divided into six subcategories. For the purposes of this taxonomy, concrete elements are those that are specifically designed to closely mimic the real-world objects or actions that they represent.

Conversely, abstract elements communicate information in a manner that is principally symbolic or figurative.

The two element types which fall under the concrete designation are *3D models* and *animations*. Figure 13 shows 3D models being used for an assembly instruction application [24]. In this context, the 3D models subcategory refers exclusively to models which closely approximate the exact shape and size of physical objects within the real-world workspace—such as detailed CAD models. *Only* virtual 3D objects which explicitly mimic a real-world object are classified as concrete.

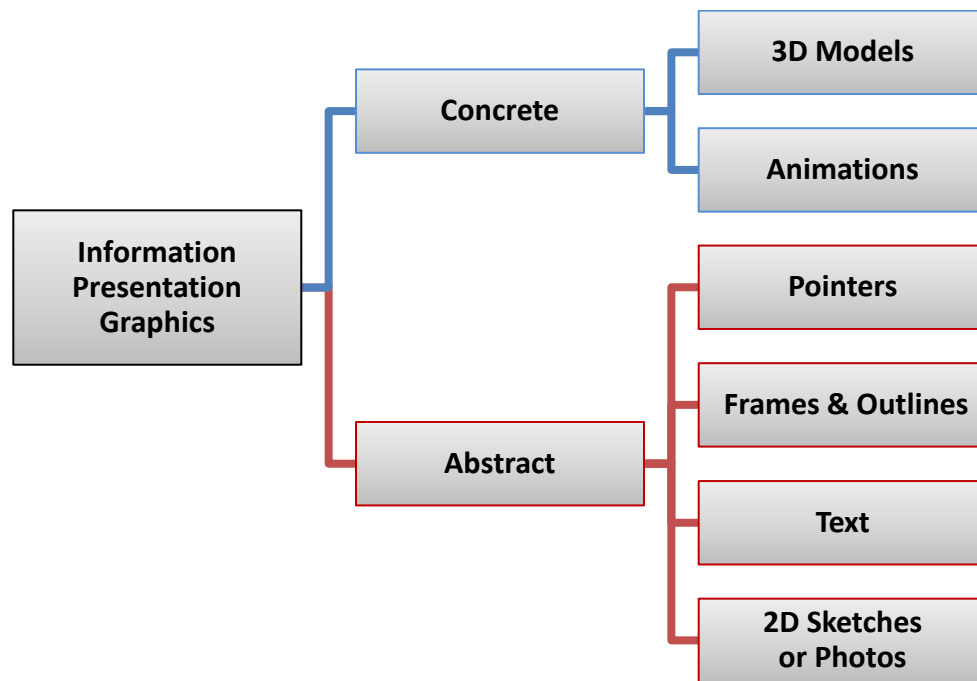


Figure 12. A hierarchy outlining the breakdown of GUI elements.



Figure 13. Example of using a 3D model to indicate part manipulation [24].

The animations subcategory falls under the concrete designation so long as the animations are intended to resemble actual motions which the assembler is meant to take—e.g. the insertion of a CAD model into a slot where the assembler is supposed to insert the corresponding physical part. Animations that do not fit this description—such as a “bouncing” animation meant to draw the user’s attention—do not fit into the concrete designation because they are directing a user’s attention to new information communicated by the graphic being animated rather than information conveyed by the animation itself.

The abstract designation consists of four subcategories—the first of which is *pointers*. A pointer is an object used to indicate direction or position symbolically. A classic and almost universally understood example is an arrow. Pointers are classified as abstract because they only indicate a real-world motion or direction symbolically, via a well-known shape.

The second abstract subcategory is *frames and outlines*. These can take several forms such as a 2D frame outlining a physical part, a 3D box bounding a particular area, or even individual lines meant to highlight a long, narrow region. Highly-detailed wireframe models

which closely resemble physical objects are more appropriately assigned to the concrete 3D models subcategory.

The *text* abstract subcategory is useful for communicating a wide range of information, and includes words or numerals displayed in a 2D format in a fixed position on the display as well as those associated with specific elements in the physical scene. Provided adequate space, the information being conveyed by text can be described very explicitly.

The final abstract subcategory is *2D sketches and photos*. Photos can depict parts which are relevant to an assembly step, or may be taken of an assembler performing an assembly task. 2D sketches are typically line drawings intended to symbolize real-world objects or actions. As these descriptions illustrate, the distinction between concrete and abstract is less well-defined within this subcategory than any other; but the abstract designation is the correct one. The best way to highlight this classification is to consider a “perfect” AR system, in which virtual objects that are meant to mimic physical objects are, in fact, completely indistinguishable from the real-world. In such a setting a 2D sketch or photograph would not appear to be a concrete augmentation, but would rather be seen as a supplementary representation. The element within this subcategory that comes closest to meeting the concrete criterion is a photograph used as a billboard—that is, having a specific location in the 3D world and rotating such that the photograph is always perpendicular to the user’s line of sight. In this case, however, the photograph is not being treated as a 2D image but rather in a manner very similar to a low-fidelity 3D model—which is a concrete element.

Hierarchy Application

Table 2 illustrates the above hierarchy (along the left axis) being applied to assembly scenarios in general in light of the four previously outlined assembly subtasks (along the top axis). The two classification schemes are combined based on the notion that any pairwise link between a hierarchy class and an assembly subtask is at least partially a valid option for information communication. All 32 of the resulting pairwise matchups are given a cell identifier in Table 2 for use in later tables. Note that in this section this framework is used only to reflect on existing AR literature. It will be applied to the design of new interfaces in the final section of the chapter.

Table 2. Pairwise matchups of hierarchized GUI elements and assembly subtasks.

		Identify Part	Indicate Part Manipulation	Identify Tool	Indicate Tool Usage
Concrete	3D Models	A1	B1	C1	D1
	Animations	A2	B2	C2	D2
	Other	A3	B3	C3	D3
	<i>Count</i>	12	14	1	1
Abstract	Pointers	A4	B4	C4	D4
	Frames & Outlines	A5	B5	C5	D5
	Text	A6	B6	C6	D6
	2D Sketches & Photos	A7	B7	C7	D7
	Other	A8	B8	C8	D8
	<i>Count</i>	31	41	1	6
	<i>Total Count</i>	43	55	2	7

The rows in Table 2 which are labeled *Count* and *Total Count* reflect the information that has been gathered from a number of AR research efforts about how the interfaces used in their systems fit into this overall framework. These counts are intended to give an overall impression of current common practices as well as to illustrate how the framework can be used.

Further detail on each of the pairwise matchups, including how the exact counts were arrived at, is shown in four more tables (Tables 3-6)—each pertaining to one of the four assembly subtasks. The left axes list all of the categories from the graphical interface element hierarchy, while the top axes indicate the relevant cell identifier from Table 2, the citations for research systems which utilize elements from each category, and the total number of sources cited in the table row. The sources cited in these tables all address at least one of the following domains: AR for assembly, AR for maintenance tasks, or AR for assembly planning. Table 3 deals with the part identification subtask (cells A1-A8), Table 4 deals with indication of part manipulation (cells B1-B8), Table 5 deals with tool identification (cells C1-C8), and Table 6 deals with indication of tool usage (cells D1-D8).

Table 3. Source list for assembly breakdown task A—Identify Part.

		Cell	Sources	Count
Concrete	3D Models	A1	[5][6][18][24][33][34] [35][36][37][38][39]	11
	Animations	A2	<i>*Comments in text</i>	0
	Other	A3	[40]	1
Abstract	Pointers	A4	[10][18][23][33][34][36][41][42]	8
	Frames & Outlines	A5	[23][24][40][43][44]	5
	Text	A6	[4][7][18][24][33][37] [38][42][44][45][46]	11
	2D Sketches & Photos	A7	[12][20][24][36][42][45]	6
	Other	A8	[18]	1

Table 4. Source list for assembly breakdown task B—Indicate Part Manipulation.

		Cell	Sources	Count
Concrete	3D Models	B1	[5][6][10][18][24][33][34] [35][37][38][41][44]	12
	Animations	B2	<i>*Comments in text</i>	0
	Other	B3	[24][40]	2
Abstract	Pointers	B4	[4][5][6][10][12][18][20][21] [24][33][34][38][41][42][45]	15
	Frames & Outlines	B5	[4][5][21][33][34][35][41][42][47]	9
	Text	B6	[4][7][12][24][33][34][36] [39][40][41][42][44][45]	13
	2D Sketches & Photos	B7	[20][34][36][45]	4
	Other	B8	-	0

Table 5. Source list for assembly breakdown task C—Identify Tool.

		Cell	Sources	Count
Concrete	3D Models	C1	[18]	1
	Animations	C2	<i>*Comments in text</i>	0
	Other	C3	-	0
Abstract	Pointers	C4	-	0
	Frames & Outlines	C5	-	0
	Text	C6	-	0
	2D Sketches & Photos	C7	[24]	1
	Other	C8	-	0

Table 6. Source list for assembly breakdown task D—Indicate Tool Usage.

		Cell	Sources	Count
Concrete	3D Models	D1	[18]	1
	Animations	D2	<i>*Comments in text</i>	0
	Other	D3	-	0
Abstract	Pointers	D4	[10][36][41][46]	4
	Frames & Outlines	D5	-	0
	Text	D6	[36][41]	2
	2D Sketches & Photos	D7	-	0
	Other	D8	-	0

Several observations can be made based on the counts given in Tables 2-6. The most readily obvious is the fact that the vast majority of AR assembly research either does not involve the use of tools, or the corresponding publications neither discuss nor depict the interface elements used to communicate information about tools. Also readily evident is the trend that most researchers prefer abstract interface elements over the concrete. This trend should be noted in conjunction with the reality that concrete interface elements are typically more difficult and time-consuming to implement.

Another clear trend is what appears to be a complete lack of concrete animation usage. However, this trend is almost certainly inaccurate thanks to the method by which these counts were compiled—that is, the reading of AR research publications. Most such publications choose not to discuss interface element usage in the text, electing instead to provide a handful of screenshots to give the reader a general impression of the GUI employed. This practice does not present a problem for observing the majority of interface elements since most can be classified satisfactorily while static. Animations, however, cannot be observed in the same fashion and therefore lack a precise count.

Overall, the most common elements for part identification are text and 3D models comprising a tie, while the most popular method for indicating part manipulation is the use of pointers. Tool considerations, if addressed at all, are often handled via pointers as well.

INTERFACE EVALUATION

This section will utilize the framework laid out in the previous two sections to assess the merits of each type of interface element as an information communication method for each of the

four assembly subtasks. In each case a theoretical best element or combination of elements from both the concrete and abstract designations will be chosen, and the better choice between the two designations will be selected as well. This procedure involved creating a “picture prototype” illustrating various ways in which each interface element could potentially be used to address each assembly subtask. Design principles for AR interfaces such as those proposed by Dünser et al. are also applied to support the decision making process.

Part Identification

For the concrete designation, 3D models will be the best method for part identification in nearly every scenario. The human visual system is very adept at identifying objects by shape as well as subsequently recognizing other, similar objects. As long as disparate objects do not appear very similar to each other, the depiction of a 3D model will typically allow a user to quickly and accurately determine which part he or she is meant to interact with. Animations have very little to offer this subtask, since they are more well-suited to demonstrating motion than they are to describing the nature of a static object.

Within the abstract designation several elements are plausible part identifiers, but frames and outlines are selected here as the most efficient. Not only is the bounding of an object an excellent way to single it out quickly and accurately, but the user can discern additional information about the size and shape of the part via the scale and aspect ratio of the frame. Pointers are also quick and effective elements for indicating a part location, but lack the additional spatial cue provided by frames. Text and 2D sketches and photos take more time to process than either of the other two alternatives and therefore should only be utilized for this

subtask when subtle distinctions between different parts cannot be discerned by frames or pointers alone.

Overall, between 3D models and frames, the best choice is entirely situation dependent—with models being more effective for training applications and frames being more effective for routine assembly or maintenance support. Part identification via 3D models requires the user to make a visual comparison of the component's features between the model and the physical part. This comparison takes time and is therefore undesirable in an application meant to support an experienced worker, but also aids learning by forcing the user to take note of the part's prominent features and is thus beneficial to the training of a user who is unfamiliar with the assembly. The rapid part identification offered by frames is highly desirable for a scenario where the user is already familiar with the basics of the assembly and only needs AR support to help him or her locate the relevant components more quickly and reliably.

Indication of Part Manipulation

Within the concrete designation, the combination of both 3D models and animations is the most useful method for indication part manipulation. Again, humans are very accustomed to determining the location and orientation of an object visually, and will be able to do so very effectively with 3D models provided that there are salient cues available rather than small, difficult-to-recognize variations between multiple possible orientations. Then, with the proper location and orientation indicated, the addition of animations can communicate the path by which the part should reach its final location. By combining both of these interface elements the entire part manipulation procedure can be illustrated quickly and explicitly to the user. In cases where a final position or orientation is complex or otherwise difficult to understand, 3D models

of components which have already been placed on the assembly, but are occluded by other parts, can be augmented in the correction positions relative to the model of the new part. Such supplementary models serve a fundamentally different purpose—that is, a point of reference within the assembly rather than an object which is to be actively manipulated—and therefore should be distinguished via some visual attribute such as color or transparency. Such distinguishing attributes will guide the user in interpreting these extra models as serving an auxiliary purpose.

Under the abstract designation pointers are the best choice in most cases, and should often be combined with text to explicate details of the part manipulation. The selection of pointers is based on their ability to simultaneously indicate a location and a direction. This dual functionality allows them to communicate both the final part location and the path by which it should reach that location in a concise manner. However, the orientation which the part should take once it reaches that location is often impossible to illustrate clearly with pointers alone, and thus text can be utilized in conjunction. Brief phrases such as “insert narrow end first” or “white side down” are often all that is required to complete the indication of part manipulation. 2D sketches and photos may be useful if an orientation is unusually complex and therefore difficult to describe textually, but such situations are considered rare enough that 2D sketches and photos should not be considered the primary method of communication regarding part manipulation. Frames and outlines can be useful for indicating part location; however these lack any information about orientation or direction. Additionally, the use of frames for both part identification *and* manipulation indication would force the user to expend additional mental effort distinguishing the role of the various frames present in each assembly step.

Across the concrete and abstract categories, 3D models with animations will be the best choice in the vast majority of scenarios. This stems from the fact that a single object—the animated 3D model—communicates the entirety of the necessary information to the user, making the method extremely concise. Pointers with supplemental text, while also concise, cannot convey the same level of detail as quickly or reliably since they lack the same intuitive visual cues. The only situations where this selection will sometimes fall short are those where particularly subtle part features will not be immediately obvious when communicated by a 3D model, and thus could lead to errors in orientation. In such cases a second stage of animation can be an effective technique for communicating the necessary details to the user, such as the rotation of a part after it has already reached its target location. However, because complex assembly steps such as these require the user to slow down and expend additional mental effort regardless of the interface used, the ability of text and other abstract elements to be especially explicit makes them a strong choice under exceptional circumstances.

Tool Identification *and* Indication of Tool Usage

The discussions regarding the best interface elements used to identify tools and indicate tool usage have been combined into a single subsection because a single argument applies to both—namely, that consistency with the previous two assembly subtasks should be implemented whenever possible. Maintaining consistency is strongly advocated by Dünser et al. [1].

Preserving internal consistency dictates that if frames are employed for part identification then they should be used for tool identification as well. A user who has become accustomed to looking for frames when determining which objects are relevant to the current assembly step should not be expected to change his or her thinking when those objects happen to be tools.

Similarly, if 3D models and animations are used for indication of part manipulation, they should be for indication of tool usage as well. A user who has been inspecting animated 3D models in order to determine how to manipulate a part will be expecting to do the same when determining how to manipulate a tool.

However, a minor break in consistency should also be leveraged in order to give the user a cue that information is being communicated which pertains to a tool rather than an assembly component. Tools serve a fundamentally different purpose than assembly components themselves and the interface elements which deal with tools should reinforce this distinction in the user's thinking. A minor break in consistency can be achieved by varying a single visual attribute of the interface element such as color or transparency. A detailed investigation of the merits of various visual attributes involves a deeper level of abstraction within the interface element hierarchy which is outside the scope of this research.

CHAPTER 4 – EXPERIMENTAL DESIGN AND SETUP

In order to assess the relative efficiency of the user interface elements discussed in the previous chapter, as well as to investigate the merits of AR for assembly as opposed to other instructional methods, a user study was conducted. This chapter is dedicated to outlining the setup and implementation of that user study by describing the hardware utilized, the experimental hypotheses and design of accompanying interfaces, and the logistical study structure.

APPARATUS

Hardware

The AR system supporting this user study was implemented on a single workstation, the arrangement of which can be seen in Figure 14. The primary components consisted of two LCD displays—one each for the subject and the experimenter—an overhead webcam, and a desktop computer. The LCD displays for the subject and experimenter measure 24” and 23”, respectively. The webcam is a Logitech Pro 9000—a common consumer-grade model. The desktop computer has the following specifications:

- Intel Xeon 3.47GHz CPU
- NVIDIA Quadro 5000 video card
- 6.00GB RAM
- Windows 7 Enterprise 64-bit OS

The user’s monitor was placed just above the physical assembly space, allowing him to look up slightly in order to quickly reference the augmented scene from a viewpoint closely approximating his direct view of the physical scene. The AR system performed tracking via four

fiducial markers attached to the workbench, the assembly base, and the base components of two subassemblies.



Figure 14. The workstation used to conduct the user study.

The test product utilized by this study was an axial piston motor, which the users were required to assemble—shown in its fully assembled state in Figure 15. This unit was provided by Sauer Danfoss and is a hydraulically controlled, variable (L/K series) motor. The entire unit measures roughly 7” tall, 7” wide, and 9” long including the output shaft, and weighs 34 pounds.



Figure 15. The axial piston motor assembly provided by Sauer Danfoss.

The tools necessary to assemble the motor were an 8mm hex key (also commonly referred to as an Allen wrench) and a pair of snap ring pliers. Each tool was used for exactly one assembly step—namely the securing of the motor end cap with five cap screws and the installation of a snap ring around the output shaft. The remaining 14 assembly steps were all performed by hand, and are described and illustrated by the three sets of assembly instructions which comprise Appendices B, C, and D—the first of which is a paper instruction manual written based on advice from Sauer Danfoss, while the latter two consist of screenshots of the AR interfaces discussed in the next section of this chapter.

To perform *complete* disassembly and reassembly of the motor would require several additional specialized tools, making the full procedure impractical for the user study. As such, a small number of assembly operations were excluded. The primary exclusions involved two sets of bearings which cannot be removed or reinstalled without a bearing puller or bearing press, or other similar tools. The two bearing sets include a ball bearing mounted on the motor's central shaft, and two journal bearings mounted on the inside of the motor housing. For the study, all of these bearings were treated as if they were permanent features of the parts to which they were attached. From an information presentation point of view no opportunities were lost due to these exclusions since bearing installations are insertion tasks, of which this assembly included many.

The second exclusion involved a rubber seal which is oriented around the motor's output shaft. Sauer Danfoss advised that this seal only be removed using a slide hammer—a process which effectively destroyed the seal. Therefore the installation of the seal over the output shaft was ignored for the study, along with the snap ring and circular plate which hold the seal in place. Again, no unique information presentation opportunities were lost due to this exclusion since the circular plate and seal installations comprise yet another pair of insertion tasks and the

snap ring installation is an exact duplicate of a prior task in which a snap ring secures the motor's output shaft.

Software

The AR software employed for this user study was an application entitled ARMaker, designed by Rafael Radkowski. This application utilizes ARToolkit and OpenSceneGraph, and enables its users to very easily create multiple AR scenes containing numerous virtual elements and to integrate those scenes in any sequence. Authoring of such scenes involved writing an XML script to load models or images, specify positions and animations, and link objects to AR markers. An additional Python script was also written to control switching between scenes and other application logic. The XML and Python scripts written for this study are included in Appendices E and F, respectively.

For the ARMaker software to function properly and provide accurate registration of the augmented elements, the system must be calibrated for the webcam being used. Calibration was carried out using the GML Camera Calibration Toolbox [48]. Completion of the calibration procedure allowed for the webcam's focal length and principal point—intrinsic values of the camera itself—to be communicated to the ARMaker application.

HYPOTHESES AND INTERFACE DESIGN

This section deals with the design of two interfaces for use in the user study as well as several research hypotheses which are tied to various interface features. All of the hypotheses

will be presented in the first subsection, while the subsequent two subsections will provide information about the interfaces which have been designed to test those hypotheses.

These hypotheses will reflect the three test cases which comprise the overall study—concrete AR (CAR), abstract AR (AAR), and paper-based instructions (PBI). The design of the two AR interfaces is outlined in the two subsections following the hypotheses subsection. Further information about all three test cases is given in the final full section of the chapter.

Hypotheses

In this subsection three hypotheses will be proposed, all of which deal with assembly times and error rates in some form.

Hypothesis (1) posits that abstract AR elements are more suitable for part and tool identification than concrete AR elements alone. Within this study this comparison is made using frames and 3D models, but other abstract versus concrete comparisons are viable. This hypothesis is based on the logic that although visually comparing 3D models to physical parts is very intuitive—as described in Chapter 3—bounding the physical parts with a frame requires no visual comparison at all and is therefore both faster and more reliable.

Hypothesis (2) posits that concrete AR elements are more suitable for indication of part and tool manipulation than abstract AR elements. Within this study this comparison is made using 3D models and animations as opposed to primarily, but not exclusively, pointers and text, but other abstract versus concrete comparisons are viable. The rationale behind this hypothesis also rests on the principles described in Chapter 3, wherein 3D models and animations were shown to be extremely concise indicators of complex manipulations, while abstract elements either provide very sparse information (as in the case of pointers alone) or run the risk of

cluttering the display and bewildering the user (as in the case of pointers, detailed text, and 2D sketches all being used simultaneously).

Hypothesis (3) expands to include paper-based instructions and posits that, for the types of interfaces utilized by this study, total assembly times will be fastest using concrete AR, followed by paper-based instructions, and slowest for abstract AR. The logic behind this ordering is twofold. First, an interface utilizing only abstract AR elements was anticipated to suffer from the limitations regarding part and tool manipulation information previously described in defense of Hypothesis (2). This was expected to cause users to spend more time attempting to interpret assembly instructions. Second, while concrete AR and paper-based instructions were both expected to communicate effectively, it was anticipated that concrete AR elements would do so very quickly in comparison to the time needed to read detailed textual information.

Hypothesis (4) addresses training, and posits that AR is more suitable than paper-based instructions for assembly and maintenance training. This hypothesis is founded on the concepts cited in Chapter 2 which assert that presenting information in a physical location aids in memorization, allowing for that information to be retrieved during later assembly tasks. Concrete and abstract AR interface elements are both capable of collocating information in this way, while paper-based instructions are not.

Concrete Interface

Because several of the research hypotheses pit concrete AR elements against abstract AR elements, two full sets of AR instructions were designed making use of only one or the other.

The concrete interface utilizes both of the two types of concrete AR elements—3D models and

animations. Users were guided through assembly steps sequentially, and screenshots spanning the entirety of the concrete interface instructions are provided in Appendix C.

In addressing the part identification subtask, 3D CAD models provided by Sauer Danfoss were augmented into the scene for every step. Each of the assembly components was deemed unique enough in shape to be easily distinguishable based in the visualization of a CAD model alone. Identification of the two tools—the 8mm hex key and the snap ring pliers—was also handled utilizing 3D models—in this case two publically available tool models.

Both the indication of part manipulation and tool usage subtasks were handled via the addition of animations to the same 3D models. Each model was shown moving from an initial position, away from the overall assembly, into the location and orientation where the assembler was meant to manipulate it. The path between the two animation endpoints was made to resemble the path which the user should take as well. Also, in order to ensure that the user was able to process all of the relevant information, animations were designed to repeat in an endless cycle until the user moves on to the next step. This tactic reflects the progressive disclosure of information-rich interfaces that Gabbard advises based on the work of Hix and Hartson [27][31].

In keeping with another of Gabbard's principles, this one based on the work of Wloka and Anderson [27][29], proper occlusion was maintained for 3D models using a phantom model of the large motor housing. In two cases, shown in Figure 16 and Figure 17, it was deemed important for an occluded part or parts to be made visible in order to provide a reference location. Two such components were parts which had already been placed on the assembly, and were thus differentiated from other 3D models by altering another visual attribute—increased transparency. The first (Figure 16) was crucial in order to illustrate that the arm of the swashplate component must mesh with the previously-inserted servo piston. The second (Figure 17) was

important in order to make the user aware that the central shaft onto which the cylinder block subassembly was being placed had teeth midway along its surface, which must be aligned with the subassembly being inserted.

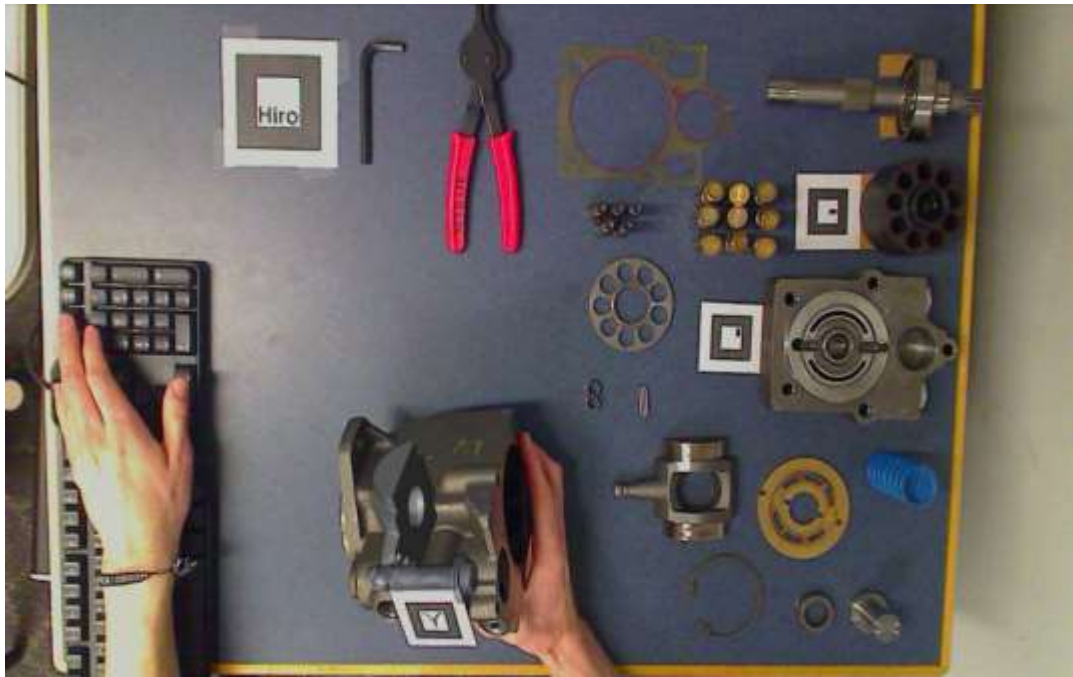


Figure 16. Screenshot of the semitransparent servo piston.

As mentioned in Chapter 3, it is important to differentiate assembly components from tools in the user's interpretation of the interface elements—often by altering visual attributes of the elements. For the concrete interface the visual attribute which was altered was color. Components were left in their default medium gray while the tools were colored red.



Figure 17. Screenshot of the semitransparent toothed shaft.

Abstract Interface

The design of the abstract interface made use of all four elements to some degree, but relied primarily on pointers and frames. Users were guided through assembly steps sequentially, and screenshots spanning the entirety of the abstract interface instructions are provided in Appendix D.

The part identification subtask was fulfilled exclusively by frames. As Chapter 3 outlined, this approach quickly and unambiguously communicates which component the assembler is meant to interact with. For the same reason, frames were also the only element used to address the tool identification subtask. In order to differentiate tool identification frames, the color attribute was changed to red and a subtle abstract, attention-directing, bouncing animation was introduced.

Pointers were utilized, in part if not in whole, to fulfill the indication of the part manipulation subtask in every single step. This decision reflects an effort to maintain interface consistency, as advocated by Dünser et al. [1]. Specifically, the classic arrow was utilized to indicate not only where each component should be placed, but also the direction from which it should be brought into that location.

In cases where the above two elements were insufficient to describe an assembly step, text was used to provide additional information pertaining to part manipulation. An excellent example of this is depicted in Figure 18. The text in this figure provides both the basic information about which side of the cylinder block subassembly should be inserted into the housing first, as well as the more complex information about aligning the teeth of multiple components.



Figure 18. Screenshot of additional textual instructions.

When an assembly operation was unusual enough that the addition of text was still deemed insufficient to convey the necessary information, a 2D sketch was added. This is illustrated in Figure 19. The sketch illustrates the path by which the swashplate can be guided into the housing (which is further specified by a secondary arrow augmented alongside the housing itself) as well as the critical meshing of the swashplate arm and the servo piston.



Figure 19. Screenshot of an additional 2D sketch.

Additional interface elements pertaining to tool usage were not implemented in the abstract interface. As Figure 18 and Figure 19 demonstrate, the use of multiple abstract elements add to an already somewhat cluttered display, and the extra elements used to convey to the assembler that a tool is necessary to the step would add to this clutter even more. Such information overload tends reduce the information absorbed by the user rather than increase it. Thus, adding still more elements regarding tool usage was deemed more harmful than helpful.

For this study the usage of these tools was considered straightforward and it was reasonable to expect that the assembler would be able to perceive their usage without any additional prompts.

USER STUDY STRUCTURE

This user study was a between-subjects study, consisting of 33 participants split into three test cases via random assignment. The participants were recruited initially via several Iowa State University email lists, and some recipients of the recruitment emails were followed up with in person. The email lists used included the mechanical engineering undergraduate student list, the mechanical engineering graduate student list, and the mass mail list for the Virtual Reality Application Center (VRAC) and the HCI graduate student program. Via word of mouth a handful of participants from other fields of study, such as industrial and agricultural engineering, volunteered as well. The average age of the participants was 22.5, and the gender breakdown was 63.6% male, 36.4% female.

Splitting participants into the CAR, AAR, and PBI test cases was initially done via true random assignment using a six-sided die. Each of the three cases was associated with two opposing sides of the die, which was rolled prior to the arrival of each participant. However, over the course of the study preliminary results began to suggest that more finely grained data was needed, leading to the implementation of an additional information gathering technique as described later in this subsection. In order to ensure that the new, more highly-detailed information was gathered in sufficient numbers for both AR cases, the final 10 participants alternated evenly between CAR and AAR.

Upon arrival, each volunteer was brought individually to the study workstation (pictured in Figure 14 in the first section of this chapter) which was located in a secluded space free from distractions. After reading and signing the informed consent document, included in Appendix G, volunteers were given a brief pre-questionnaire to record age, sex, level of education, and field of study or profession. This questionnaire is included in Appendix H. Depending on the test case to which the volunteer had been assigned, he or she was told that assembly instructions would be presented either on the LCD screen in front of them or via a paper instruction manual.

Participants were also told that each on-screen step or each page corresponded to a single assembly step, and that their goal was to finish the entire assembly process as efficiently as possible while still taking as much time as necessary to feel that the information being presented was fully understood. Additionally, participants in the AR test cases were told that the left and right arrows on the keyboard would be used to navigate between assembly steps, and that they should avoid occluding the paper markers from the camera's view. After confirming that the volunteer had no further questions the AR software was initialized or the participant was handed the paper instruction manual, a stopwatch was started, and the volunteer began assembly.

During the assembly procedure the experimenter observed the participant's actions in order to write down any errors committed, as well as any repeated tasks or other events deemed noteworthy, according to the following protocol. Errors were classified according two criteria very similar to those employed by Gharsellaoui et al. [44]. First, errors were designated as either *errors of part* (EoP), in which the subject grasps with his or her hand the wrong assembly component or tool, or *errors of orientation* (EoO), in which the subject attempts to place a component or use a tool in an incorrect location or with the wrong orientation. Errors were also classified based on whether the participant *attempted* to take an erroneous assembly step or

succeeded in taking an erroneous step. An erroneous step was classified as successful if the subject moved on to the next assembly step without recognizing his or her mistake. Recognizing the mistake during a subsequent assembly step and fixing the error did not prevent the original error from being classified as successful. Finally, a repeated task was defined as any performance of an assembly step more than once. Repeated tasks often resulted from discovering previously made errors, but also occurred occasionally when a participant appeared unsure about the step which he or she was currently undertaking.

Data regarding assembly times was also gathered via two methods. Initially, a stopwatch was used to assess the total time taken to complete the assembly procedure. As mentioned previously, it was later recognized that data regarding not only total assembly time but also the time spent on each individual step would be beneficial. Thus, for the final 10 participants, the Python script was altered to write to a file the times at which the user switched between assembly steps in the AR application. This alteration was entirely invisible to the application user, and therefore did not necessitate that the data gathered from the final participants be analyzed separately from that of the previous participants.

Once the assembly procedure was complete, the participant was immediately given a post-questionnaire in which he or she could give feedback about the assembly process and the assembly instructions. The entire post-questionnaire is provided in Appendix H following the pre-questionnaire. The primary questions involved providing rankings on a five-point scale regarding the difficulty of various portions of the task (such as where to place parts, how to orient parts, when to use tools, etc.), the helpfulness of the assembly instructions, and the participant's confidence both before and after engaging in the assembly process. Two short-answer questions also asked for any previous experience that the participant had with assembly

tasks similar to the one undertaken for the study, as well as any general feedback that he or she may wish to give. The last page of the post-questionnaire was used exclusively by the experimenter to record total assembly time and any information about errors, and was always removed prior to giving the questionnaire to the study participant.

CHAPTER 5 – EXPERIMENTAL RESULTS AND DISCUSSION

The raw results for average total time and average overall error rate are depicted in Figure 20 and Figure 21, respectively, which feature the three test cases along the horizontal axes, total time in seconds or total number of errors committed on the vertical axes, as well as minimum and maximum values illustrated using error bars. However, the data depicted includes that of nine participants who failed to complete the assembly as intended—whether knowingly or not. Since these users failed to complete the assembly this data is not necessarily comparable to that of those who did. Drawing only from the data of those who completed the entire assembly as intended yields the alternate completion time chart shown in Figure 22 and the alternate error rate chart shown in Figure 23.

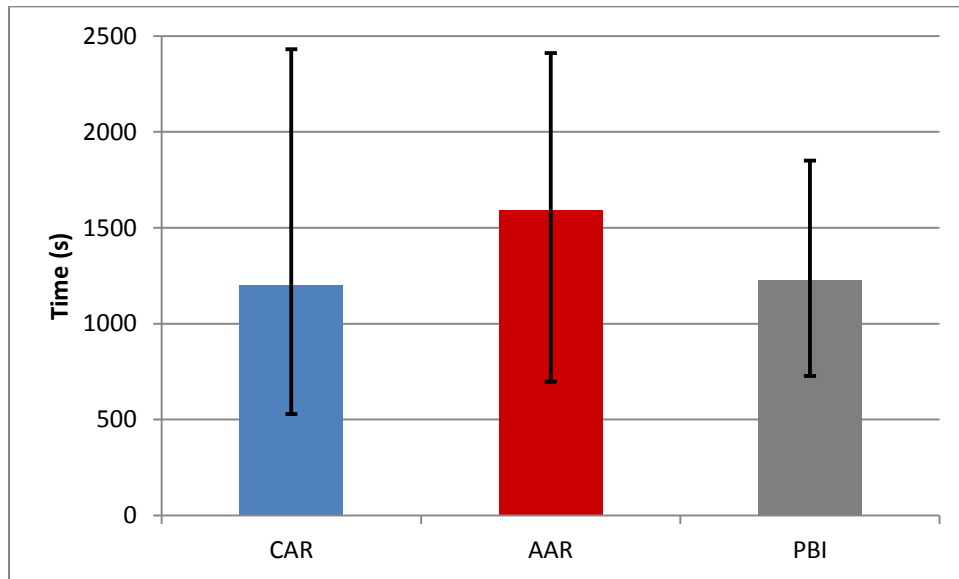


Figure 20. Average completion times with maximum and minimum values.

The number of participants who failed to complete the assembly correctly for the AAR case was five, for the CAR case four, and for the PBI case zero. The absence of failures in the PBI cases can almost certainly be largely explained by the fact that the paper-based instructions themselves included pictures of the assembly *after* the performance of every individual step. These pictures allowed PBI participants to make a post-task evaluation that CAR and AAR participants were unable to perform.

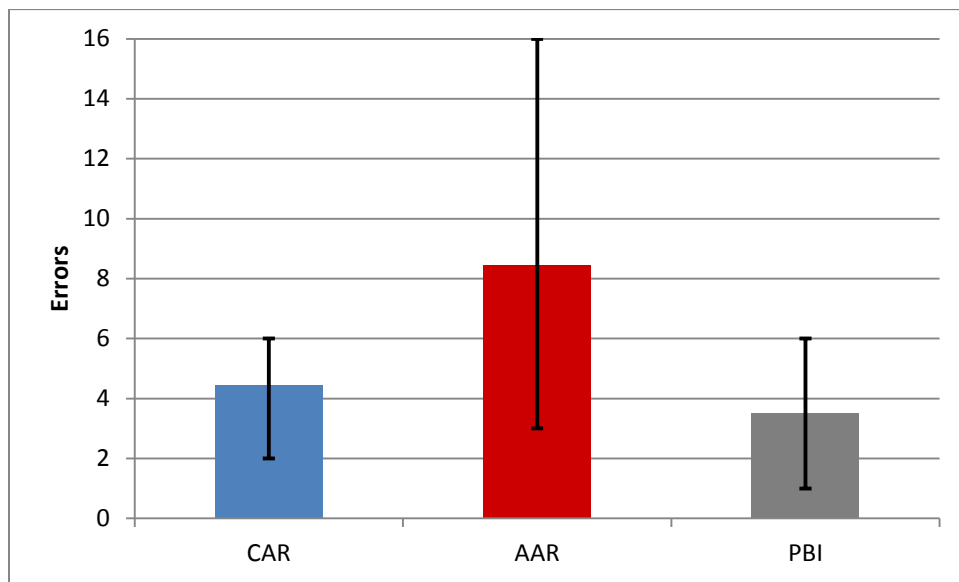


Figure 21. Average total error rates with maximum and minimum values.

An initial appraisal of the results themselves reveals that the AAR case saw the highest average error rates across the board, as well as the longest average completion times both with and without data from participants who failed to complete the assembly as intended. More in-depth scrutiny reveals that Hypothesis (1) was not supported while Hypotheses (2), (3), and (4) were all at least partially supported. A breakdown of the methods used to assess each hypothesis, as well as an explanation of the statistical analyses, follow in the next four sections.

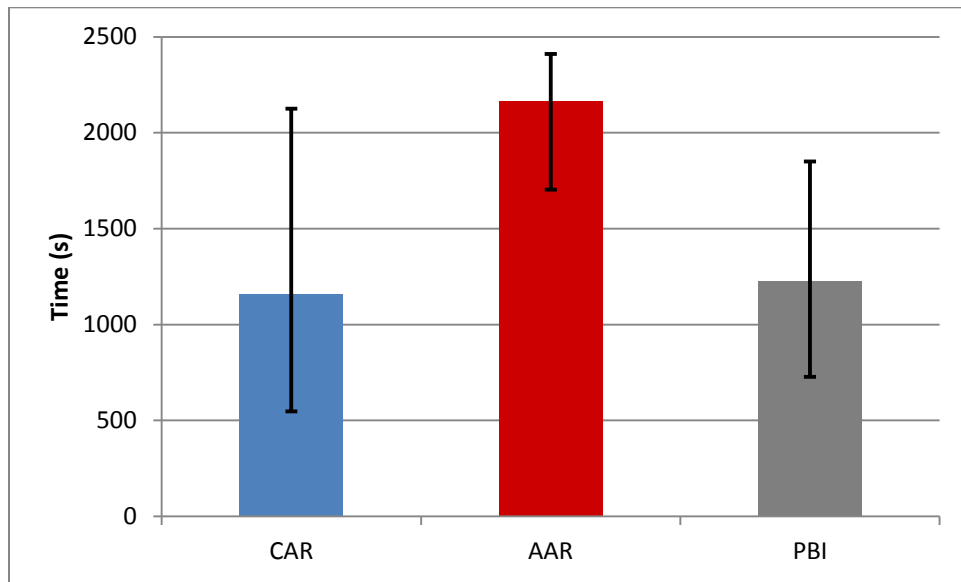


Figure 22. Average completion times for correct assemblies only with maximum and minimum values.

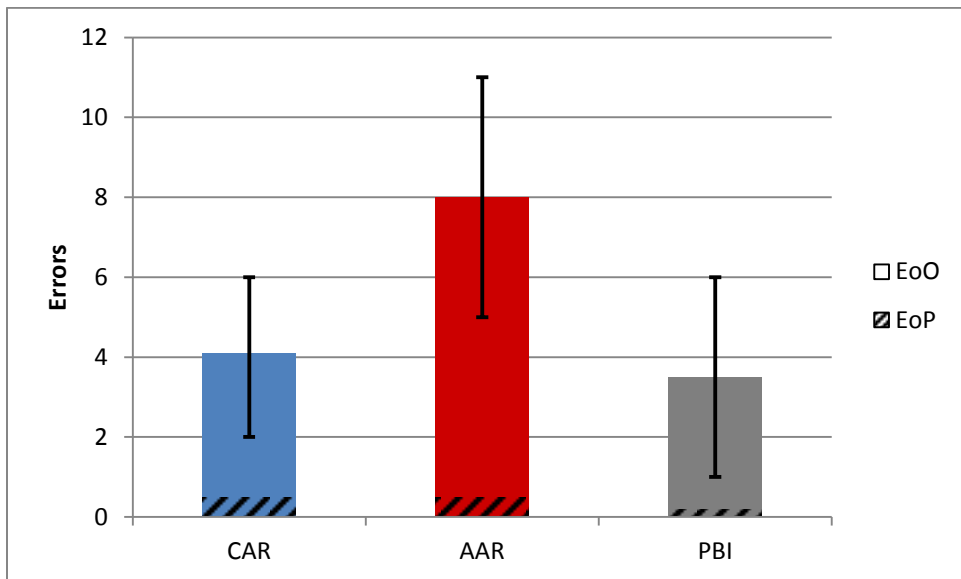


Figure 23. Average orientation error and part error rates for correct assemblies only with maximum and minimum values shown for total errors.

HYPOTHESIS (1)

Hypothesis (1) posited that abstract AR elements (in this case frames) are more suitable for part and tool identification than concrete AR elements (here, 3D models) alone. This hypothesis was evaluated by examining specifically errors of part (EoP)—that is, errors in which an assembler grasped the incorrect part with his or her hand during an assembly step. The EoP rates for both cases are depicted using a slashed pattern in Figure 23. As the figure illustrates, very few EoP's were committed under any test case.

Given that the EoP average was lower for the CAR case than the AAR (opposite from what Hypothesis (1) proposed) statistical significance was sought only to establish any difference between the two cases. Representing the EoP rate for the CAR and AAR cases as EoP_{CAR} and EoP_{AAR} , respectively, the null and alternative hypotheses are as follows:

$$H_0: EoP_{CAR} = EoP_{AAR}$$

$$H_a: EoP_{CAR} \neq EoP_{AAR}$$

An F-test was conducted between the two EoP data sets to determine the likelihood of equal variance between the sets (homoscedasticity). This test found that $F(8,13) = 1.223$, $p > 0.3$, dictating that the data be treated as homoscedastic. Consequently, a two-tailed t-test for two-sample, homoscedastic data was conducted. The results of this test were $t(21) = 0.123$, $p > 0.8$, and thus vastly insufficient to reject the null hypothesis.

There are two primary explanations for this outcome. First, the poor performance of the AAR instruction for indication of part manipulation, which will be explored further in the subsequent subsection, led many users to question whether they were in fact interacting with the correct part. These uncertainties were made clear when many AAR participants, upon struggling with part manipulation, set down the correct part and began to experiment with incorrect parts.

Had AAR part manipulation instructions been more easily understood, it is very likely that the corresponding EoP rate would have dropped.

Additionally, as mentioned previously, the number of EoP's committed was very low to begin with. The parts which comprise the motor assembly are all very unique, causing any advantage that AAR might have had over CAR in terms of part indication to be significantly diluted. An assembly which was composed of more similar, easily-confused parts would be more useful in making distinctions between CAR and AAR in terms of suitability for part identification.

HYPOTHESIS (2)

Hypothesis (2) posited that concrete AR elements (here, the combination of 3D models and animations) are more suitable for indication of part and tool manipulation than abstract AR elements (in this case primarily pointers and text, as well as 2D sketches). This hypothesis was evaluated first by examining specifically errors of orientation (EoO)—that is, errors in which an assembler attempts to place a part in the wrong location or with the wrong orientation. These error rates, which comprised the majority of the total error rates in both test cases, are depicted again in Figure 24. Additionally, this hypothesis was evaluated by examining the total time spent on assembly steps two and six (shown in Figure 25) which were observed to be especially prone to EoO's.

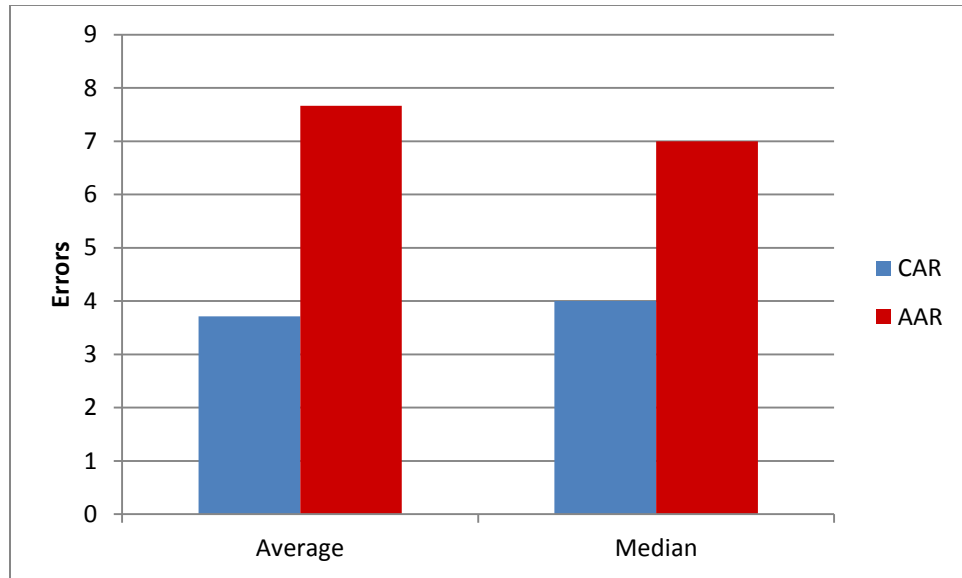


Figure 24. Errors of orientation.

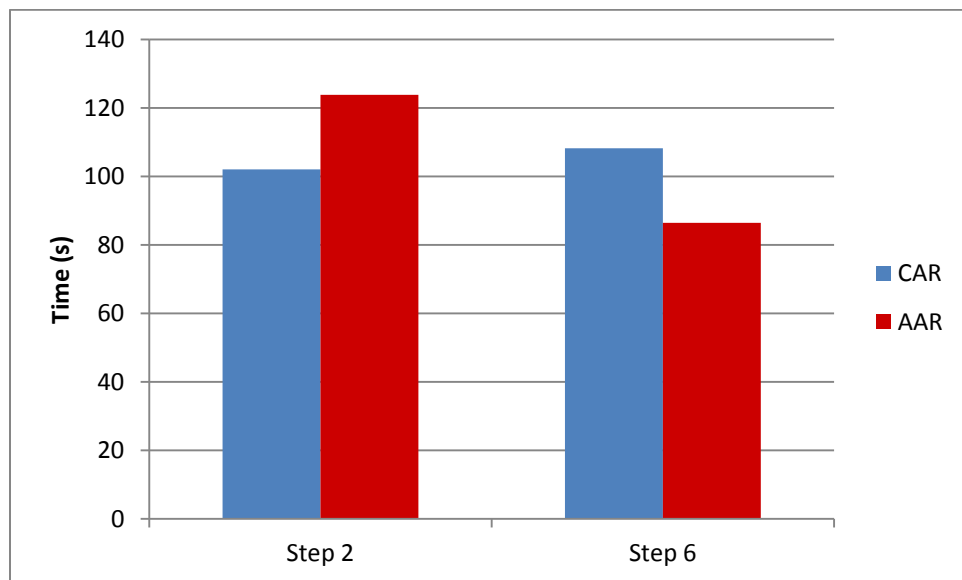


Figure 25. Average completion times for steps two and six.

Given that the average EoO rate was lower for the CAR case than the AAR, statistical significance for this relationship was sought. Representing the EoO rate for the CAR and AAR cases as EoO_{CAR} and EoO_{AAR} , respectively, the null and alternative hypotheses are as follows:

$$H_0: EoO_{CAR} = EoO_{AAR}$$

$$H_a: EoO_{CAR} < EoO_{AAR}$$

As before an F-test was conducted which found that $F(8,13) = 0.190$, $p > 0.9$, indicating homoscedasticity. Consequently, a one-tailed t-test for two-sample, homoscedastic data was conducted. The results of this test were $t(21) = 4.264$, $p < 0.01$, thereby rejecting the null hypothesis at a 99% confidence level. Thus it can be concluded with 99% confidence that the true average EoO rate for the CAR test case is lower than that for AAR test case.

The average completion times for steps two and six, however, do not agree with each other. Therefore, statistical significance was sought only to establish any difference between the two cases. Representing the completion times for steps two and six of the CAR case as $T_{CAR,2}$ and $T_{CAR,6}$ and of the AAR case as $T_{AAR,2}$ and $T_{AAR,6}$, the null and alternative hypotheses are as follows:

$$H_0: T_{CAR,2} = T_{AAR,2} \quad T_{CAR,6} = T_{AAR,6}$$

$$H_a: T_{CAR,2} \neq T_{AAR,2} \quad T_{CAR,6} \neq T_{AAR,6}$$

Two F-tests found $F(4,4) = 1.645$, $p > 0.3$ for step two and $F(4,4) = 0.270$, $p > 0.8$ for step six, indicating homoscedasticity for both. Consequently, two-tailed t-tests for two-sample, homoscedastic data were conducted for both steps. The results of these tests were $t(8) = 0.585$, $p > 0.5$ for step two and $t(8) = 0.376$, $p > 0.7$ for step six—both insufficient to reject the null hypotheses.

The failure of the second evaluation method to support Hypothesis (2) while the first method succeeded can be explained partially by the small number of participants for which completion times of individual steps were recorded, as well as by very high variability. For instance, recorded completion times on step six alone were as low as 28 seconds and as high as 5.2 minutes. Achieving statistical significance from such a highly-varied population requires significantly more data than was collected by measuring only five participants from each case. However, despite the statistical challenges posed by high variability, the variability itself is an indicator that different users did not interpret the same information from the interface elements in the same way for these steps. This suggests that clear communication concerning complex operations like those involved in these two assembly steps would benefit from a multi-layered approach such as the combination of both concrete *and* abstract interface elements. Despite all this, the results of the EoO evaluation method continue to support the preference of Hypothesis (2) for concrete elements when it comes to indication of part manipulation in general.

HYPOTHESIS (3)

Hypothesis (3) expanded to include the PBI case and posited that, for the particular interfaces utilized in this user study, total assembly times would be fastest for the CAR case, followed by PBI, and slowest with AAR. This hypothesis, being concerned with completion time by definition, was evaluated using the total assembly time excluding the results for participants who did not successfully complete the assembly. These times were previously shown in Figure 22, and are repeated in a slightly different format in Figure 26. The rationale for excluding times in which participants failed to complete the assembly successfully is based on the supposition

that assemblers who fail to complete an assembly in a real-world setting will typically seek assistance from a manager or coworker. Such assistance would add a new dynamic to the assembly process which neither this thesis nor this user study account for.

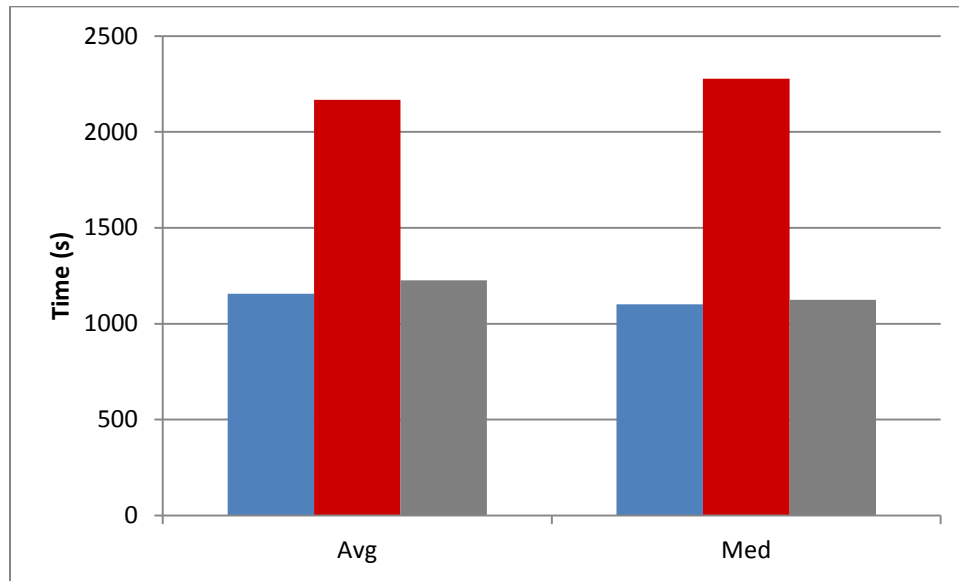


Figure 26. Alternate presentation of completion times for correct assemblies only.

Because the average completion time for the AAR case was noticeably higher than the CAR and PBI cases, while the latter two were relatively similar, statistical analyses were carried out separately for each of the three possible pairwise comparisons of each test case. Beginning with the two most salient cases—AAR versus CAR and PBI—the representation of completion times for those three cases are, in that order, T_{AAR} , T_{CAR} , and T_{PBI} . Thus the first two null and alternative hypotheses are as follows:

$$\begin{array}{ll}
 H_0: & T_{AAR} = T_{CAR} \qquad T_{AAR} = T_{PBI} \\
 H_a: & T_{AAR} > T_{CAR} \qquad T_{AAR} > T_{PBI}
 \end{array}$$

Two F-tests found $F(8,13) = 0.927$, $p > 0.5$ for AAR versus CAR and $F(8,9) = 1.431$, $p > 0.3$ for AAR versus PBI, indicating homoscedasticity for both. Consequently, one-tailed t-tests for two-sample, homoscedastic data were conducted for both comparisons. The results of these tests were $t(21) = 3.221$, $p < 0.01$ for AAR versus CAR and $t(17) = 3.594$, $p < 0.01$ for AAR versus PBI—both sufficient to reject the null hypotheses at a 99% confidence level. Thus it can be concluded with 99% confidence that the true average completion times for both the CAR and the PBI cases is lower than that of the AAR case.

Statistical analysis between the CAR and PBI completion times was carried out to investigate the significance of the slightly lower average time that the CAR case held versus PBI. Representing the completion times as before, the null and alternative hypothesis are:

$$H_0: T_{CAR} = T_{PBI}$$

$$H_a: T_{CAR} < T_{PBI}$$

An F-test found $F(9,13) = 1.121$, $p > 0.4$ for this comparison, indicating homoscedasticity.

Therefore, a one-tailed t-test for two-sample, homoscedastic data was conducted. The results of this test were $t(22) = 0.323$, $p > 0.3$, and thus insufficient to reject the null hypothesis.

Although the components of Hypothesis (3) involving the AAR completion time were supported, the lack of support for the CAR versus PBI comparison is perceived to be the result of a difference in feedback. As mentioned previously, assemblers in the PBI case had access to pictures of each completed assembly step. Conversely, assemblers in the CAR case received no such information and therefore dedicated additional time towards attempting to confirm that their actions were correct.

HYPOTHESIS (4)

Hypothesis (4) addressed training and posited that AR, either concrete or abstract, is more suitable than paper-based instructions for assembly and maintenance training. For the purposes of this study, this hypothesis was evaluated via feedback received from participants on the post-questionnaire. The three relevant questions asked participants to rank different confidence levels on a five point scale and read as follows:

- a. How confident did you feel about your ability to perform the assembly task when the study began?
 Not Confident 1 2 3 4 5 Very Confident
- b. How confident would you feel about your ability to perform this task again now that the study is complete?
 Not Confident 1 2 3 4 5 Very Confident
- c. How confident do you feel about your ability to perform OTHER tasks similar to this now that the study is complete?
 Not Confident 1 2 3 4 5 Very Confident

The first question was used as a baseline and subtracted from the participant's responses to the second and third questions in order to establish a measure of confidence increase.

Because trainees are typically trained to work on the exact assemblies for which they will later be responsible, increases in confidence pertaining to performing the same task again (as addressed in the second question) were considered first. These are shown in Figure 27. In accordance with Hypothesis (4), comparisons were made between the PBI case and each of the AR cases. Representing confidence ranking increases for the CAR, AAR, and PBI cases as CI_{CAR} , CI_{AAR} , and CI_{PBI} , respectively, the null and alternative hypotheses are:

$$H_0: CI_{CAR} = CI_{PBI} \quad CI_{AAR} = CI_{PBI}$$

$$H_a: CI_{CAR} > CI_{PBI} \quad CI_{AAR} > CI_{PBI}$$

Two F-tests found $F(9,13) = 0.986$, $p > 0.4$ for CAR versus PBI and $F(8,9) = 0.206$, $p > 0.9$ for AAR versus PBI, indicating homoscedasticity for both. Consequently, one-tailed t-tests for two-sample, homoscedastic data were conducted for both comparisons. The results of the CAR versus PBI test were $t(22) = 2.254$, $p < 0.02$, thereby sufficient to reject the null hypothesis at a 95% confidence interval. Thus it can be concluded with 95% confidence that the true average increase in confidence about performing the same task again is higher for the CAR case than for the PBI case. However, the results for the AAR versus PBI case were $t(17) = 0.709$, $p > 0.2$, and thus insufficient to reject the corresponding null hypothesis.

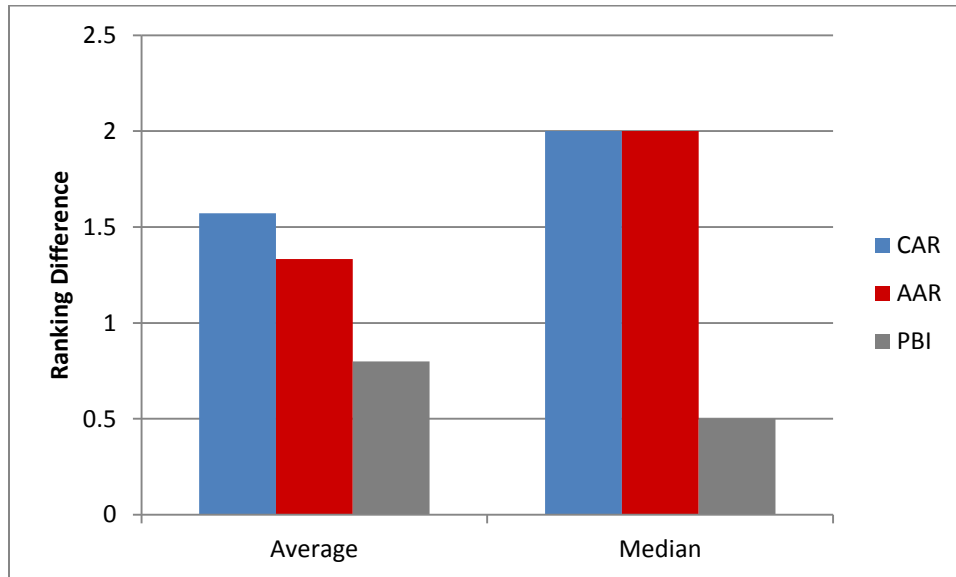


Figure 27. Confidence increases for performing the same task.

The lack of support for the AAR versus PBI component of Hypothesis (4) can be attributed to many of the same factors discussed in Hypotheses (2) and (3) which led to high

AAR error rates and long AAR completion times. Requiring extended periods of time to understand part manipulation instructions will already begin to degrade a participant's confidence; and such degradation will be compounded when numerous orientation errors are made despite his or her efforts. Furthermore, statistical analysis for the AAR case was strongly influence by a much larger degree of variability in responses by AAR participants than CAR participants. The sample standard deviation in confidence increases for the CAR case was 0.756, while the same for AAR case was 2.179. Such disparity in variability can be largely linked to confidence *decreases* reported by some AAR participants—one as large as a three point drop—whereas no CAR participants reported a confidence decrease at all.

In an effort to investigate the validity of Hypothesis (4) even further, the comparison between the AR cases and the PBI case was extended to include confidence rating increases pertaining both to performing the same task again as well as to performing other, similar tasks as addressed in the third confidence ranking. The combined responses are shown in Figure 28.

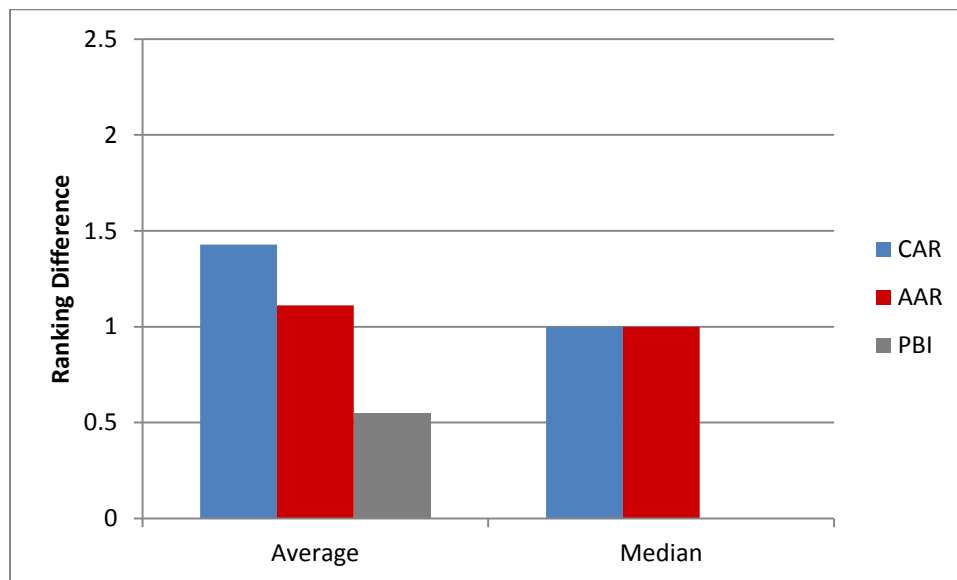


Figure 28. Confidence increases for performing the same task *and* other, similar tasks.

Statistical analysis for this data set made use of identical null and alternative hypotheses as the previous investigation of Hypothesis (4). On that basis two F-tests found $F(19,27) = 0.770$, $p > 0.7$ for CAR versus PBI and $F(17,19) = 0.401$, $p > 0.9$ for AAR versus PBI, indicating homoscedasticity for both. Consequently, one-tailed t-tests for two-sample, homoscedastic data were conducted for both comparisons as before. The results of the CAR versus PBI test were $t(46) = 3.075$, $p < 0.01$, and thereby once again sufficient to reject the null hypothesis, this time at a 99% confidence interval. Thus it can be concluded with 99% confidence that the true average increase in confidence about performing the same task again *and* performing other similar tasks is higher for the CAR case than for the PBI case. However, results for the AAR versus PBI test were $t(36) = 1.153$, $p > 0.1$, and thus still insufficient to reject the corresponding null hypothesis.

The secondary analysis of confidence increase failed to validate the AAR case over the PBI case for the same reasons as the first analysis. However, the CAR case was found to be favorable over the PBI case with an even greater degree of confidence, lending further support to the CAR versus PBI component of Hypothesis (4).

Finally, additional comments given by several participants also add support to Hypothesis (4) in terms of intuitiveness for new users. For instance, one participant from the CAR case commented, “[The assembly] was pretty self-explanatory if you have the AR.” Conversely, multiple participants in the PBI case commented that the overall task was not intuitive. One such participant bluntly called the motor itself “really confusing.” The fact that such incongruent comments were made about the same object evidences the impact made by the information presentation method.

CHAPTER 6 – CONCLUSIONS AND FUTURE WORK

RESEARCH CONCLUSIONS

This thesis has taken two approaches to the evaluation of information presentation GUI elements for AR assembly tasks. The first approach involved the development of a framework within which assembly tasks are broken down into their component subtasks and interface elements are organized into a hierarchy. This framework is useful for exploring which interface elements are the most suitable for each of the various assembly subtasks, and to this end a number of AR systems from literature were assessed. This assessment revealed that certain interface element usage patterns were occasionally preferred by some researchers, but that any semblance of a standard practice was lacking.

In order to begin making strides towards an improved understanding of best practices for AR interfaces, a user study was conducted in which participants were instructed how to assemble a hydraulic axial piston motor using both paper-based instructions and two differing sets of AR-based instructions. Specific investigations of the study results supported the hypotheses that concrete AR elements are more suitable to convey part manipulation information than abstract AR elements, that concrete AR instructions lead to faster assembly times than abstract AR instructions alone, and that concrete AR instructions lead to greater increases in user confidence than paper-based instructions, but failed to support the hypothesis that abstract AR elements are more suitable for part identification than concrete AR elements. Note that each of these conclusions was reached in light of the particular interfaces utilized for this user study, and that the usage of different variations or combinations of interface elements may yield different results.

Stepping back from the individual hypotheses, this study also illustrated very effectively that the benefits promised by AR for assembly applications will not be realized automatically. A well-designed interface is crucial for AR to reach its potential; and without one, paper-based instructions can easily outperform AR systems. To ensure the former rather than the latter, the designers of AR assembly interfaces would be wise to leverage the strengths of both concrete and abstract interface elements as appropriate for each and every assembly step and subtask.

FUTURE WORK

In addition to further research regarding different variations of the hypotheses examined by this research, future work in the area of information presentation interface elements for AR assembly should focus on two goals—gaining a more detailed understanding of the strengths and weaknesses of concrete and abstract elements for various assembly subtasks, and developing principles for successfully combining the two types of elements when appropriate. In regards to the former this thesis made a case for preferring concrete elements for most indication of part manipulation subtasks, but future work should seek to do the same to a greater degree for the part identification subtask. This could be approached in two ways in order to avoid the obstacle outlined in Chapter 5 (that is, interference caused by poor communication regarding part manipulation). One approach would be to design two interfaces which have different interface elements (concrete or abstract) for part identification, but identical elements for indication of part manipulation. Another approach would be to shift focus away from the number of errors and instead record the time that it takes each participant to identify the correct part, yielding even more finely-grained timing data than was collected for the second portion of this study.

The second goal—developing principles for the combination of concrete and abstract elements—could be effectively pursued by classifying the types of part manipulation subtasks that are not well-served by concrete interface elements alone. Two examples of such subtasks include differentiating between part orientations that appear visually very similar, or simultaneously demonstrating both large, course-grained motions and the intricate meshing of small part features. Generating insight into how subtasks such as these can be described to the user in a way that is detailed enough to be complete but concise enough to avoid clutter or general confusion will go a long way towards improving the efficiency of future AR assembly interfaces.

BIBLIOGRAPHY

- [1] A. Dünser, R. Grasset, H. Seichter, and M. Billinghurst, "Applying HCI principles to AR systems design," in *Proceedings of the 2nd International Workshop at the IEEE Virtual Reality 2007 Conference*, Charlotte, NC, 2007.
- [2] R. T. Azuma, "A survey of augmented reality," *Presence-Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997.
- [3] G. Welch and E. Foxlin, "Motion tracking: No silver bullet, but a respectable arsenal," *Computer Graphics and Applications, IEEE*, vol. 22, no. 6, pp. 24–38, 2002.
- [4] C. Ke, B. Kang, D. Chen, and X. Li, "An augmented reality-based application for equipment maintenance," in *Affective Computing and Intelligent Interaction*, J. Tao, T. Tan, and R. W. Picard, Eds. Springer Berlin Heidelberg, 2005, pp. 836–841.
- [5] S. Feiner, B. Macintyre, and D. Seligmann, "Knowledge-based augmented reality," *Communications of the ACM*, vol. 36, no. 7, pp. 53–62, 1993.
- [6] A. Tang, C. Owen, F. Biocca, and W. Mou, "Comparative effectiveness of augmented reality in object assembly," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2003, pp. 73–80.
- [7] U. Neumann and A. Majoros, "Cognitive, performance, and systems issues for augmented reality applications in manufacturing and maintenance," in *Proceedings of the IEEE Virtual Reality Annual International Symposium*, 1998, pp. 4–11.
- [8] F. Biocca, A. Tang, D. Lamas, J. Gregg, R. Brady, and P. Gai, "How do users organize virtual tools around their body in immersive virtual and augmented environment?: An exploratory study of egocentric spatial mapping of virtual tools in the mobile infosphere," Media Interface and Network Design Labs, Michigan State University, East Lansing, MI, 2001.
- [9] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *Computer Graphics and Applications, IEEE*, vol. 21, no. 6, pp. 34–47, 2001.
- [10] D. Reiners, D. Stricker, G. Klinker, and S. Müller, "Augmented reality for construction tasks: doorlock assembly," *Proc. IEEE and ACM IWAR*, vol. 98, no. 1, pp. 31–46, 1998.
- [11] S. You, U. Neumann, and R. Azuma, "Hybrid inertial and vision tracking for augmented reality registration," in *Proceedings of the IEEE Virtual Reality Symposium*, 1999, pp. 260–267.
- [12] S. K. Ong, M. L. Yuan, and A. Y. C. Nee, "Augmented reality applications in manufacturing: a survey," *International Journal of Production Research*, vol. 46, no. 10, pp. 2707–2742, May 2008.

- [13] “Assemble,” *Webster’s New World Dictionary of the American Language*. New World Dictionaries, New York, p. 82, 1982.
- [14] K. Ikeuchi and T. Suehiro, “Towards an Assembly Plan from Observation,” in *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, 1992, pp. 2171–2177.
- [15] L. S. Homem de Mello and A. C. Sanderson, “Representations of mechanical assembly sequences,” *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 211–227, 1991.
- [16] L. S. Homem de Mello and A. C. Sanderson, “AND/OR graph representation of assembly plans,” *IEEE Transactions on Robotics and Automation*, vol. 6, no. 2, pp. 188–199, 1990.
- [17] L. S. Homem de Mello and A. C. Sanderson, “A correct and complete algorithm for the generation of mechanical assembly sequences,” *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 228–240, 1991.
- [18] S. J. Henderson and S. Feiner, “Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret,” in *Proceedings of the 8th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2009, pp. 135–144.
- [19] K. M. Baird and W. Barfield, “Evaluating the effectiveness of augmented reality displays for a manual assembly task,” *Virtual Reality*, vol. 4, no. 4, pp. 250–259, 1999.
- [20] A. C. Boud, D. J. Haniff, C. Baber, and S. J. Steiner, “Virtual reality and augmented reality as a training tool for assembly tasks,” in *Proceedings of the IEEE International Conference on Information Visualization*, 1999, pp. 32–36.
- [21] S. Wiedenmaier, O. Oehme, L. Schmidt, and H. Luczak, “Augmented Reality (AR) for Assembly Processes Design and Experimental Evaluation,” *International Journal of Human-Computer Interaction*, vol. 16, no. 3, pp. 497–514, 2003.
- [22] S. G. Hart and L. E. Staveland, “Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research,” *Human mental workload*, vol. 1, pp. 139–183, 1988.
- [23] B. Schwerdtfeger and G. Klinker, “Supporting order picking with Augmented Reality,” in *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 2008, pp. 91–94.
- [24] J. Zauner, M. Haller, A. Brandl, and W. Hartman, “Authoring of a mixed reality assembly instructor for hierarchical structures,” in *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2003, pp. 237–246.
- [25] “graphical user interface (GUI),” *BusinessDictionary.com*. WebFinance, Inc., 2013.

- [26] D. Kalkofen, E. Mendez, and D. Schmalstieg, "Interactive focus and context visualization for augmented reality," in *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 1–10.
- [27] J. Gabbard, "Researching Usability Design and Evaluation Guidelines for Augmented Reality (AR) Systems." Virginia Polytechnic Institute and State University, 2001.
- [28] C. D. Wickens and P. Baker, "Chapter 13, Cognitive Issues in Virtual Reality," in *Virtual Environments and Advanced Interface Design*, W. Barfield and T. A. Furness, Eds. Oxford University Press, 1995, pp. 516–541.
- [29] M. M. Wloka and B. G. Anderson, "Resolving occlusion in augmented reality," in *Proceedings of the 1995 symposium on Interactive 3D graphics*, 1995, pp. 5–12.
- [30] S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon, "Windows on the world: 2D windows for 3D augmented reality," in *Proceedings of the 6th annual ACM symposium on User interface software and technology*, 1993, pp. 145–155.
- [31] D. Hix and H. R. Hartson, *Developing User Interfaces: Ensuring Usability Through Product and Process*. New York: John Wiley and Sons, 1993.
- [32] A. A. Rizzo, G. J. Kim, S.-C. Yeh, M. Thiebaut, J. Hwang, and J. G. Buckwalter, "Development of a benchmarking scenario for testing 3D user interface devices and interaction methods," in *Proceedings of the 11th International Conference on Human Computer Interaction, Las Vegas, Nevada, USA*, 2005.
- [33] S. Li, T. Peng, C. Xu, Y. Fu, and Y. Liu, "A Mixed Reality-Based Assembly Verification and Training Platform," *Virtual and Mixed Reality*, pp. 576–585, 2009.
- [34] V. Raghavan, J. Molineros, and R. Sharma, "Interactive evaluation of assembly sequences using augmented reality," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 435–449, 1999.
- [35] J. Sääski, T. Salonen, M. Hakkarainen, S. Siltanen, C. Woodward, and J. Lempiäinen, "Integration of design and assembly using augmented reality," in *Micro-Assembly Technologies and Applications*, vol. 260, S. Ratchev and S. Koelemeijer, Eds. Boston: Springer, 2008, pp. 395–404.
- [36] B. Schwald and B. De Laval, "An augmented reality system for training and assistance to maintenance in the industrial context," *Journal of WSCG*, vol. 11, no. 1, 2003.
- [37] S. K. Ong, Y. Pang, and A. Y. C. Nee, "Augmented Reality Aided Assembly Design and Planning," *CIRP Annals - Manufacturing Technology*, vol. 56, no. 1, pp. 49–52, Jan. 2007.
- [38] A. Liverani, G. Amati, and G. Caligiana, "A CAD-augmented Reality Integrated Environment for Assembly Sequence Check and Interactive Validation," *Concurrent Engineering*, vol. 12, no. 1, pp. 67–77, Mar. 2004.

- [39] Y. Shen, S. K. Ong, and A. Y. C. Nee, "A framework for multiple-view product representation using Augmented Reality," in *Proceedings of the International Conference on Cyberworlds (CW'06)*, 2006, pp. 157–164.
- [40] S. Weibel, U. Bockholt, and J. Keil, "Design criteria for AR-based training of maintenance and assembly tasks," in *Proceedings of the International Conference on Virtual and Mixed Reality*, Orlando, FL, 2011, vol. 6773, pp. 123–132.
- [41] T. P. Caudell and D. W. Mizell, "Augmented reality: An application of heads-up display technology to manual manufacturing processes," in *Proceedings of the 25th Hawaii International Conference on System Sciences*, 1992, vol. 2, pp. 659–669.
- [42] W. Friedrich, D. Jahn, and L. Schmidt, "ARVIKA-augmented reality for development, production and service," in *Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 2002, pp. 3–4.
- [43] F. Biocca, A. Tang, C. Owen, and F. Xiao, "Attention funnel: omnidirectional 3D cursor for mobile augmented reality platforms," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2006, pp. 1115–1122.
- [44] A. Gharsellaoui, J. Oliver, and S. Garbaya, "Benchtop Augmented Reality Interface for Enhanced Manual Assembly," in *Proceedings of the IEEE Aerospace Conference*, Toulouse, France, 2011.
- [45] M. L. Yuan, S. K. Ong, and A. Y. Nee, "Assembly guidance in augmented reality environments using a virtual interactive tool," Singapore-MIT Alliance (SMA), Innovation in Manufacturing Systems and Technology (IMST), 2005.
- [46] G. Reinhart and C. Patron, "Integrating Augmented Reality in the assembly domain-fundamentals, benefits and applications," *CIRP Annals-Manufacturing Technology*, vol. 52, no. 1, pp. 5–8, 2003.
- [47] A. Webster, S. Feiner, B. MacIntyre, W. Massie, and T. Krueger, "Augmented Reality in Architectural Construction, Inspection, and Renovation," in *Proceedings of the ASCE Third Congress on Computing in Civil Engineering*, 1996, pp. 913–919.
- [48] A. Velizhev, *GML C++ Camera Calibration Toolbox*. Graphics and Media Lab: Lomonosov Moscow State University.

APPENDIX A – Gabbard’s Design Principles [27]

Guidelines: VE Users

Take into account user experience (i.e., support both expert and novice users), user’s physical abilities (e.g., handedness), and users' technical aptitudes (e.g. orientation, spatial visualization, and spatial memory).

Support users with varying degrees of domain knowledge.

When designing collaborative environments, support social interaction among users (e.g., group communication, role-play, informal interaction) and cooperative task performance (e.g., facilitate social organization, construction, and execution of plans)

Take into account the number and locations of potential users.

For test bed AR environments (i.e., those used for research purposes), calibration methods should be subject-specific. That is, the calibration should account for individual differences.

For test bed AR environments (i.e., those used for research purposes), calibration methods should provide no additional or residual cues that may be exploited by subjects.

Provide (access to) information about other collaborative users even when they are physically occluded or "remotely" located.

In social/collaborative environments, support face-to-face communication by presenting visual information within a user’s field-of-view that would otherwise require the user to look away. Avoid interaction techniques (and devices) that require a noticeable portion of a user’s attention.

Guidelines: VE User Tasks

Design interaction mechanisms and methods to support user performance of serial tasks and task sequences.

Support concurrent task execution and user multiprocessing.

Provides stepwise, subtask refinement including the ability to undo and go "back" when navigating information spaces.

Guidelines: Navigation and Locomotion

Support appropriate types of user navigation (e.g., naive search, primed search, exploration), facilitate user acquisition of survey knowledge (e.g., maintain a consistent spatial layout)

When augmenting landscape and terrain layout, consider [Darken and Sibert, 1995] organizational principles.

When appropriate, include spatial labels, landmarks, and a compass

Provide information so that users can always answer the questions: Where am I now? What is my current attitude and orientation? Where do I want to go? How do I travel there?

Guidelines: Object Selection

Strive for body-centered interaction. Support multimodal interaction.

Use non-direct manipulation means (such as query-based selection) when selection criteria are temporal, descriptive, or relational.

Strive for high frame rates and low latency to assist users in three-dimensional target acquisition

Provide accurate depiction of location and orientation of graphics and text.

Guidelines: Object Manipulation

Support two-handed interaction (especially for manipulation-based tasks).

For two-handed manipulation tasks, assign dominant hand to fine-grained manipulation relative to the non-dominant hand

Guidelines: User Representation and Presentation

For AR-based social environments (e.g., games), allow users to create, present, and customize private and group-wide information.

For AR-based social environments (e.g., games), provide equal access to "public" information.

In collaborative environments, allow users to share tracking information about themselves (e.g., gesture based information) to others. Allow users to control presentation of both themselves and others (e.g., to facilitate graceful degradation).

Guidelines: VE Agent Representation and Presentation

Include agents that are relevant to user tasks and goals.

Organize multiple agents according to user tasks and goals

Allow agent behavior to dynamically adapt, depending upon context, user activity, etc.

Represent interactions among agents and users (rules of engagement) in a semantically consistent, easily visualizable manner.

Guidelines: Virtual Surrounding and Setting

Support significant occlusion-based visual cues to the user, by maintaining proper occlusion between real and virtual objects.

When possible, determine occlusion, dynamically, in real-time (i.e., at every graphics frame).

When presenting inherently 2D information, consider employing 2D text and graphics of the sort supported by current window systems

In collaborative environments, support customized views (including individual markers, icons and annotations) that can be either shared or kept private

To avoid display clutter in collaborative environments, allow users to control the type and extent of visual information (per participant) presented.

Optimize stereoscopic visual perception by ensuring that left and right eye images contain minimum vertical disparities. Minimize lag between creation of left and right eye frames.

Guidelines: VE System and Application Information

Use progressive disclosure for information-rich interfaces. Pay close attention to the visual, aural, and haptic organization of presentation (e.g., eliminate unnecessary information, minimize overall and local density, group related information, and emphasize information related to user tasks).

Strive to maintain interface consistency across applications

Language and labeling for commands should clearly and concisely reflect meaning.
System messages should be worded in a clear, constructive manner so as to encourage user engagement (as opposed to user alienation)

For large environments, include a navigational grid and/or a navigational map.

When implementing maps, consider to [Darken and Sibert, 1995] map design principles

Present domain-specific data in a clear, unobtrusive manner such that the information is tightly coupled to the environment and vice-versa.

Strive for unique, powerful presentation of application-specific data, providing insight not possible through other presentation means

Guidelines: Tracking User Location and Orientation

Assess the extent to which degrees of freedom are integrable [sic] and separable within the context of representative user tasks.

Eliminate extraneous degrees of freedom by implementing only those dimensions which users perceive as being related to given tasks.

Multiple (integral) degrees of freedom input is well-suited for coarse positioning tasks, but not for tasks which require precision.

When assessing appropriate tracking technology relative to user tasks, one should consider working volume, desired range of motion, accuracy and precision required, and likelihood of tracker occlusion.

Calibration requirements for AR tracking systems should include:

1. calibration methods which are statistically robust,
2. a variety of calibration approaches for different circumstances, and,
3. metrology equipment that is sufficiently accurate, convenient to use.

For test bed AR environments (i.e., those used for research purposes), calibration methods should be independent. That is, separate parts of the entire calibration should not rely on each.

Relative latency is a source of misregistration and should be reduced.

Devices should be both spatially and temporally registered (supports effective integration

of user interaction devices which may vary in type, accuracy, bandwidth, dynamics and frequency).

Match the number of degrees of freedom (in the physical device and interaction techniques) to the inherent nature of the task. For example, menu selection is a 2D task and as such should not require a device or interaction technique with more than two degrees of freedom.

Consider using a Kalman Filter in head tracking data to smooth the motion and decrease lag.

Trackers should be accurate to small fraction of a degree in orientation and a few millimeters in position.

In head-tracked based AR systems, errors in measured head orientation usually cause larger registration offsets (errors) than object orientation errors do.

Minimize the combined latency of the tracker and the graphics engine.

Tracking systems (singleton tracking system or hybrids) should work at long ranges (i.e., support mobile users).

Minimize dynamic errors (maximize dynamic registration) by 1) reducing system lag, 2) reducing apparent lag, 3) matching temporal streams (with video-based systems), and 4) predicting future locations.

Guidelines: Data Gloves and Gesture Recognition

Allow gestures to be defined by users incrementally, with the option to change or edit gestures on the fly.

Avoid gesture in abstract 3D spaces; instead use relative gesturing.

Guidelines: Speech Recognition and Natural Language

Strive for seamless integration of annotation, provide quick, efficient, and unobtrusive means to record and playback annotations.

Allow users to edit, remove, and extract or save annotations

Guidelines: Visual Feedback -- Graphical Presentation

Timing and responsiveness of an AR system are crucial elements (e.g., effect user performance).

Strive for consistency among the various visual (and other sensory) cues which are used to infer information about the combined virtual and real world.

For stereoscopic applications, employ headsets that support adjustable interpupillary distances (IPD) between approximately 45mm to 75mm.

Allow that user to optimize the visual display, (e.g., support user-controlled (and preset) illuminance and contrast levels.

Ensure that wearable display is sufficiently comfortable and optically transparent for the user.

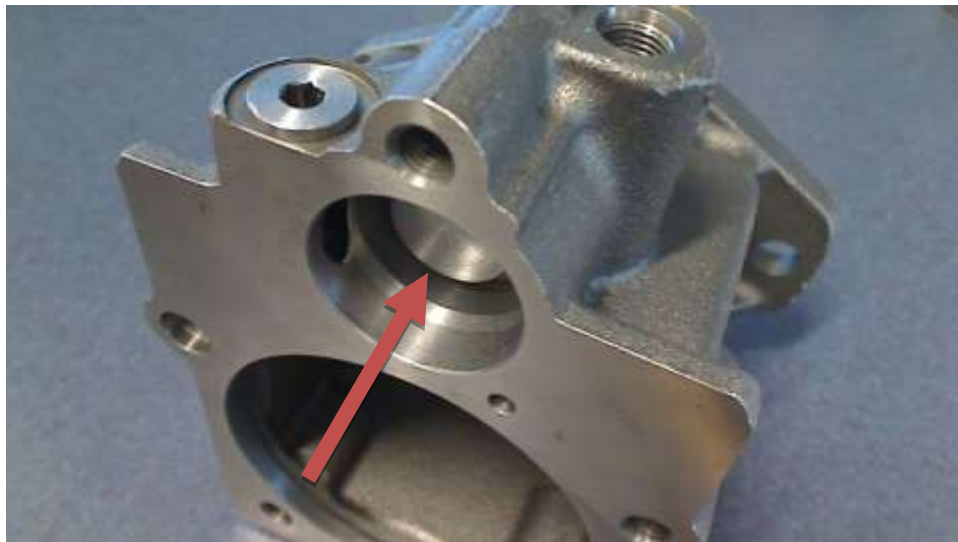
Minimize static errors by isolating and evaluating 1) optical distortion, 2) errors in the tracking system(s), 3) mechanical misalignments, and 4) incorrect viewing parameters (e.g., field of view, tracker-to-eye position and orientation, interpupillary distance)

APPENDIX B – PAPER INSTRUCTION MANUAL

SD-LVKV-4254050 Assembly Instructions



- 1) Insert the servo piston, seal-side second, into the upper hole of the housing. Orient the servo piston so that the perpendicular hole is visible through, and aligned with, hole L1 in the top of the housing.





- 2) Insert the swashplate into the main body of the housing so that the circular sides rest on the red journal bearings inside the housing. Ensure that the ball at the end of the swashplate arm extends upwards into the servo hole and rests within the vertical hole in the servo piston itself.





- 3) Insert the toothed shaft and bearing, bearing-side second, into the housing from the back side (with the white part number) and through the swashplate. Ensure that the bearing is seated all the way into the housing.





- 4) Use a set of snap-ring pliers to seat the retaining ring inside the back of the housing, within the groove adjacent to the shaft bearing.





- 5) Set the housing subassembly aside and insert the three (3) slipper hold-down pins into the slots in the top of the cylinder block.





- 6) Gently set the slipper retainer guide, indented side down, on top of the three (3) slipper hold-down pins.





- 7) Balance the slipper retainer, chamfered-side down, on top of the slipper retainer guide. Align the nine (9) holes in the slipper retainer with the nine (9) holes in the cylinder block.





- 8) Alternating sides, carefully insert all nine (9) piston and slipper units through the slipper retainer and into the cylinder block. The slipper heads should seat snugly against the slipper retainer.





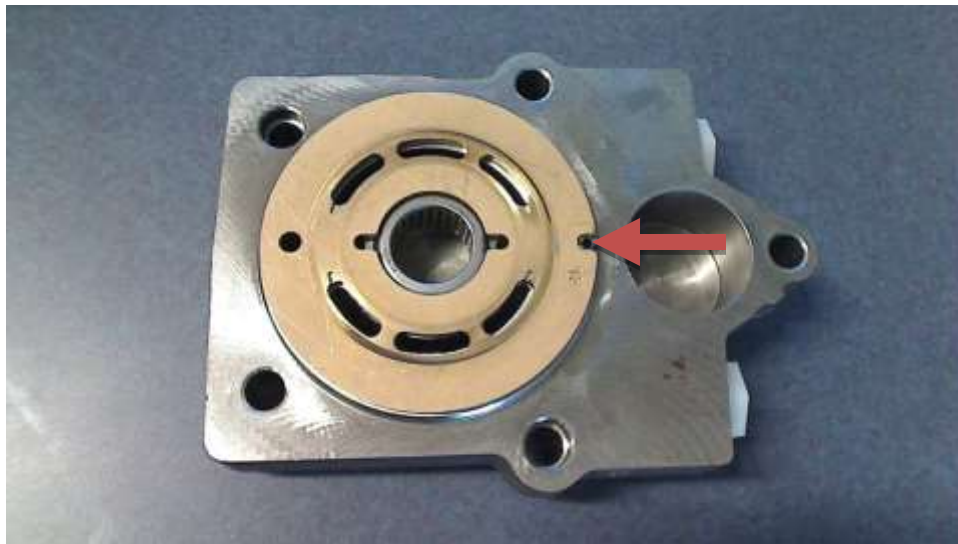
- 9) Ensure that the internal teeth of the cylinder block and the slipper retainer guide are aligned. Then carefully slide the entire cylinder block subassembly, slipper-side (top-side) first, over the toothed shaft and into the main body of the housing from the front side (the larger face). When the teeth of both the cylinder block and the slipper retainer guide are correctly meshed with the teeth of the shaft, the bottom of the cylinder block will be able to slide just past flush with the front face of the housing.



→ Go to next page

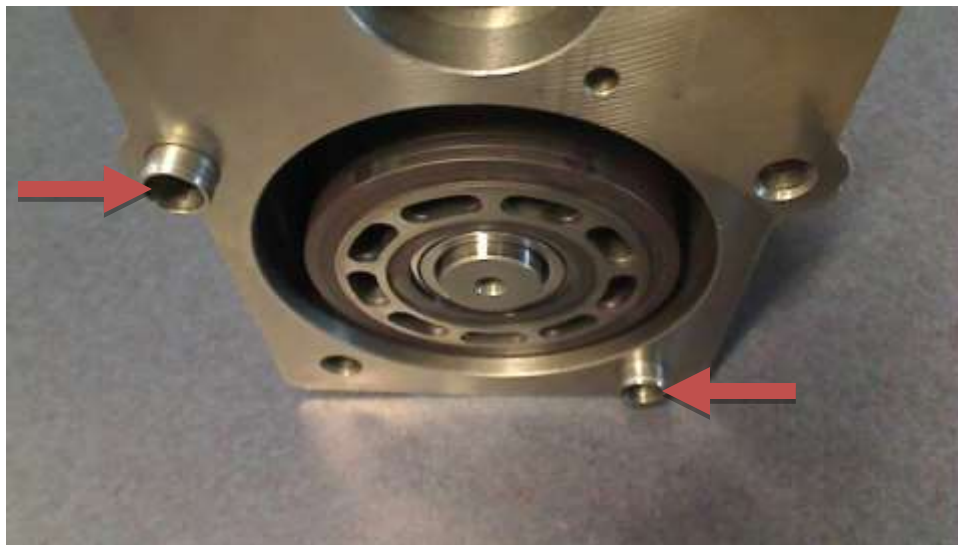


- 10) Set the housing subassembly aside and place the valve plate, brass-side up, onto the inner face of the end cap. Use firm pressure to seat the top edge of the valve plate onto the small timing pin.





- 11) Set the end cap aside and place the two (2) small hollow locators into the depressions on the front face of the main housing. The two (2) depressions are located around two (2) out of the total five (5) cap screw holes.



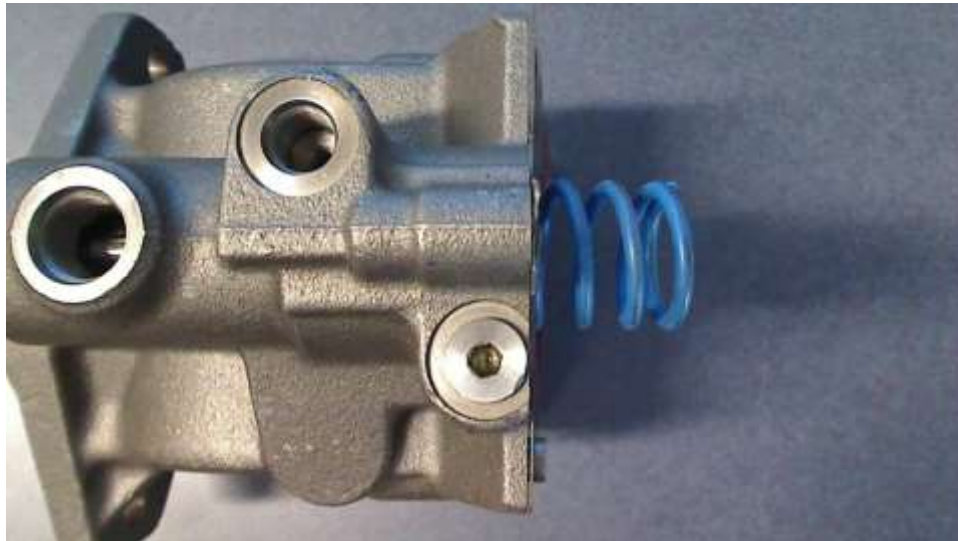


- 12) Align the gasket with the front face of the housing, seating it over the two (2) hollow locators and ensuring that the six (6) small holes in the gasket are aligned with the corresponding holes in the housing face.





- 13) Insert the servo spring into the upper servo hole of the housing so that it rests against the servo piston.





- 14) Insert the spring seat into the servo spring. If necessary, apply pressure so that the spring seat and servo spring force the servo piston all the way into the back of the upper servo hole of the housing.





- 15) Mate the entire housing subassembly with the end cap subassembly. Ensure alignment of the toothed shaft, spring seat, and hollow locators with the corresponding holes in the end cap. The pressure from the servo spring will prevent the two halves from remaining in contact until the next step is completed.





- 16) Use an 8mm Allen wrench and the five (5) cap screws to secure the end cap to the housing. Be sure to begin threading all five (5) screws before tightening any screws fully.



ASSEMBLY COMPLETE

APPENDIX C – CONCRETE AR SCREENSHOTS



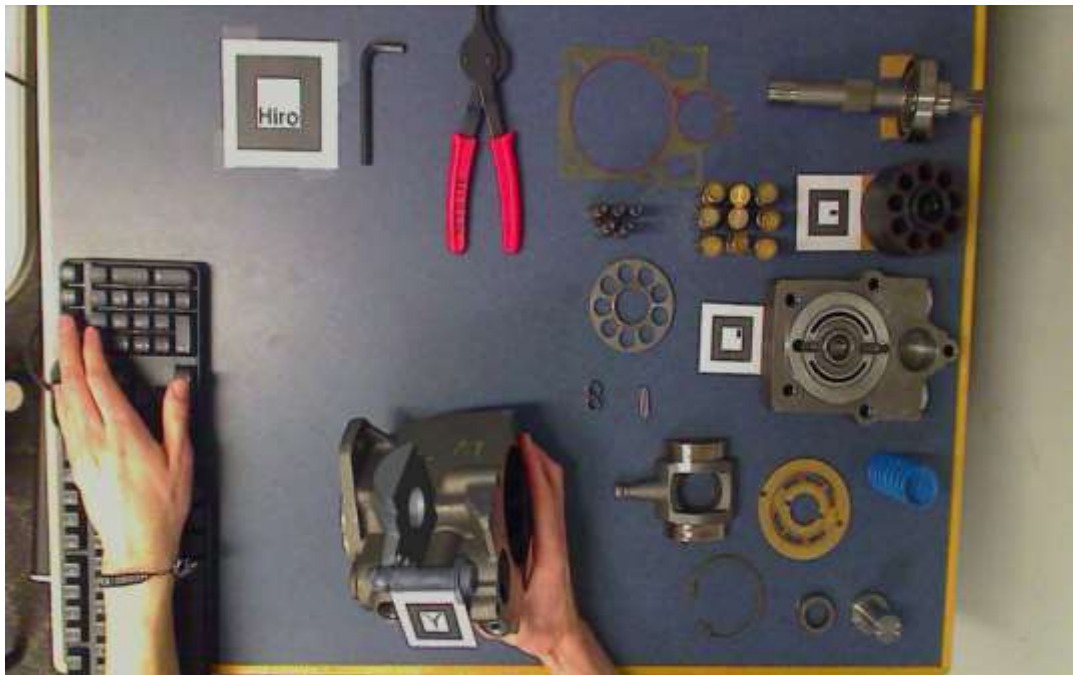
Welcome Screen



Step 1 – Piston model animates into housing with occlusion.



Step 2a – Beginning of swashplate animation.



Step 2b – End of swashplate animation



Step 3 – Shaft animates into housing with occlusion.



Step 4 – Snap ring and pliers animate up to housing with occlusion.



Step 5 – Pins animate into cylinder block.



Step 6 – Retainer guide animates onto pins and cylinder block.



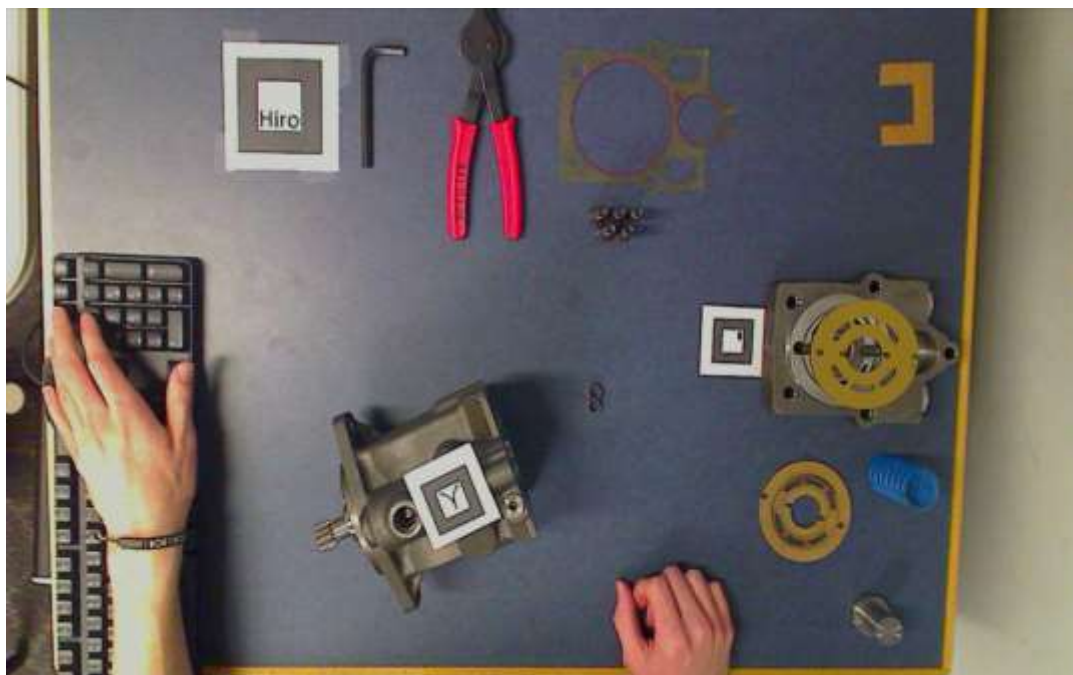
Step 7 – Retainer animates onto retainer guide and rotates to align holes.



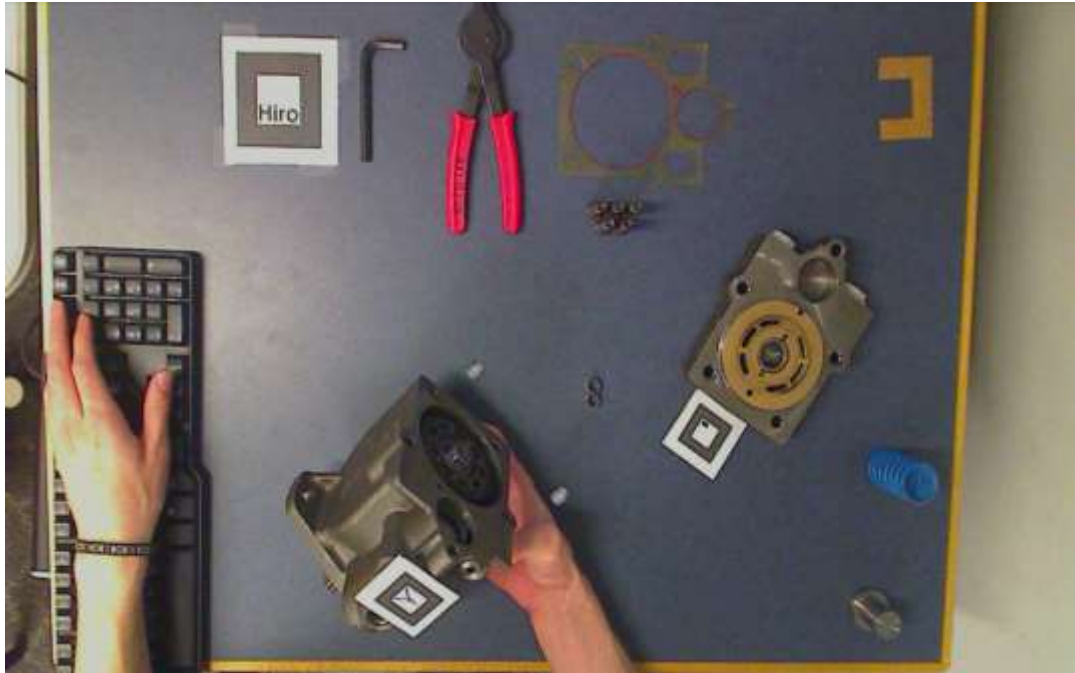
Step 8 – Pistons animate into cylinder block, opposing sides first.



Step 9 – Cylinder block animates onto shaft without occlusion. (Text instructions are not considered an actual assembly step.)



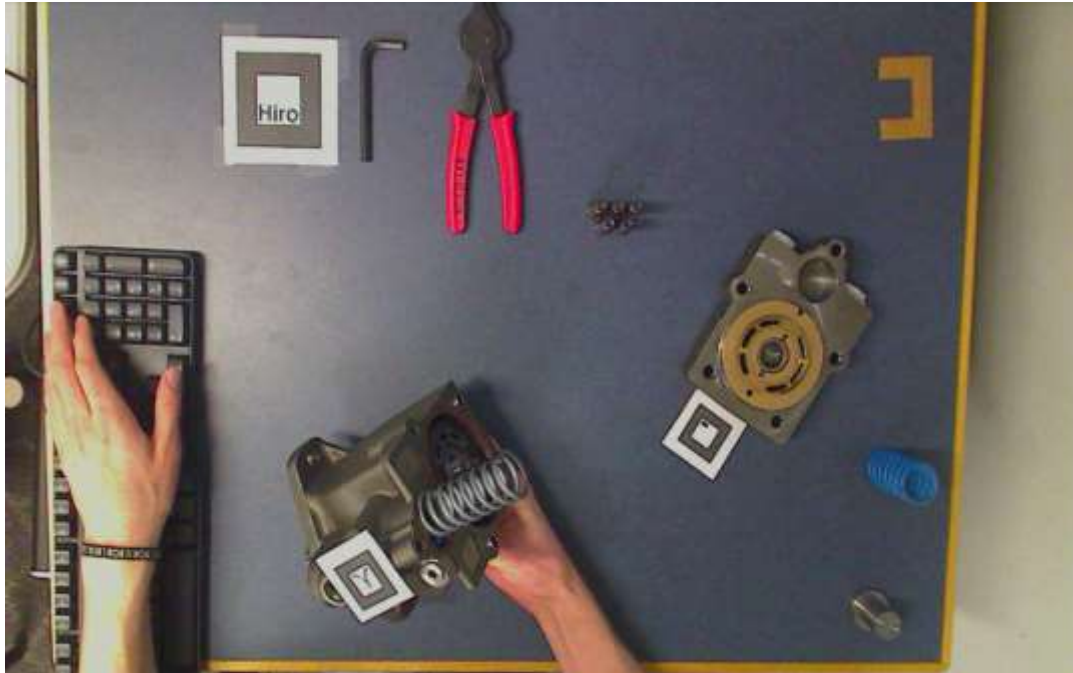
Step 10 – Valve plate animates onto end cap.



Step 11 – Hollow locators animate up to housing.



Step 12 – Gasket animates up to housing.



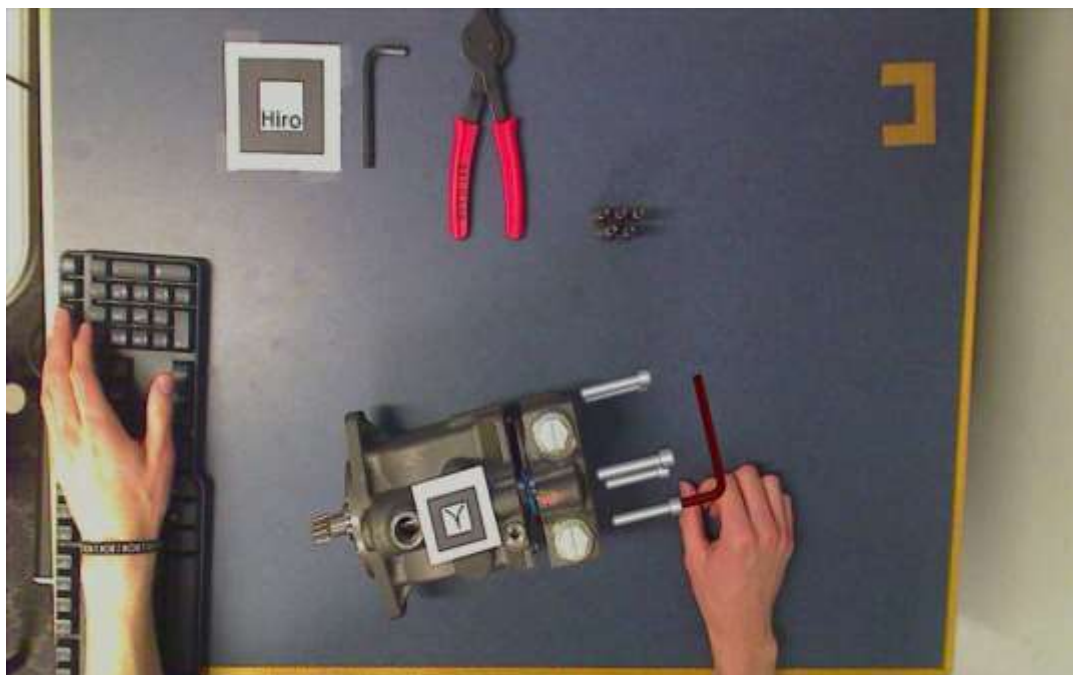
Step 13 – Spring animates into housing with occlusion.



Step 14 – Spring seat animates into spring with occlusion.



Step 15 – End cap animates up to housing. (Text instructions are not considered an actual assembly step.)



Step 16 – Screws animate into end cap and housing with occlusion. Hex key rotates.

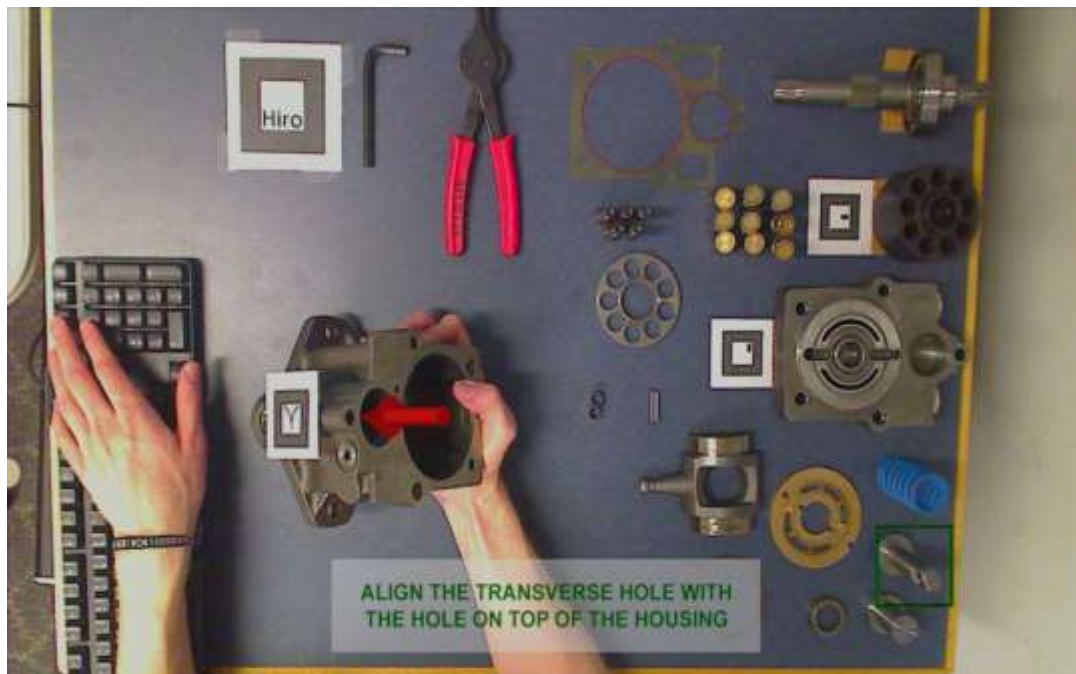


Dismissal Screen

APPENDIX D – ABSTRACT AR SCREENSHOTS



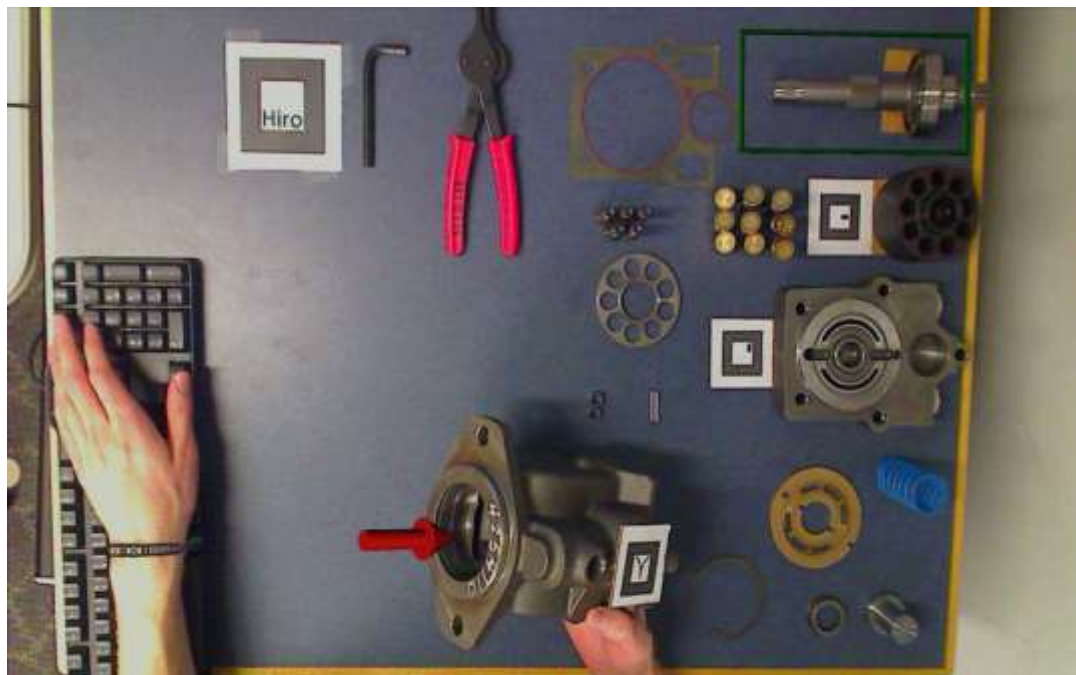
Welcome Screen



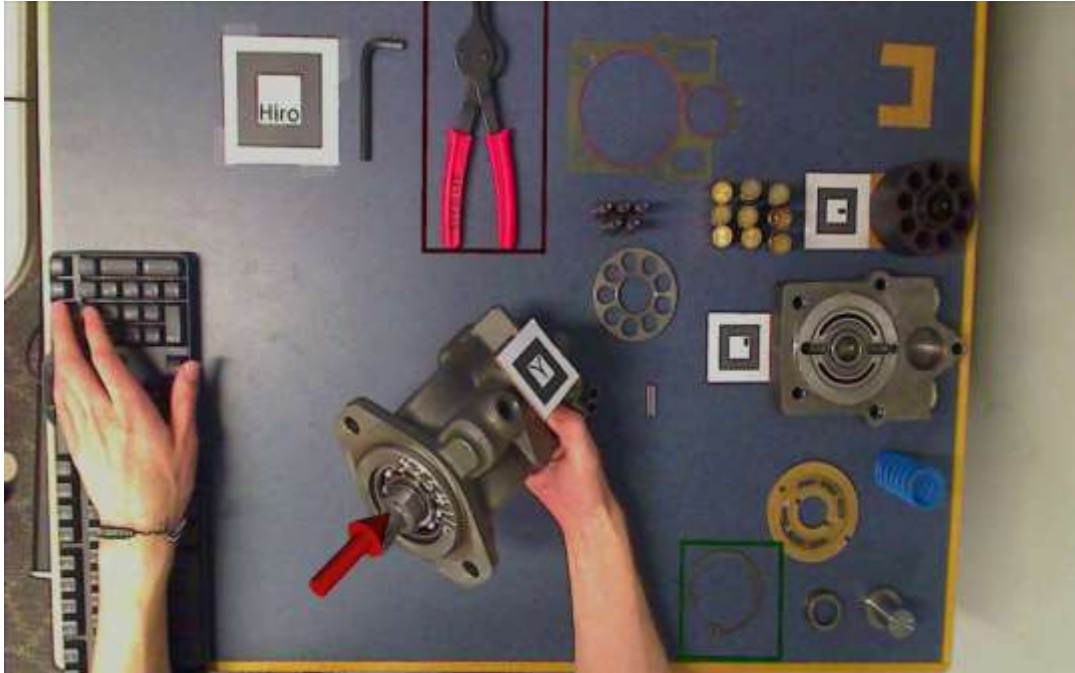
Step 1



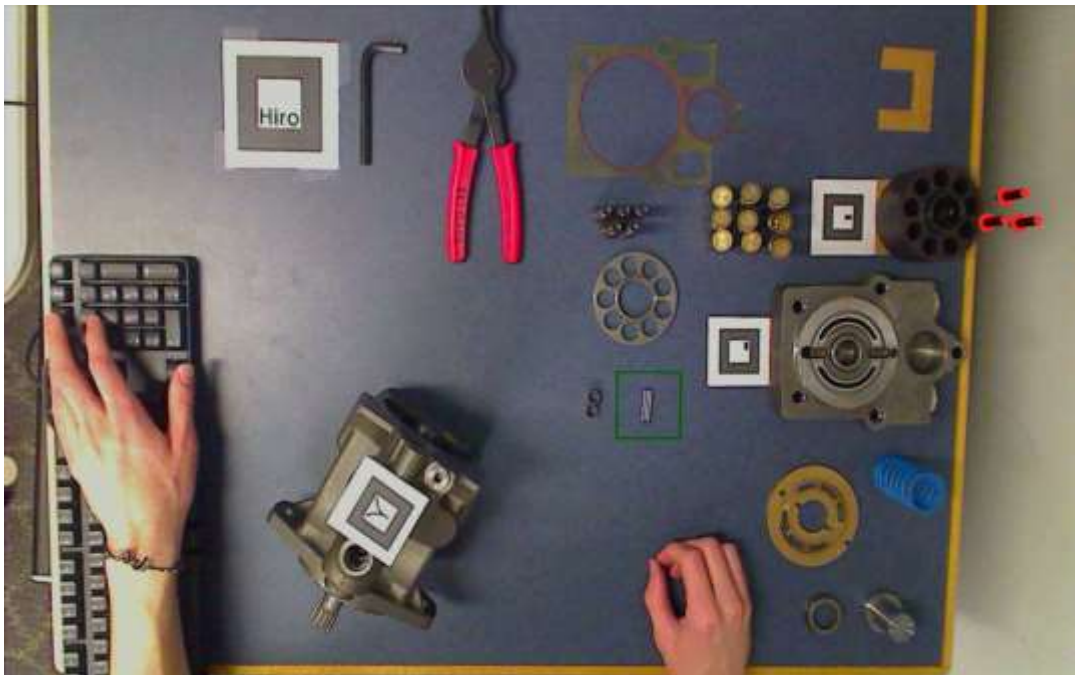
Step 2



Step 3



Step 4 – Red frame bounces.



Step 5



Step 6



Step 7



Step 10



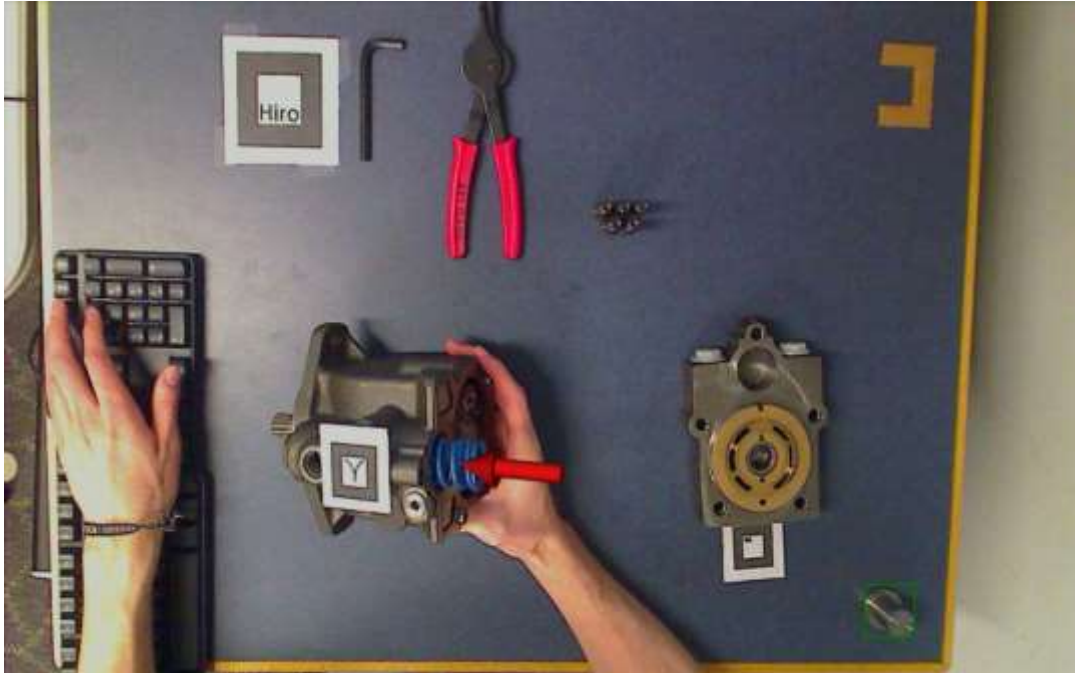
Step 11



Step 12



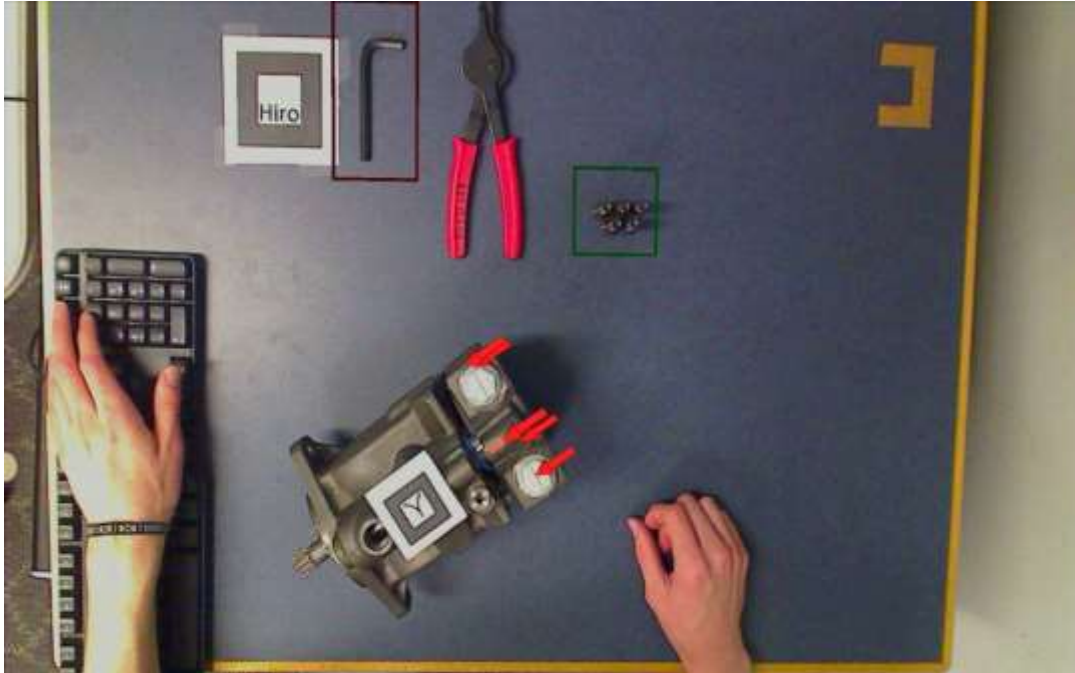
Step 13



Step 14



Step 15



Step 16 – Red frame bounces.



Dismissal Screen

APPENDIX E – XML SCRIPTS

Motor_AR_appC.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ns1:ARPrototypingToolkit xmlns:ns1="ARProToXML"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="https://vrac.iastate.edu/~rafael/xml/ARProTo.xsd">
  <Statemachine file=" ../models/computerdemo/pump_statemachine_C.py" />

  <Models>

    <!-- Start Screen -->

    <Content2D userID="Startup Text" label="Startup Text">
      <Position X="960.0" Y="120.0" Z="0.0"/>
      <File name=" ../models/computerdemo/Startup Text.png"/>
    </Content2D>

    <Content2D userID="Startup Text 2" label="Startup Text 2">
      <Position X="960.0" Y="960.0" Z="0.0"/>
      <File name=" ../models/computerdemo/Startup Text 2.png"/>
    </Content2D>

    <!-- STEP 1 -->

    <Model3D userID="S1C Piston" label="S1C Piston">
      <Position X="-30.0" Y="0" Z="-55.0"/>
      <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
      <File name=" ../models/computerdemo/servo piston (copy2).3ds"/>
      <Animation anim_name="S1C Piston Animation">
        <Keyframe time="0" X="0" Y="0" Z="130" />
        <Keyframe time="1.5" X="0" Y="0" Z="0" />
      </Animation>
    </Model3D>

    <!-- STEP 2 -->

    <Model3D userID="S2C Piston" label="S2C Piston">
      <Position X="-30.0" Y="0" Z="-40.0"/>
      <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
      <File name=" ../models/computerdemo/servo piston (copy2) -
semitrans.3ds"/>
```

</Model3D>

```
<Model3D userID="S2C Swashplate" label="S2C Swashplate">
  <Position X="30" Y="0" Z="-115.5"/>
  <Orientation RX="-1.57" RY="-1.57" RZ="0.0"/>
  <File name="../models/computerdemo/swashplate.3ds"/>
  <Animation anim_name="S2C Swashplate Animation">
    <Keyframe time="0" X="0" Y="-150" Z="0" type="t" />
    <Keyframe time="1" X="0" Y="-75" Z="0" type="t" />
    <Keyframe time="2" X="0" Y="0" Z="0" type="t" />
    <Keyframe time="2.5" X="0" Y="0" Z="0" type="t" />

    <Keyframe time="0" X="0" Y="0" Z="1.57" type="r" />
    <Keyframe time="1" X="0" Y="0" Z="1.57" type="r" />
    <Keyframe time="2" X="0" Y="0" Z="0" type="r" />
    <Keyframe time="2.5" X="0" Y="0" Z="0" type="r" />
  </Animation>
</Model3D>
```

<!-- STEP 3 -->

```
<ModelPhantom userID="Housing 76" label="Housing 76">
  <Position X="76" Y="0" Z="-127"/>
  <Orientation RX="1.57" RY="-1.57" RZ="3.14"/>
  <File name="../models/computerdemo/housing.3ds"/>
</ModelPhantom>
```

```
<Model3D userID="S3C Shaft" label="S3C Shaft">
  <Position X="63.0" Y="0" Z="-125.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/shaft only.3ds"/>
  <Animation anim_name="S3C Shaft Animation">
    <Keyframe time="0" X="200" Y="0" Z="0" />
    <Keyframe time="1.5" X="0" Y="0" Z="0" />
  </Animation>
</Model3D>
```

```
<Model3D userID="S3C Seal1" label="S3C Seal1">
  <Position X="70.0" Y="0" Z="-125.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/seal lip.3ds"/>
  <Animation anim_name="S3C Seal1 Animation">
    <Keyframe time="0" X="200" Y="0" Z="0" />
    <Keyframe time="1.5" X="0" Y="0" Z="0" />
  </Animation>
```

</Model3D>

```
<Model3D userID="S3C Seal2" label="S3C Seal2">
  <Position X="73.0" Y="0" Z="-125.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="3.14"/>
  <File name="../models/computerdemo/seal lip.3ds"/>
  <Animation anim_name="S3C Seal2 Animation">
    <Keyframe time="0" X="-200" Y="0" Z="0" />
    <Keyframe time="1.5" X="0" Y="0" Z="0" />
  </Animation>
</Model3D>
```

<!-- STEP 4 -->

```
<Model3D userID="S4C Ring" label="S4C Ring">
  <Position X="100" Y="0" Z="-150"/>
  <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
  <File name="../models/computerdemo/retaining ring.3ds"/>
  <Animation anim_name="S4C Ring Animation">
    <Keyframe time="0" X="0" Y="0" Z="0"/>
    <Keyframe time="1" X="0" Y="0" Z="-100"/>
  </Animation>
</Model3D>
```

```
<Model3D userID="S4C Pliers" label="S4C Pliers">
  <Scale SX="1.8" SY="1.8" SZ="1.8"/>
  <Position X="160" Y="-110" Z="-150"/>
  <Orientation RX="0" RY="0" RZ="1.57"/>
  <File name="../models/computerdemo/90-degree pliers 2.3ds"/>
  <Animation anim_name="S4C Pliers Animation">
    <Keyframe time="0" X="0" Y="0" Z="0"/>
    <Keyframe time="1" X="0" Y="-55.556" Z="0"/>
  </Animation>
</Model3D>
```

<!-- STEP 5 -->

```
<Model3D userID="S5C Pin1" label="S5C Pin1">
  <Position X="-9.9" Y="-64" Z="65"/>
  <Orientation RX="0.0" RY="1.57" RZ="0.0"/>
  <File name="../models/computerdemo/hold-down pin.3ds"/>
  <Animation anim_name="S5C Pin1 Animation">
    <Keyframe time="0" X="0" Y="0" Z="0"/>
    <Keyframe time="1" X="-80" Y="0" Z="0"/>
  </Animation>
</Model3D>
```

```

    </Animation>
  </Model3D>

  <Model3D userID="S5C Pin2" label="S5C Pin2">
    <Position X="-9.9" Y="-90" Z="65"/>
    <Orientation RX="0.0" RY="1.57" RZ="0.0"/>
    <File name="../models/computerdemo/hold-down pin.3ds"/>
    <Animation anim_name="S5C Pin2 Animation">
      <Keyframe time="0" X="0" Y="0" Z="0"/>
      <Keyframe time="1" X="-80" Y="0" Z="0"/>
    </Animation>
  </Model3D>

  <Model3D userID="S5C Pin3" label="S5C Pin3">
    <Position X="11" Y="-78" Z="65"/>
    <Orientation RX="0.0" RY="1.57" RZ="0.0"/>
    <File name="../models/computerdemo/hold-down pin.3ds"/>
    <Animation anim_name="S5C Pin3 Animation">
      <Keyframe time="0" X="0" Y="0" Z="0"/>
      <Keyframe time="1" X="-80" Y="0" Z="0"/>
    </Animation>
  </Model3D>

  <!-- STEP 6 -->

  <Model3D userID="S6C Guide" label="S6C Guide">
    <Position X="0.0" Y="-77" Z="65"/>
    <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
    <File name="../models/computerdemo/retainer guide.3ds"/>
    <Animation anim_name="S6C Guide Animation">
      <Keyframe time="0" X="0" Y="0" Z="0" />
      <Keyframe time="1.75" X="80" Y="0" Z="0" />
    </Animation>
  </Model3D>

  <!-- STEP 7 -->

  <Model3D userID="S7C Retainer" label="S7C Retainer">
    <Position X="0.0" Y="-65" Z="70"/>
    <Orientation RX="1.57" RY="0.0" RZ="0.0"/>
    <File name="../models/computerdemo/retainer2.3ds"/>
    <Animation anim_name="S7C Retainer Animation">
      <Keyframe time="0" X="0" Y="80" Z="0" type="t" />
      <Keyframe time="1" X="0" Y="0" Z="0" type="t" />

```

```

        <Keyframe time="2" X="0" Y="0" Z="0" type="t" />

        <Keyframe time="0" X="0" Y="0" Z="0" type="r" />
        <Keyframe time="1" X="0" Y="0" Z="0" type="r" />
        <Keyframe time="2" X="0" Y="0.785" Z="0" type="r" />
    </Animation>
</Model3D>

<!-- STEP 8 -->
<!-- Models start with hole nearest marker and move CCW -->

<Model3D userID="S8C Slipper1" label="S8C Slipper1">
    <Position X="1" Y="-44" Z="40"/>
    <Orientation RX="0.0" RY="1.57" RZ="0.0"/>
    <File name="../models/computerdemo/slipper-piston.3ds"/>
    <!-- Early Animation -->
    <Animation anim_name="S8C Slipper1 Animation">
        <Keyframe time="0" X="-100" Y="0" Z="0" />
        <Keyframe time="1" X="0" Y="0" Z="0" />
        <Keyframe time="2" X="0" Y="0" Z="0" />
        <Keyframe time="2.5" X="0" Y="0" Z="0" />
    </Animation>
</Model3D>

<Model3D userID="S8C Slipper2" label="S8C Slipper2">
    <Position X="-15" Y="-50" Z="40"/>
    <Orientation RX="0.0" RY="1.57" RZ="0.0"/>
    <File name="../models/computerdemo/slipper-piston.3ds"/>
    <Animation anim_name="S8C Slipper2 Animation">
        <Keyframe time="0" X="-100" Y="0" Z="0" />
        <Keyframe time="1" X="-100" Y="0" Z="0" />
        <Keyframe time="2" X="0" Y="0" Z="0" />
        <Keyframe time="2.5" X="0" Y="0" Z="0" />
    </Animation>
</Model3D>

<Model3D userID="S8C Slipper3" label="S8C Slipper3">
    <Position X="-25" Y="-65" Z="40"/>
    <Orientation RX="0.0" RY="1.57" RZ="0.0"/>
    <File name="../models/computerdemo/slipper-piston.3ds"/>
    <Animation anim_name="S8C Slipper3 Animation">
        <Keyframe time="0" X="-100" Y="0" Z="0" />
        <Keyframe time="1" X="-100" Y="0" Z="0" />
        <Keyframe time="2" X="0" Y="0" Z="0" />
        <Keyframe time="2.5" X="0" Y="0" Z="0" />
    </Animation>
</Model3D>

```

```

</Animation>
</Model3D>

<Model3D userID="S8C Slipper4" label="S8C Slipper4">
  <Position X="-24.5" Y="-84" Z="40"/>
  <Orientation RX="0.0" RY="1.57" RZ="0.0"/>
  <File name="../models/computerdemo/slipper-piston.3ds"/>
  <Animation anim_name="S8C Slipper4 Animation">
    <Keyframe time="0" X="-100" Y="0" Z="0" />
    <Keyframe time="1" X="-100" Y="0" Z="0" />
    <Keyframe time="2" X="0" Y="0" Z="0" />
    <Keyframe time="2.5" X="0" Y="0" Z="0" />
  </Animation>
</Model3D>

<Model3D userID="S8C Slipper5" label="S8C Slipper5">
  <Position X="-12" Y="-98" Z="40"/>
  <Orientation RX="0.0" RY="1.57" RZ="0.0"/>
  <File name="../models/computerdemo/slipper-piston.3ds"/>
  <Animation anim_name="S8C Slipper5 Animation">
    <Keyframe time="0" X="-100" Y="0" Z="0" />
    <Keyframe time="1" X="-100" Y="0" Z="0" />
    <Keyframe time="2" X="0" Y="0" Z="0" />
    <Keyframe time="2.5" X="0" Y="0" Z="0" />
  </Animation>
</Model3D>

<Model3D userID="S8C Slipper6" label="S8C Slipper6">
  <Position X="8.8" Y="-99" Z="40"/>
  <Orientation RX="0.0" RY="1.57" RZ="0.0"/>
  <File name="../models/computerdemo/slipper-piston.3ds"/>
  <!-- Early Animation -->
  <Animation anim_name="S8C Slipper6 Animation">
    <Keyframe time="0" X="-100" Y="0" Z="0" />
    <Keyframe time="1" X="0" Y="0" Z="0" />
    <Keyframe time="2" X="0" Y="0" Z="0" />
    <Keyframe time="2.5" X="0" Y="0" Z="0" />
  </Animation>
</Model3D>

<Model3D userID="S8C Slipper7" label="S8C Slipper7">
  <Position X="24" Y="-84" Z="40"/>
  <Orientation RX="0.0" RY="1.57" RZ="0.0"/>
  <File name="../models/computerdemo/slipper-piston.3ds"/>
  <Animation anim_name="S8C Slipper7 Animation">
    <Keyframe time="0" X="-100" Y="0" Z="0" />

```

```

        <Keyframe time="1" X="-100" Y="0" Z="0" />
        <Keyframe time="2" X="0" Y="0" Z="0" />
        <Keyframe time="2.5" X="0" Y="0" Z="0" />
    </Animation>
</Model3D>

<Model3D userID="S8C Slipper8" label="S8C Slipper8">
    <Position X="28" Y="-64" Z="40"/>
    <Orientation RX="0.0" RY="1.57" RZ="0.0"/>
    <File name="../models/computerdemo/slipper-piston.3ds"/>
    <Animation anim_name="S8C Slipper8 Animation">
        <Keyframe time="0" X="-100" Y="0" Z="0" />
        <Keyframe time="1" X="-100" Y="0" Z="0" />
        <Keyframe time="2" X="0" Y="0" Z="0" />
        <Keyframe time="2.5" X="0" Y="0" Z="0" />
    </Animation>
</Model3D>

<Model3D userID="S8C Slipper9" label="S8C Slipper9">
    <Position X="19" Y="-49" Z="40"/>
    <Orientation RX="0.0" RY="1.57" RZ="0.0"/>
    <File name="../models/computerdemo/slipper-piston.3ds"/>
    <Animation anim_name="S8C Slipper9 Animation">
        <Keyframe time="0" X="-100" Y="0" Z="0" />
        <Keyframe time="1" X="-100" Y="0" Z="0" />
        <Keyframe time="2" X="0" Y="0" Z="0" />
        <Keyframe time="2.5" X="0" Y="0" Z="0" />
    </Animation>
</Model3D>

<!-- STEP 9 -->

<Model3D userID="S9C Shaft" label="S9C Shaft">
    <Position X="70.0" Y="0" Z="-110.0"/>
    <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
    <File name="../models/computerdemo/shaft only 2 semitrans.3ds"/>
</Model3D>

<Model3D userID="S9C Kit" label="S9C Kit">
    <Position X="55.0" Y="9" Z="-90.0"/>
    <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
    <File name="../models/computerdemo/kit subassembly.3ds"/>
    <Animation anim_name="S9C Kit Animation">
        <Keyframe time="0" X="-140" Y="0" Z="0" />
        <Keyframe time="1.5" X="0" Y="0" Z="0" />
    </Animation>
</Model3D>

```

```

        </Animation>
    </Model3D>

    <Content2D userID="S9AC Text" label="S9AC Text">
        <Position X="960.0" Y="960.0" Z="0.0"/>
        <File name="../models/computerdemo/S9AC Text.png"/>
    </Content2D>

    <!-- STEP 10 -->

    <Model3D userID="S10C Plate" label="S10C Plate">
        <Position X="-5.0" Y="-93" Z="45.0"/>
        <Orientation RX="-1.57" RY="0.0" RZ="3.14"/>
        <File name="../models/computerdemo/valve plate 2.3ds"/>
        <Animation anim_name="S10C Plate Animation">
            <Keyframe time="0" X="0" Y="-100" Z="0" />
            <Keyframe time="1.5" X="0" Y="0" Z="0" />
        </Animation>
    </Model3D>

    <!-- STEP 11 -->

    <Model3D userID="S11C Locator1" label="S11C Locator1">
        <Position X="-35" Y="58" Z="-92"/>
        <Orientation RX="0" RY="0" RZ="1.57"/>
        <File name="../models/computerdemo/hollow locator.3ds"/>
        <Animation anim_name="S11C Locator1 Animation">
            <Keyframe time="0" X="0" Y="100" Z="0" />
            <Keyframe time="1.5" X="0" Y="0" Z="0" />
        </Animation>
    </Model3D>

    <Model3D userID="S11C Locator2" label="S11C Locator2">
        <Position X="-35" Y="-40" Z="-165"/>
        <Orientation RX="0" RY="0" RZ="1.57"/>
        <File name="../models/computerdemo/hollow locator.3ds"/>
        <Animation anim_name="S11C Locator2 Animation">
            <Keyframe time="0" X="0" Y="100" Z="0" />
            <Keyframe time="1.5" X="0" Y="0" Z="0" />
        </Animation>
    </Model3D>

    <!-- STEP 12 -->

```



```

<Model3D userID="S12C Gasket" label="S12C Gasket">
  <Position X="-40.0" Y="15" Z="-112.0"/>
  <Orientation RX="1.57" RY="0.0" RZ="1.57"/>
  <File name="../models/computerdemo/gasket.3ds"/>
  <Animation anim_name="S12C Gasket Animation">
    <Keyframe time="0" X="0" Y="0" Z="-100" />
    <Keyframe time="1.5" X="0" Y="0" Z="0" />
  </Animation>
</Model3D>

```

```

<!-- STEP 13 -->

```

```

<Model3D userID="S13C Spring" label="S13C Spring">
  <Position X="-5" Y="40" Z="-45.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="1.57"/>
  <File name="../models/computerdemo/servo spring.3ds"/>
  <Animation anim_name="S13C Spring Animation">
    <Keyframe time="0" X="0" Y="100" Z="0" />
    <Keyframe time="1.5" X="0" Y="0" Z="0" />
  </Animation>
</Model3D>

```

```

<!-- STEP 14 -->

```

```

<Model3D userID="S14C Seat" label="S14C Seat">
  <Position X="-70" Y="0" Z="-45.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="-1.57"/>
  <File name="../models/computerdemo/spring seat.3ds"/>
  <Animation anim_name="S14C Seat Animation">
    <Keyframe time="0" X="0" Y="-100" Z="0" />
    <Keyframe time="1.5" X="0" Y="0" Z="0" />
  </Animation>
</Model3D>

```

```

<!-- STEP 15 -->

```

```

<Model3D userID="S15C Cap" label="S15C Cap">
  <Position X="102" Y="0" Z="-117"/>
  <Orientation RX="1.57" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/end cap subassembly.3ds"/>
  <Animation anim_name="S15C Cap Animation">
    <Keyframe time="0" X="-100" Y="0" Z="0" />

```

```

        <Keyframe time="1.5" X="0" Y="0" Z="0" />
    </Animation>
</Model3D>

<Content2D userID="S15AC Text" label="S15AC Text">
    <Position X="960.0" Y="960.0" Z="0.0"/>
    <File name="../../models/computerdemo/S15AC Text.png"/>
</Content2D>

<!-- STEP 16 -->

<ModelPhantom userID="S16C Cap" label="S16C Cap">
    <Position X="88" Y="0" Z="-127"/>
    <Orientation RX="1.57" RY="0.0" RZ="0.0"/>
    <File name="../../models/computerdemo/end cap subassembly.3ds"/>
</ModelPhantom>

<ModelPhantom userID="Housing 87" label="Housing 87">
    <Position X="87" Y="0" Z="-127"/>
    <Orientation RX="1.57" RY="-1.57" RZ="3.14"/>
    <File name="../../models/computerdemo/housing.3ds"/>
</ModelPhantom>

<Model3D userID="S16C Screw1" label="S16C Screw1">
    <Position X="-87" Y="0" Z="-20"/>
    <Orientation RX="1.57" RY="0.0" RZ="0.0"/>
    <File name="../../models/computerdemo/cap screw.3ds"/>
    <Animation anim_name="S16C Screw1 Animation">
        <Keyframe time="0" X="-100" Y="0" Z="0" />
        <Keyframe time="1.5" X="0" Y="0" Z="0" />
    </Animation>
</Model3D>

<Model3D userID="S16C Screw2" label="S16C Screw2">
    <Position X="-88" Y="55" Z="-95"/>
    <Orientation RX="1.57" RY="0.0" RZ="0.0"/>
    <File name="../../models/computerdemo/cap screw.3ds"/>
    <Animation anim_name="S16C Screw2 Animation">
        <Keyframe time="0" X="-100" Y="0" Z="0" />
        <Keyframe time="1.5" X="0" Y="0" Z="0" />
    </Animation>
</Model3D>

<Model3D userID="S16C Screw3" label="S16C Screw3">
    <Position X="-88" Y="40" Z="-170"/>

```

```

    <Orientation RX="1.57" RY="0.0" RZ="0.0"/>
    <File name="../models/computerdemo/cap screw.3ds"/>
    <Animation anim_name="S16C Screw3 Animation">
        <Keyframe time="0" X="-100" Y="0" Z="0" />
        <Keyframe time="1.5" X="0" Y="0" Z="0" />
    </Animation>
</Model3D>

<Model3D userID="S16C Screw4" label="S16C Screw4">
    <Position X="-88" Y="-40" Z="-170"/>
    <Orientation RX="1.57" RY="0.0" RZ="0.0"/>
    <File name="../models/computerdemo/cap screw.3ds"/>
    <Animation anim_name="S16C Screw4 Animation">
        <Keyframe time="0" X="-100" Y="0" Z="0" />
        <Keyframe time="1.5" X="0" Y="0" Z="0" />
    </Animation>
</Model3D>

<Model3D userID="S16C Screw5" label="S16C Screw5">
    <Position X="-88" Y="-53" Z="-95"/>
    <Orientation RX="1.57" RY="0.0" RZ="0.0"/>
    <File name="../models/computerdemo/cap screw.3ds"/>
    <Animation anim_name="S16C Screw5 Animation">
        <Keyframe time="0" X="-100" Y="0" Z="0" />
        <Keyframe time="1.5" X="0" Y="0" Z="0" />
    </Animation>
</Model3D>

<Model3D userID="S16C Wrench" label="S16C Wrench">
    <Position X="-95" Y="55" Z="-95"/>
    <Orientation RX="-1.57" RY="0.0" RZ="0.0"/>
    <File name="../models/computerdemo/Llave Allen 8 mm.3ds"/>
    <Animation anim_name="S16C Wrench Animation">
        <Keyframe time="0" X="-100" Y="0" Z="0" type="t"/>
        <Keyframe time="1.5" X="0" Y="0" Z="0" type="t"/>

        <Keyframe time="0" X="-6.28" Y="0" Z="0" type="r"/>
        <Keyframe time="1.5" X="0" Y="0" Z="0" type="r"/>
    </Animation>
</Model3D>

<!-- Dismissal -->

<Content2D userID="Dismissal Text" label="Dismissal Text">
    <Position X="960.0" Y="540.0" Z="0.0"/>

```

```

    <File name="../../models/computerdemo/Dismissal Text.png"/>
  </Content2D>

```

```

</Models>

```

```

<ARToolKit camera_config="camera_config" threshold="160" scale="1" >

```

```

  <Pattern id="0" >
    <File value="patt.kanji"/>
    <Dimension value="40"/>
    <Model3D userID="S1C Piston"/>
      <Model3D userID="S2C Piston"/>
      <Model3D userID="S2C Swashplate"/>
      <Model3D userID="S3C Shaft"/>
      <Model3D userID="S3C Seal1"/>
      <Model3D userID="S3C Seal2"/>
      <Model3D userID="Housing 76"/>
      <Model3D userID="S4C Pliers"/>
      <Model3D userID="S4C Ring"/>
      <Model3D userID="S9C Shaft"/>
      <Model3D userID="S9C Kit"/>
      <Model3D userID="S12C Gasket"/>
      <Model3D userID="S11C Locator1"/>
      <Model3D userID="S11C Locator2"/>
      <Model3D userID="S13C Spring"/>
      <Model3D userID="S14C Seat"/>
      <Model3D userID="S15C Cap"/>
      <Model3D userID="S16C Cap"/>
      <Model3D userID="S16C Screw1"/>
      <Model3D userID="S16C Screw2"/>
      <Model3D userID="S16C Screw3"/>
      <Model3D userID="S16C Screw4"/>
      <Model3D userID="S16C Screw5"/>
      <Model3D userID="S16C Wrench"/>
      <Model3D userID="Housing 87"/>
    </Pattern>

```

```

  <Pattern id="1" >
    <File value="patt.sample1"/>
    <Dimension value="40"/>
      <Model3D userID="S5C Pin1"/>
      <Model3D userID="S5C Pin2"/>
      <Model3D userID="S5C Pin3"/>
      <Model3D userID="S6C Guide"/>
      <Model3D userID="S7C Retainer"/>

```

```

        <Model3D userID="S8C Slipper1"/>
        <Model3D userID="S8C Slipper2"/>
        <Model3D userID="S8C Slipper3"/>
        <Model3D userID="S8C Slipper4"/>
        <Model3D userID="S8C Slipper5"/>
        <Model3D userID="S8C Slipper6"/>
        <Model3D userID="S8C Slipper7"/>
        <Model3D userID="S8C Slipper8"/>
        <Model3D userID="S8C Slipper9"/>
    </Pattern>

    <Pattern id="2" >
    <File value="patt.sample2"/>
    <Dimension value="40"/>
        <Model3D userID="S10C Plate"/>
    </Pattern>

</ARToolKit>

<!--<Menu>
    <MenuButton id="BackButton" size_x="80" size_y="80" pos_x="50"
pos_y="50" url="../models/computerdemo/buttons/back.png" toggle="false" />
    <MenuButton id="ReadyButton" parent="BackButton" size_x="80" size_y="80"
pos_x="1800" pos_y="50" url="../models/computerdemo/buttons/ready.png" toggle="false" />
</Menu>-->

<Configurations startConfig="Start Screen" >

    <Configuration configID="Start Screen" >
        <Content userID="Startup Text" />
        <Content userID="Startup Text 2" />
    </Configuration>

    <Configuration configID="S1C" >
        <Content userID="S1C Piston" />
        <Content userID="Housing 87" />
    </Configuration>

    <Configuration configID="S2C" >
        <Content userID="S2C Piston" />
        <Content userID="S2C Swashplate" />
    </Configuration>

    <Configuration configID="S3C" >

```

```

        <Content userID="S3C Shaft" />
        <Content userID="S3C Seal1" />
        <Content userID="S3C Seal2" />
        <Content userID="Housing 76" />
    </Configuration>

    <Configuration configID="S4C" >
        <Content userID="S4C Ring" />
        <Content userID="S4C Pliers" />
        <Content userID="Housing 76" />
    </Configuration>

    <Configuration configID="S5C" >
        <Content userID="S5C Pin1" />
        <Content userID="S5C Pin2" />
        <Content userID="S5C Pin3" />
    </Configuration>

    <Configuration configID="S6C" >
        <Content userID="S6C Guide" />
    </Configuration>

    <Configuration configID="S7C" >
        <Content userID="S7C Retainer" />
    </Configuration>

    <Configuration configID="S8C" >
        <Content userID="S8C Slipper1" />
        <Content userID="S8C Slipper2" />
        <Content userID="S8C Slipper3" />
        <Content userID="S8C Slipper4" />
        <Content userID="S8C Slipper5" />
        <Content userID="S8C Slipper6" />
        <Content userID="S8C Slipper7" />
        <Content userID="S8C Slipper8" />
        <Content userID="S8C Slipper9" />
    </Configuration>

    <Configuration configID="S9C" >
        <Content userID="S9C Shaft" />
        <Content userID="S9C Kit" />
        <Content userID="S9AC Text" />
    </Configuration>

    <Configuration configID="S10C" >
        <Content userID="S10C Plate" />

```

```

</Configuration>

<Configuration configID="S11C" >
    <Content userID="S11C Locator1" />
    <Content userID="S11C Locator2" />
</Configuration>

<Configuration configID="S12C" >
    <Content userID="S12C Gasket" />
</Configuration>

<Configuration configID="S13C" >
    <Content userID="S13C Spring" />
    <Content userID="Housing 87" />
</Configuration>

<Configuration configID="S14C" >
    <Content userID="S14C Seat" />
    <Content userID="Housing 87" />
</Configuration>

<Configuration configID="S15C" >
    <Content userID="S15C Cap" />
    <Content userID="S15AC Text" />
</Configuration>

<Configuration configID="S16C" >
    <Content userID="S16C Cap" />
    <Content userID="Housing 87" />
    <Content userID="S16C Screw1" />
    <Content userID="S16C Screw2" />
    <Content userID="S16C Screw3" />
    <Content userID="S16C Screw4" />
    <Content userID="S16C Screw5" />
    <Content userID="S16C Wrench" />
</Configuration>

<Configuration configID="Dismissal" >
    <Content userID="Dismissal Text" />
</Configuration>

</Configurations>

</ns1:ARPrototypingToolkit>

```

Motor_AR_appA.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<ns1:ARPrototypingToolkit xmlns:ns1="ARProToXML"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://vrac.iastate.edu/~rafael/xml/ARProTo.xsd">
  <Statemachine file=" ../models/computerdemo/pump_statemachine_A.py" />

  <Models>

    <!-- Start Screen -->

    <Content2D userID="Startup Text" label="Startup Text">
      <Position X="960.0" Y="120.0" Z="0.0"/>
      <File name=" ../models/computerdemo/Startup Text.png"/>
    </Content2D>

    <Content2D userID="Startup Text 2" label="Startup Text 2">
      <Position X="960.0" Y="960.0" Z="0.0"/>
      <File name=" ../models/computerdemo/Startup Text 2.png"/>
    </Content2D>

    <!-- STEP 1 -->

    <Model3D userID="S1A Arrow" label="S1A Arrow">
      <Position X="-0.5" Y="0.0" Z="-0.4"/>
      <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
      <File name=" ../models/computerdemo/Arrow.3ds"/>
    </Model3D>

    <Model3D userID="S1A Frame" label="S1A Frame">
      <Scale SX="1.2" SY="1.2" SZ="1.2"/>
      <Position X="5.9" Y="-4.0" Z="0.0"/>
      <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
      <File name=" ../models/computerdemo/frame-square1.3ds"/>
    </Model3D>

    <Content2D userID="S1A Text" label="S1A Text">
      <Position X="960.0" Y="120.0" Z="0.0"/>
      <File name=" ../models/computerdemo/S1A Text.png"/>
    </Content2D>

    <!-- STEP 2 -->

```



```

<Model3D userID="S2A Arrow1" label="S2A Arrow1">
  <Position X="-0.5" Y="0.0" Z="-1.1"/>
  <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow.3ds"/>
</Model3D>

<Model3D userID="S2A Arrow2" label="S2A Arrow2">
  <Position X="0.4" Y="0.0" Z="-0.6"/>
  <Orientation RX="0.0" RY="-2.655" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

<Model3D userID="S2A Frame" label="S2A Frame">
  <Scale SX="2" SY="2" SZ="2"/>
  <Position X="4.1" Y="-3.2" Z="0.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/frame-square1.3ds"/>
</Model3D>

<Content2D userID="S2A Text" label="S2A Text">
  <Position X="960.0" Y="120.0" Z="0.0"/>
  <File name="../models/computerdemo/S2A Text.png"/>
</Content2D>

<Content2D userID="S2A Sketch" label="S2A Sketch">
  <Position X="300.0" Y="880.0" Z="0.0"/>
  <File name="../models/computerdemo/Sketch S2.png"/>
</Content2D>

<!-- STEP 3 -->

<Model3D userID="S3A Arrow" label="S3A Arrow">
  <Position X="1.2" Y="0.0" Z="-0.9"/>
  <Orientation RX="0.0" RY="1.57" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow.3ds"/>
</Model3D>

<Model3D userID="S3A Frame" label="S3A Frame">
  <Scale SX="1.8" SY="1.8" SZ="1.8"/>
  <Position X="5.4" Y="0.2" Z="0.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/frame-rect1.3ds"/>
</Model3D>

```

<!-- STEP 4 -->

```
<Model3D userID="S4A Arrow" label="S4A Arrow">
  <Position X="1.2" Y="0.0" Z="-0.9"/>
  <Orientation RX="0.0" RY="1.57" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow.3ds"/>
</Model3D>

<Model3D userID="S4A Frame1" label="S4A Frame1">
  <Scale SX="1.5" SY="1.7" SZ="1.5"/>
  <Position X="4.2" Y="-4.2" Z="0.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/frame-square1.3ds"/>
</Model3D>

<Model3D userID="S4A Frame2" label="S4A Frame2">
  <Scale SX="1.8" SY="1.8" SZ="1.8"/>
  <Position X="2" Y="-0.2" Z="0.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="1.57"/>
  <File name="../models/computerdemo/frame-rect1-red.3ds"/>
  <Animation anim_name="S4A Frame2 Animation">
    <Keyframe time="0" X="0" Y="0" Z="0" />
    <Keyframe time="1" X="0" Y="0" Z="0.50" />
  </Animation>
</Model3D>
```

<!-- STEP 5 -->

```
<Model3D userID="S5A Arrow1" label="S5A Arrow1">
  <Position X="-0.099" Y="-0.64" Z="0.7"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

<Model3D userID="S5A Arrow2" label="S5A Arrow2">
  <Position X="-0.099" Y="-0.92" Z="0.7"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

<Model3D userID="S5A Arrow3" label="S5A Arrow3">
  <Position X="0.11" Y="-0.8" Z="0.7"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>
```

```
<Model3D userID="S5A Frame" label="S5A Frame">
  <Position X="3.5" Y="-2.55" Z="0.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/frame-square1.3ds"/>
</Model3D>
```

```
<!-- STEP 6 -->
```

```
<Model3D userID="S6A Arrow" label="S6A Arrow">
  <Position X="0" Y="-0.77" Z="0.7"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow.3ds"/>
</Model3D>
```

```
<Model3D userID="S6A Frame" label="S6A Frame">
  <Scale SX="0.8" SY="0.8" SZ="0.8"/>
  <Position X="5.1" Y="-4.4" Z="0.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/frame-square1.3ds"/>
</Model3D>
```

```
<Content2D userID="S6A Sketch" label="S6A Sketch">
  <Position X="300.0" Y="880.0" Z="0.0"/>
  <File name="../models/computerdemo/Sketch S6.png"/>
</Content2D>
```

```
<!-- STEP 7 -->
```

```
<Model3D userID="S7A Arrow" label="S7A Arrow">
  <Position X="0" Y="-0.77" Z="0.85"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow.3ds"/>
</Model3D>
```

```
<Model3D userID="S7A Frame" label="S7A Frame">
  <Scale SX="1.6" SY="1.6" SZ="1.6"/>
  <Position X="3.4" Y="-1.7" Z="0.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/frame-square1.3ds"/>
</Model3D>
```

```
<Content2D userID="S7A Text" label="S7A Text">
  <Position X="960.0" Y="120.0" Z="0.0"/>
```

```

    <File name="../models/computerdemo/S7A Text.png"/>
  </Content2D>

```

```

<!-- STEP 8 -->

```

```

<!-- Arrows start with hole nearest marker and move CCW -->

```

```

<Model3D userID="S8A Arrow1" label="S8A Arrow1">
  <Position X="0.01" Y="-0.44" Z="0.7"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

```

```

<Model3D userID="S8A Arrow2" label="S8A Arrow2">
  <Position X="-0.15" Y="-0.5" Z="0.7"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

```

```

<Model3D userID="S8A Arrow3" label="S8A Arrow3">
  <Position X="-0.25" Y="-0.65" Z="0.7"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

```

```

<Model3D userID="S8A Arrow4" label="S8A Arrow4">
  <Position X="-0.245" Y="-0.84" Z="0.7"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

```

```

<Model3D userID="S8A Arrow5" label="S8A Arrow5">
  <Position X="-0.12" Y="-0.98" Z="0.7"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

```

```

<Model3D userID="S8A Arrow6" label="S8A Arrow6">
  <Position X="0.088" Y="-0.99" Z="0.7"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

```

```

<Model3D userID="S8A Arrow7" label="S8A Arrow7">
  <Position X="0.24" Y="-0.84" Z="0.7"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>

```

```

        <File name="../models/computerdemo/Arrow-sm.3ds"/>
    </Model3D>

    <Model3D userID="S8A Arrow8" label="S8A Arrow8">
        <Position X="0.28" Y="-0.64" Z="0.7"/>
        <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
        <File name="../models/computerdemo/Arrow-sm.3ds"/>
    </Model3D>

    <Model3D userID="S8A Arrow9" label="S8A Arrow9">
        <Position X="0.19" Y="-0.49" Z="0.7"/>
        <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
        <File name="../models/computerdemo/Arrow-sm.3ds"/>
    </Model3D>

    <Model3D userID="S8A Frame" label="S8A Frame">
        <Scale SX="1.8" SY="1.5" SZ="1.5"/>
        <Position X="4.3" Y="-0.9" Z="0.0"/>
        <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
        <File name="../models/computerdemo/frame-square1.3ds"/>
    </Model3D>

    <!-- STEP 9 -->

    <Model3D userID="S9A Arrow" label="S9A Arrow">
        <Position X="-0.8" Y="0.0" Z="-1.1"/>
        <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
        <File name="../models/computerdemo/Arrow.3ds"/>
    </Model3D>

    <Model3D userID="S9A Frame" label="S9A Frame">
        <Scale SX="1.5" SY="1.5" SZ="1.5"/>
        <Position X="0.0" Y="-0.7" Z="0.0"/>
        <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
        <File name="../models/computerdemo/frame-square1.3ds"/>
    </Model3D>

    <Content2D userID="S9A Text" label="S9A Text">
        <Position X="960.0" Y="120.0" Z="0.0"/>
        <File name="../models/computerdemo/S9A Text.png"/>
    </Content2D>

    <Content2D userID="S9AC Text" label="S9AC Text">
        <Position X="960.0" Y="960.0" Z="0.0"/>
        <File name="../models/computerdemo/S9AC Text.png"/>

```

</Content2D>

<!-- STEP 10 -->

```
<Model3D userID="S10A Arrow1" label="S10A Arrow1">
  <Position X="0.0" Y="-1.33" Z="0.5"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow.3ds"/>
</Model3D>
```

```
<Model3D userID="S10A Arrow2" label="S10A Arrow2">
  <Position X="0.0" Y="-0.5" Z="0.5"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>
```

```
<Model3D userID="S10A Frame" label="S10A Frame">
  <Scale SX="1.8" SY="1.8" SZ="1.8"/>
  <Position X="5.0" Y="-3.5" Z="0.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../models/computerdemo/frame-square1.3ds"/>
</Model3D>
```

```
<Content2D userID="S10A Text" label="S10A Text">
  <Position X="960.0" Y="120.0" Z="0.0"/>
  <File name="../models/computerdemo/S10A Text.png"/>
</Content2D>
```

<!-- STEP 11 -->

```
<Model3D userID="S11A Arrow1" label="S11A Arrow1">
  <Position X="-0.6" Y="-0.4" Z="-1.55"/>
  <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow.3ds"/>
</Model3D>
```

```
<Model3D userID="S11A Arrow2" label="S11A Arrow2">
  <Position X="-0.6" Y="0.53" Z="-0.85"/>
  <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow.3ds"/>
</Model3D>
```

```
<Model3D userID="S11A Frame" label="S11A Frame">
  <Scale SX="0.9" SY="0.9" SZ="0.9"/>
```

```

    <Position X="3.0" Y="-2.5" Z="0.0"/>
    <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
    <File name="../models/computerdemo/frame-square1.3ds"/>
  </Model3D>

```

```

<!-- STEP 12 -->
<!-- Arrows start with top hole and move CCW -->

```

```

<Model3D userID="S12A Arrow1" label="S12A Arrow1">
  <Position X="-0.45" Y="0.0" Z="-0.1"/>
  <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

```

```

<Model3D userID="S12A Arrow2" label="S12A Arrow2">
  <Position X="-0.45" Y="0.55" Z="-0.85"/>
  <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

```

```

<Model3D userID="S12A Arrow3" label="S12A Arrow3">
  <Position X="-0.45" Y="0.4" Z="-1.6"/>
  <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

```

```

<Model3D userID="S12A Arrow4" label="S12A Arrow4">
  <Position X="-0.45" Y="-0.4" Z="-1.6"/>
  <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

```

```

<Model3D userID="S12A Arrow5" label="S12A Arrow5">
  <Position X="-0.45" Y="-0.53" Z="-0.9"/>
  <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

```

```

<Model3D userID="S12A Arrow6" label="S12A Arrow6">
  <Position X="-0.45" Y="-.19" Z="-0.6"/>
  <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
  <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

```

```

<Model3D userID="S12A Frame" label="S12A Frame">

```

```

    <Scale SX="1.8" SY="2.5" SZ="2.2"/>
    <Position X="3.5" Y="0.0" Z="0.0"/>
    <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
    <File name="../../models/computerdemo/frame-rect1.3ds"/>
  </Model3D>

```

```

<!-- STEP 13 -->

```

```

<Model3D userID="S13A Arrow" label="S13A Arrow">
  <Position X="-0.5" Y="0.0" Z="-0.4"/>
  <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
  <File name="../../models/computerdemo/Arrow.3ds"/>
</Model3D>

```

```

<Model3D userID="S13A Frame" label="S13A Frame">
  <Scale SX="1.2" SY="1.2" SZ="1.2"/>
  <Position X="5.9" Y="-3.2" Z="0.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../../models/computerdemo/frame-square1.3ds"/>
</Model3D>

```

```

<!-- STEP 14 -->

```

```

<Model3D userID="S14A Arrow" label="S14A Arrow">
  <Position X="-0.8" Y="0.0" Z="-0.4"/>
  <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
  <File name="../../models/computerdemo/Arrow.3ds"/>
</Model3D>

```

```

<Model3D userID="S14A Frame" label="S14A Frame">
  <Scale SX="0.8" SY="0.8" SZ="0.8"/>
  <Position X="5.7" Y="-4.4" Z="0.0"/>
  <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
  <File name="../../models/computerdemo/frame-square1.3ds"/>
</Model3D>

```

```

<!-- STEP 15 -->

```

```

<!-- Shaft -->

```

```

<Model3D userID="S15A Arrow1" label="S15A Arrow1">
  <Position X="-0.8" Y="0.0" Z="-1.1"/>
  <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
  <File name="../../models/computerdemo/Arrow.3ds"/>

```



```

</Model3D>

<!-- Locator1 -->
<Model3D userID="S15A Arrow2" label="S15A Arrow2">
    <Position X="-0.6" Y="-0.4" Z="-1.55"/>
    <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
    <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

<!-- Locator2 -->
<Model3D userID="S15A Arrow3" label="S15A Arrow3">
    <Position X="-0.6" Y="0.53" Z="-0.85"/>
    <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
    <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

<!-- Spring Seat -->
<Model3D userID="S15A Arrow4" label="S15A Arrow4">
    <Position X="-0.8" Y="0.0" Z="-0.4"/>
    <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
    <File name="../models/computerdemo/Arrow.3ds"/>
</Model3D>

<Model3D userID="S15A Frame" label="S15A Frame">
    <Scale SX="1.8" SY="2.5" SZ="2.2"/>
    <Position X="0.0" Y="-0.95" Z="0.2"/>
    <Orientation RX="0.0" RY="0.0" RZ="1.57"/>
    <File name="../models/computerdemo/frame-rect1.3ds"/>
</Model3D>

<Content2D userID="S15AC Text" label="S15AC Text">
    <Position X="960.0" Y="960.0" Z="0.0"/>
    <File name="../models/computerdemo/S15AC Text.png"/>
</Content2D>

<!-- STEP 16 -->

<Model3D userID="S16A Arrow1" label="S16A Arrow1">
    <Position X="-0.85" Y="0.0" Z="-0.1"/>
    <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
    <File name="../models/computerdemo/Arrow-sm.3ds"/>
</Model3D>

<Model3D userID="S16A Arrow2" label="S16A Arrow2">
    <Position X="-0.85" Y="0.55" Z="-0.85"/>

```

```

        <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
        <File name="../models/computerdemo/Arrow-sm.3ds"/>
    </Model3D>

    <Model3D userID="S16A Arrow3" label="S16A Arrow3">
        <Position X="-0.85" Y="0.4" Z="-1.6"/>
        <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
        <File name="../models/computerdemo/Arrow-sm.3ds"/>
    </Model3D>

    <Model3D userID="S16A Arrow4" label="S16A Arrow4">
        <Position X="-0.85" Y="-0.4" Z="-1.6"/>
        <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
        <File name="../models/computerdemo/Arrow-sm.3ds"/>
    </Model3D>

    <Model3D userID="S16A Arrow5" label="S16A Arrow5">
        <Position X="-0.85" Y="-0.53" Z="-0.9"/>
        <Orientation RX="0.0" RY="-1.57" RZ="0.0"/>
        <File name="../models/computerdemo/Arrow-sm.3ds"/>
    </Model3D>

    <Model3D userID="S16A Frame1" label="S16A Frame1">
        <Scale SX="1.3" SY="1.3" SZ="1.3"/>
        <Position X="3.2" Y="-0.9" Z="0.0"/>
        <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
        <File name="../models/computerdemo/frame-square1.3ds"/>
    </Model3D>

    <Model3D userID="S16A Frame2" label="S16A Frame2">
        <Scale SX="1.2" SY="1.2" SZ="1.2"/>
        <Position X="1.1" Y="0.0" Z="0.0"/>
        <Orientation RX="0.0" RY="0.0" RZ="1.57"/>
        <File name="../models/computerdemo/frame-rect1-red.3ds"/>
        <Animation anim_name="S16A Frame2 Animation">
            <Keyframe time="0" X="0" Y="0" Z="0" />
            <Keyframe time="1" X="0" Y="0" Z="0.50" />
        </Animation>
    </Model3D>

    <!-- Dismissal -->

    <Content2D userID="Dismissal Text" label="Dismissal Text">
        <Position X="960.0" Y="540.0" Z="0.0"/>
        <File name="../models/computerdemo/Dismissal Text.png"/>

```

</Content2D>

<!--TESTING-->

<Model3D userID="S1Ab Frame" label="S1Ab Frame">
 <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
 <File name="../models/computerdemo/frame-rect1.3ds"/>
 </Model3D>

<Model3D userID="S1Ac Frame" label="S1Ac Frame">
 <Orientation RX="0.0" RY="0.0" RZ="0.0"/>
 <File name="../models/computerdemo/frame-rect1.3ds"/>
 </Model3D>
 <!--/TESTING-->

</Models>

<ARToolKit camera_config="camera_config" threshold="160" scale="100" >

<Pattern id="0" >
 <File value="patt.kanji"/>
 <Dimension value="40"/>
 <Model3D userID="S1A Arrow"/>
 <Model3D userID="S2A Arrow1"/>
 <Model3D userID="S2A Arrow2"/>
 <Model3D userID="S3A Arrow"/>
 <Model3D userID="S4A Arrow"/>
 <Model3D userID="S9A Arrow"/>
 <Model3D userID="S11A Arrow1"/>
 <Model3D userID="S11A Arrow2"/>
 <Model3D userID="S12A Arrow1"/>
 <Model3D userID="S12A Arrow2"/>
 <Model3D userID="S12A Arrow3"/>
 <Model3D userID="S12A Arrow4"/>
 <Model3D userID="S12A Arrow5"/>
 <Model3D userID="S12A Arrow6"/>
 <Model3D userID="S13A Arrow"/>
 <Model3D userID="S14A Arrow"/>
 <Model3D userID="S15A Arrow1"/>
 <Model3D userID="S15A Arrow2"/>
 <Model3D userID="S15A Arrow3"/>
 <Model3D userID="S15A Arrow4"/>
 <Model3D userID="S16A Arrow1"/>
 <Model3D userID="S16A Arrow2"/>

```

    <Model3D userID="S16A Arrow3"/>
    <Model3D userID="S16A Arrow4"/>
    <Model3D userID="S16A Arrow5"/>
  </Pattern>

  <Pattern id="1" >
    <File value="patt.hiro"/>
    <Dimension value="80"/>
    <Model3D userID="S1A Frame"/>
    <Model3D userID="S2A Frame"/>
    <Model3D userID="S3A Frame"/>
    <Model3D userID="S4A Frame1"/>
    <Model3D userID="S4A Frame2"/>
    <Model3D userID="S5A Frame"/>
    <Model3D userID="S6A Frame"/>
    <Model3D userID="S7A Frame"/>
    <Model3D userID="S8A Frame"/>
    <Model3D userID="S10A Frame"/>
    <Model3D userID="S11A Frame"/>
    <Model3D userID="S12A Frame"/>
    <Model3D userID="S13A Frame"/>
    <Model3D userID="S14A Frame"/>
    <Model3D userID="S16A Frame1"/>
    <Model3D userID="S16A Frame2"/>
  </Pattern>

  <Pattern id="2" >
    <File value="patt.sample1"/>
    <Dimension value="40"/>
    <Model3D userID="S5A Arrow1"/>
    <Model3D userID="S5A Arrow2"/>
    <Model3D userID="S5A Arrow3"/>
    <Model3D userID="S6A Arrow"/>
    <Model3D userID="S7A Arrow"/>
    <Model3D userID="S8A Arrow1"/>
    <Model3D userID="S8A Arrow2"/>
    <Model3D userID="S8A Arrow3"/>
    <Model3D userID="S8A Arrow4"/>
    <Model3D userID="S8A Arrow5"/>
    <Model3D userID="S8A Arrow6"/>
    <Model3D userID="S8A Arrow7"/>
    <Model3D userID="S8A Arrow8"/>
    <Model3D userID="S8A Arrow9"/>
    <Model3D userID="S9A Frame"/>
    <!--<Model3D userID="S1Ab Frame"/>-->
  </Pattern>

```

```

<Pattern id="3" >
  <File value="patt.sample2"/>
  <Dimension value="40"/>
    <Model3D userID="S10A Arrow1"/>
    <Model3D userID="S10A Arrow2"/>
    <Model3D userID="S15A Frame"/>
    <!--<Model3D userID="S1Ac Frame"/>-->
  </Pattern>

</ARToolKit>

<!--<Menu>
  <MenuButton id="BackButton" size_x="80" size_y="80" pos_x="50"
pos_y="50" url="../models/computerdemo/buttons/back.png" toggle="false" />
  <MenuButton id="ReadyButton" parent="BackButton" size_x="80" size_y="80"
pos_x="1800" pos_y="50" url="../models/computerdemo/buttons/ready.png" toggle="false" />
</Menu>-->

<Configurations startConfig="Start Screen" >

  <Configuration configID="Start Screen" >
    <!--<Content userID="S1Ab Frame" />
    <Content userID="S1Ac Frame" />-->
    <Content userID="Startup Text" />
    <Content userID="Startup Text 2" />
  </Configuration>

  <Configuration configID="S1A" >
    <Content userID="S1A Arrow" />
    <Content userID="S1A Frame" />
    <Content userID="S1A Text" />
  </Configuration>

  <Configuration configID="S2A" >
    <Content userID="S2A Arrow1" />
    <Content userID="S2A Arrow2" />
    <Content userID="S2A Frame" />
    <Content userID="S2A Text" />
    <Content userID="S2A Sketch" />
  </Configuration>

  <Configuration configID="S3A" >
    <Content userID="S3A Arrow" />

```

```

        <Content userID="S3A Frame" />
    </Configuration>

    <Configuration configID="S4A" >
        <Content userID="S4A Arrow" />
        <Content userID="S4A Frame1" />
        <Content userID="S4A Frame2" />
    </Configuration>

    <Configuration configID="S5A" >
        <Content userID="S5A Arrow1" />
        <Content userID="S5A Arrow2" />
        <Content userID="S5A Arrow3" />
        <Content userID="S5A Frame" />
    </Configuration>

    <Configuration configID="S6A" >
        <Content userID="S6A Arrow" />
        <Content userID="S6A Frame" />
        <Content userID="S6A Sketch" />
    </Configuration>

    <Configuration configID="S7A" >
        <Content userID="S7A Arrow" />
        <Content userID="S7A Frame" />
        <Content userID="S7A Text" />
    </Configuration>

    <Configuration configID="S8A" >
        <Content userID="S8A Arrow1" />
        <Content userID="S8A Arrow2" />
        <Content userID="S8A Arrow3" />
        <Content userID="S8A Arrow4" />
        <Content userID="S8A Arrow5" />
        <Content userID="S8A Arrow6" />
        <Content userID="S8A Arrow7" />
        <Content userID="S8A Arrow8" />
        <Content userID="S8A Arrow9" />
        <Content userID="S8A Frame" />
    </Configuration>

    <Configuration configID="S9A" >
        <Content userID="S9A Arrow" />
        <Content userID="S9A Frame" />
        <Content userID="S9A Text" />
        <Content userID="S9AC Text" />

```

</Configuration>

```
<Configuration configID="S10A" >
  <Content userID="S10A Arrow1" />
  <Content userID="S10A Arrow2" />
  <Content userID="S10A Frame" />
  <Content userID="S10A Text" />
</Configuration>
```

```
<Configuration configID="S11A" >
  <Content userID="S11A Arrow1" />
  <Content userID="S11A Arrow2" />
  <Content userID="S11A Frame" />
</Configuration>
```

```
<Configuration configID="S12A" >
  <Content userID="S12A Arrow1" />
  <Content userID="S12A Arrow2" />
  <Content userID="S12A Arrow3" />
  <Content userID="S12A Arrow4" />
  <Content userID="S12A Arrow5" />
  <Content userID="S12A Arrow6" />
  <Content userID="S12A Frame" />
</Configuration>
```

```
<Configuration configID="S13A" >
  <Content userID="S13A Arrow" />
  <Content userID="S13A Frame" />
</Configuration>
```

```
<Configuration configID="S14A" >
  <Content userID="S14A Arrow" />
  <Content userID="S14A Frame" />
</Configuration>
```

```
<Configuration configID="S15A" >
  <Content userID="S15A Arrow1" />
  <Content userID="S15A Arrow2" />
  <Content userID="S15A Arrow3" />
  <Content userID="S15A Arrow4" />
  <Content userID="S15A Frame" />
  <Content userID="S15AC Text" />
</Configuration>
```

```
<Configuration configID="S16A" >
  <Content userID="S16A Arrow1" />
```



```

        <Content userID="S16A Arrow2" />
        <Content userID="S16A Arrow3" />
        <Content userID="S16A Arrow4" />
        <Content userID="S16A Arrow5" />
        <Content userID="S16A Frame1" />
        <Content userID="S16A Frame2" />
    </Configuration>

    <Configuration configID="Dismissal" >
        <Content userID="Dismissal Text" />
    </Configuration>

    <Configuration configID="Debug" >
        <Content userID="S1A Frame" />
        <Content userID="S2A Frame" />
        <Content userID="S3A Frame" />
        <Content userID="S4A Frame1" />
        <Content userID="S4A Frame2" />
        <Content userID="S5A Frame" />
        <Content userID="S6A Frame" />
        <Content userID="S7A Frame" />
        <Content userID="S8A Frame" />
        <Content userID="S9A Frame" />
        <Content userID="S10A Frame" />
        <Content userID="S11A Frame" />
        <Content userID="S12A Frame" />
        <Content userID="S13A Frame" />
        <Content userID="S14A Frame" />
        <Content userID="S15A Frame" />
        <Content userID="S16A Frame1" />
        <Content userID="S16A Frame2" />
    </Configuration>

</Configurations>

</ns1:ARPrototypingToolkit>

```

APPENDIX F – PYTHON SCRIPTS

motor_statemachine_C.py

```

from ctypes import *
ar = cdll.LoadLibrary("ARPyInt.dll")

# Right Arrow = 65363
# Left Arrow = 65361
scene_list = ["Start Screen", "S1C", "S2C", "S3C", "S4C", "S5C", "S6C", "S7C", "S8C", "S9C",
"S10C", "S11C", "S12C", "S13C", "S14C", "S15C", "S16C", "Dismissal"]
current = 0
max = 17

def onKeyboard(key):
    print "[Py] Key pressed:", key
    if key == 65363:      #Right Arrow
        global current
        global max
        if current < max:
            current = current+1
            print "Switch to scene: " , scene_list[current]
            ar.setScene(scene_list[current])

        else:
            print "Already on the last scene."

    if key == 65361:      #Left Arrow
        global current
        if current > 0:
            current = current-1
            print "Switch to scene: " , scene_list[current]
            ar.setScene(scene_list[current])

        else:
            print "Already on the first scene."

    if key == 100: #'D'
        print "Switch to scene: Debug" ,
        ar.setScene("Debug")

def onMenuButton(button):
    if button == "ReadyButton":
        ar.enableModel("pump");
        ar.disableModel("boarded");
    if button == "ReadyButton2":

```

```

        ar.enableModel("boarded");
        ar.disableModel("pump");
    print "[Py] Button pressed :", button

def onPatternAction(pattern):
    print "Action pattern :", pattern

def onFrame(frameid):
    n = 6
    #print "Frame number :", frameid

def onInit(frameid):
    print "INIT Application"
    ar.screenFlipHorizontal()
    ar.screenFlipVertical()

# Fires when a mouse event occurs
# model: a string that contains the model id
# model: a string that contains the command, TOUCH, FIRE, RELEASE, UNTOUCH
def onMouseEvent(model, command):
    print "Mouse event ", model, " : ", command

    if command == "TOUCH":
        ar.highlightModelOn(model)

    if command == "UNTOUCH":
        ar.highlightModelOff(model)

```

motor_statemachine_A.py

```

from ctypes import *
ar = cdll.LoadLibrary("ARPyInt.dll")

# Right Arrow = 65363
# Left Arrow = 65361
scene_list = ["Start Screen","S1A","S2A","S3A","S4A","S5A","S6A","S7A","S8A","S9A",
"S10A","S11A","S12A","S13A","S14A","S15A","S16A","Dismissal"]
current = 0
max = 17

def onKeyboard(key):
    print "[Py] Key pressed:", key
    if key == 65363:      #Right Arrow
        global current
        global max
        if current < max:
            current = current+1
            print "Switch to scene: " , scene_list[current]
            ar.setScene(scene_list[current])

        else:
            print "Already on the last scene."

    if key == 65361:      #Left Arrow
        global current
        if current > 0:
            current = current-1
            print "Switch to scene: " , scene_list[current]
            ar.setScene(scene_list[current])
        else:
            print "Already on the first scene."

    if key == 100: # 'D'
        print "Switch to scene: Debug" ,
        ar.setScene("Debug")

def onMenuButton(button):
    if button == "ReadyButton":
        ar.enableModel("pump");
        ar.disableModel("boarded");
    if button == "ReadyButton2":
        ar.enableModel("boarded");
        ar.disableModel("pump");
    print "[Py] Button pressed :", button

```

```

def onPatternAction(pattern):
    print "Action pattern :", pattern

def onFrame(frameid):
    n = 6
    #print "Frame number :", frameid

def onInit(frameid):
    print "INIT Application"
    ar.screenFlipHorizontal()
    # ar.screenFlipHorizontal()
    ar.screenFlipVertical()

# Fires when a mouse event occurs
# model: a string that contains the model id
# model: a string that contains the command, TOUCH, FIRE, RELEASE, UNTOUCH
def onMouseEvent(model, command):
    print "Mouse event ", model, " : ", command

    if command == "TOUCH":
        ar.highlightModelOn(model)

    if command == "UNTOUCH":
        ar.highlightModelOff(model)

```

APPENDIX G – INFORMED CONSENT DOCUMENT

INFORMED CONSENT DOCUMENT

Title of Study: Evaluating User Interface Elements for Augmented Reality Assembly Support

Investigators: James Oliver, Jordan Herrema, Rafael Radkowski

This is a research study. Please take your time in deciding if you would like to participate. Please feel free to ask questions at any time.

INTRODUCTION

We are interested in understanding how instructions for the assembly of mechanical devices can be presented in ways that will improve the speed and accuracy of the assembly process. You should not participate if you have any physical conditions that would prevent you from performing assembly operations with your hands and simple handheld tools, or if you are under 18 years of age.

DESCRIPTION OF PROCEDURES

If you agree to participate, you will be asked to assemble a mechanical device based on instructions given by either a printed manual or augmented reality prompts displayed on a computer monitor. In either case, no additional assistance will be offered by the investigator. Some steps may require the use of simple handheld tools; if so, this will be indicated by the instructions. The investigator will observe you during the assembly process and may take notes or collect relevant data.

Your participation is expected to last less than 30 minutes, including both the assembly process itself and the post-study questionnaire.

RISKS

While participating in this study you may experience the following risks:

If your instructions are displayed on a computer monitor, there is a small probability that you may experience nausea or general discomfort—similar to the risk inherent in video games or 3D movies. If you experience such symptoms, please close your eyes and immediately inform the experimenter. The study will be stopped and you will be allowed to withdraw with no penalties. Also, it is possible that medical conditions could cause seizures due to rapidly changing visual stimuli. If you have any known disorders of this kind, please excuse yourself from the study.

Since you will be sharing the tools and assembly components with other participants, there is a small risk of germ transmission. Therefore the tools and components will be cleaned after each usage and we suggest that you wash your hands after the study. The magnitude of this risk is no higher than normal life activities such as exchanging money or using a public keyboard.

BENEFITS

If you decide to participate in this study there will be no direct benefit to you. However, it is hoped that the information gained in this study will benefit society by increasing the understanding of how to best display instructions using new types of media.

COSTS AND COMPENSATION

You will not have any costs based on your participation in this study. There is also no compensation offered.

PARTICIPANT RIGHTS

Your participation in this study is completely voluntary and you may refuse to participate or leave the study at any time. If you decide to not participate in the study or leave the study early, it will not result in any penalty or loss of benefits to which you are otherwise entitled. You can skip any questions that you do not wish to answer.

CONFIDENTIALITY

Records identifying participants will be kept confidential to the extent permitted by applicable laws and regulations and will not be made publicly available. However, auditing departments of Iowa State University, and the Institutional Review Board (a committee that reviews and approves human subject research studies) may inspect and/or copy your records for quality assurance and data analysis. These records may contain private information.

To ensure confidentiality to the extent permitted by law, the particular results of your participation in this study will not be linked to your personal identity in any way. Also, if the results of this study are published, the identities of all participants will remain confidential.

QUESTIONS OR PROBLEMS

You are encouraged to ask questions at any time during this study.

- For further information about the study contact Dr. James Oliver at oliver@iastate.edu or Jordan Herrema at jrdnhrrm@iastate.edu.
- If you have any questions about the rights of research subjects or research-related injury, please contact the IRB Administrator, (515) 294-4566, IRB@iastate.edu, or Director, (515) 294-3115, Office for Responsible Research, Iowa State University, Ames, Iowa 50011.

PARTICIPANT SIGNATURE

Your signature indicates that you voluntarily agree to participate in this study, that the study has been explained to you, that you have been given the time to read the document, and that your questions have been satisfactorily answered. You will receive a copy of the written informed consent prior to your participation in the study.

Participant's Name (printed) _____

(Participant's Signature)

(Date)

APPENDIX H – STUDY QUESTIONNAIRES

Evaluating User Interface Elements for Augmented Reality Assembly Support

Pre Questionnaire

1. What is your age? _____
2. What is your sex? (circle one) Male Female
3. What is the highest level of education you have **completed**? (circle one)
 High School B.S./B.A. M.S./M.A. Ph.D./M.D. Other: _____
 - a. If you are a student, what is your major? _____
 - b. If you are not a student, what is your profession? _____

Participant _____ Group _____

Date _____

Evaluating User Interface Elements for Augmented Reality Assembly Support

Post Questionnaire

2. Please rank the ease of each of the following components of the assembly task by circling a number between 1 (very difficult) and 5 (very easy).

a. Understanding WHERE to place parts on the assembly:

Very Difficult 1 2 3 4 5 Very Easy

b. Understanding how to ORIENT parts on the assembly:

Very Difficult 1 2 3 4 5 Very Easy

c. Understanding HOW to use tools in the assembly process:

Very Difficult 1 2 3 4 5 Very Easy

d. Understanding WHEN to use tools in the assembly process:

Very Difficult 1 2 3 4 5 Very Easy

e. Recognizing when you had made a mistake (circle "5" **only** if you do not believe you made **any** mistakes at any point during the assembly process)

Very Difficult 1 2 3 4 5 Very Easy

3. Please answer the following questions about the assembly task overall.

a. How helpful were the instructions to the assembly process overall?

Not Helpful 1 2 3 4 5 Very Helpful

b. How easy were the assembly instructions to interact with? (Navigating the steps in the instructions, etc.)

Very Difficult 1 2 3 4 5 Very Easy

c. How confident did you feel about your ability to perform the assembly task when the study began?

Not Confident 1 2 3 4 5 Very Confident

d. How confident would you feel about your ability to perform this task again now that the study is complete?

Not Confident 1 2 3 4 5 Very Confident

e. How confident do you feel about your ability to perform OTHER tasks similar to this now that the study is complete?

Not Confident 1 2 3 4 5 Very Confident

Participant _____ Group _____

Date _____

Please BRIEFLY describe any previous experience you have had with manual assembly tasks similar to the one you performed in this study:

Do you have any additional feedback that you would like to give about the assembly instructions or about the study in general?

Your participation in the study is now complete. Thank you for your involvement!

Participant _____ Group _____

Date _____

