FixIt: An Approach towards Assisting Workers in Diagnosing Machine Malfunctions

Gudrun Klinker, Hesam Najafi, Tobias Sielhorst, Fabian Sturm, Florian Echtler, Mustafa Isik, Wolfgang Wein, and Christian Trübswetter

Technische Universität München

Institut für Informatik

Boltzmannstr. 3

85748 Garching

+49 89 289 18215

{klinker, najafi, sielhorst, sturmf, echtler, isikm, wein, truebswe}@in.tum.de

ABSTRACT

Augmented Reality (AR) is a newly emerging user interface paradigm that is currently under rapid development. AR is still in its infancy. Only very few cases exist, in which AR technology has come out of the laboratories and has been evaluated [2] or used [8] in real industrial settings.

At the Technical University of Munich, we address some of such open questions annually in three-month laboratory classes for graduate students in their junior years. By posing the problems within industrial settings (e.g., by involving an industrial sponsor) we try to ensure a realistic setting within which AR solutions are considered. Students investigate the posed problem, suggest a solution and build a very rough, prototypical demonstrator that illustrates the key ideas.

The result is a prototypical system, illustrating the key points of using AR in a specific problem context. At the same time, the system design process has also laid out (and partially addressed) both system building and very general system design issues, which will be led back onto research agenda of the university and picked up in due time.

Introduction

Augmented Reality (AR) is a newly emerging user interface paradigm that is currently under rapid development. Coined by Caudell and Mizell [1, 2] in the early nineties, the term “Augmented Reality” describes the concept of providing users with three-dimensional virtual information embedded within their real environment.[4]. Within only one decade, the concept has gained significant importance, resulting in an annual international symposium alternating between three continents [5] and major industry-supported projects [6,7].

Yet, AR is still in its infancy. Only very few cases exist, in which AR technology has come out of the laboratories and has been evaluated [3] or used [8] in real industrial settings. Reasons for this are manifold. On the one hand, AR involves many very hard research problems, such as real-time tracking at very high precision and robustness, that haven’t been solved yet. Furthermore, AR has to come out of the lab and face software engineering issues such as reusability, extensibility and scalability from prototypical demonstrators to deployable systems.

In particular, the adaptation of local tracking systems that are currently confined to the corner of a lab have to be generalized toward being usable in a real world where many objects and people are moving about in unpredictable ways. In this respect, the layout of a partially wearable, partially ubiquitous system infrastructure that supports generating augmented views to individual users has to be defined.

Furthermore, it is still unclear how virtual information should be structured and automatically generated to be easily available for real-time inclusion in an AR system. Thus, authoring tools for AR systems need to be developed. Such authoring also involves live visualization of real-time control data to diagnose and repair malfunctioning machines.

At the Technical University of Munich, we address some of such open questions anually in three-month laboratory classes for graduate students in their junior years. By posing the problems within industrial settings (e.g., by involving an industrial sponsor) we try to ensure a realistic setting within which AR solutions are considered. Students investigate the posed problem, suggest a solution and build a very rough, prototypical demonstrator that illustrates the key ideas.

The purpose of such courses is threefold:

- For the students: In addition to attending theoretical classes on Augmented Reality, students here obtain hands-on experience regarding the difficulties and subtleties of building a real AR application. By following a rigorous software engineering approach towards designing and building the system [9], students get actively involved in determining the customer's requirements for the application, laying out a generally suitable system architecture, seeking currently available solutions for required components from industry or research labs, and identifying and prioritizing open issues.

- For the customer (when available): Industrial sponsors obtain a feasibility study, analyzing options for using AR technology in their envisioned scenario. Due to the involvement of a rather large group of students, the feasibility study tends to be rather creative and visionary, yet realistic.
- For the AR research group: By thoroughly exploring the requirements posed by a particular application domain with a customer, we are able to identify research issues in a wider context than our laboratory setting – with the intention to classify them and further pursue them at a more abstract level across several applications.

  Furthermore, each course builds upon our growing software base, embedded within the Distributed Wearable Augmented Reality Framework (DWARF) [10]. Thus, we are able to test and extend this base with every student project.

Within such courses, we have been able to explore a number of problems in recent years: The authoring and augmented use of interactive electronic technical manuals (IETMS) for nuclear power plant maintenance (STARS project) [11], the AR-based visualization and evaluation of automotive designs (Fatamorgana) [12], and, most recently, the live diagnosis of malfunctions of machines (FixIt). Using a similar approach with senior graduate students after they had attended the course, we have been able to set up an AR-assisted welding system for automotive use which is now being used continuously in the early technical integration phases for new cars [8].

The FixIt Problem

Every once in a while, machinery in industrial settings does not function as intended. Reasons for this are manifold: a cable controlling one of the motors or sensors might be loose, thereby not conveying the correct control or status information between the machine an the control system. The machine might be hindered by an unforeseen physical obstacle. Or the control program might be in an unplanned state. One of the main problems in diagnosing the problem is to establish a mapping between the internal control state of the computer and the actual physical state of the machine. To this end, the maintenance person needs to have a clear understanding of both, as well as their expected/actual dynamic changes over time.

By overlaying virtual information of the control system directly onto the machine while it is in operation, Augmented Reality has the potential to help workers obtain a better understanding of the reasons for malfunctions. The result is an intricate new, highly immersive net of interactions and relationships in a man-computer-machine triangle.

System Development Process

Arranged as one of our collaborative, team-oriented student lab courses, we have built a first prototype of the FixIt system. It assists workers in diagnosing malfunctions of complex machines, such as robots, by superimposing the current control status of the machine onto the currently active machine parts – thereby indicating which components are expected to be active at what time. Discrepancies between the expected behavior and the real robot behavior help workers understand the causes of malfunctions.

Students were asked to first set up an overall system design and then to form smaller teams to each address the issues involved in building one of the components.

As a first prototype system, a toy robot by Fischer Technik [13] was built and connected to a computer running the FixIt demonstration. The current control state was overlaid on the robot by highlighting motors and sensors while they were active.

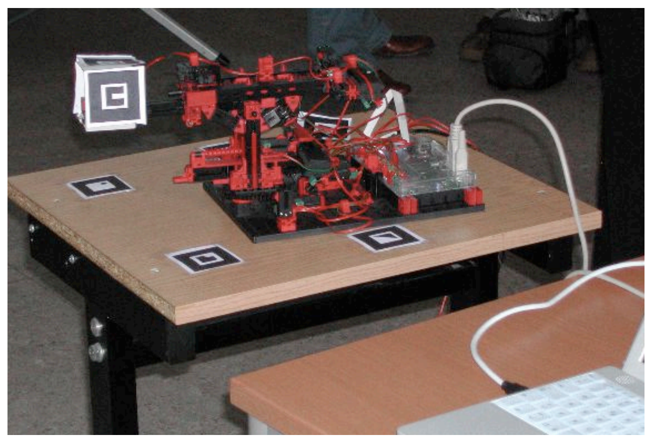


Figure 1: Real Robot with Computer

System Design

The FixIt system was designed to consist of three major components:

- A robot welding application controlling the robot. It sends commands to the robot to move its arm forward and backward, rotate around its base, rotate the arm up and down, and perform a welding action. The same commands are forwarded to the visualization component.
- A tracking component, determining the current physical state of the robot, as well as the current viewing position of a camera representing the

maintenance person. All position data is forwarded to the visualization component.

- A visualization component, overlaying the control state information onto the robot, according to the camera's current point of view.

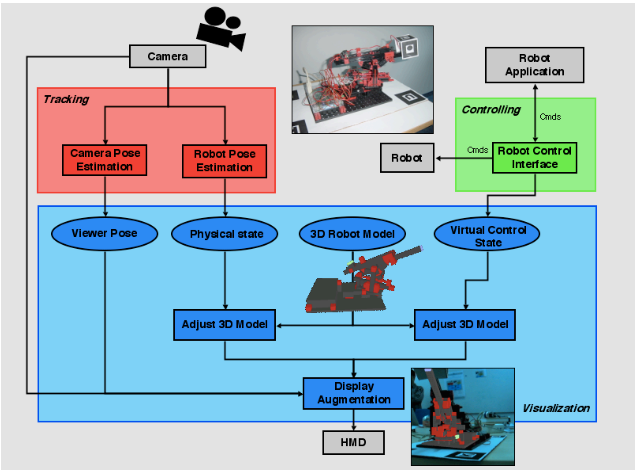


Figure 2: Schematic

Research Issues

The system requires a number of novel components and concepts that go beyond the currently typical augmented presentation of static graphical visualizations within a real environment.

- The FixIt system requires that relevant information be derived from the current system context – in particular from the current control state of the machine. To this end, control commands issued to the robot need to be accumulated and transferred into meaningful state information. Furthermore, the dynamics and dependencies of the control program, such as conditional executions depending on sensor input, looping constructs waiting for an exit condition to become true, and parallel or alternative execution, need to be formulated as program context primitives and forwarded to the visualization component.
- Since both the worker and the robot move, tracking is required both for determining the current worker position and for determining the current physical shape of a machine such that augmentations are actually placed correctly onto individual machine parts while these are moving.

  As a consequence, many assumptions built into current tracking algorithms break down. In the case of a marker-based system, markers have to be divided in different groups according to their respective mobility. Going beyond marker-based tracking towards marker-less optical tracking, this means that complex mappings have to be established between image changes due to user motion and image changes due to robot motions.
- Suitable visualization schemes are required which transform control state information into minimal augmentations – just enough to help workers understand which machine parts are involved without covering the machine up more than necessary. At the same time, visualizations for conveying the program context need to be developed, indicating to workers the reasons for malfunctions, e.g. that a motor doesn't stop because it is waiting for a tactile sensor to return a signal – which it doesn't return due to a malfunction.

First Solutions

Within the course of a three-month practical class, students have been able to implement only a rudimentary, first prototype of an augmented robot maintenance system.

Using the Distributed Wearable Augmented Reality Framework (DWARF) [10] that is available in the AR research lab, the students were able to develop first versions of the three individual system components and combine them into a peer-to-peer-based system prototype to demonstrate the overall concepts. Future work will involve follow-up classes and student diploma theses to improve upon the individual components or add new components addressing the raised issues at greater depth.

The Control Component

The FixIt control component focuses thus far on accumulating the current system state from individual robot control commands. It does not yet send program context information to the visualization component.

According to the physically possible motions and sensory evaluations of the robot at hand, the component currently focuses on sending the following, very specific status information to the visualization component:

- Motor state for each of three existing motors: {stopped, moving right, moving left}
- Status of the welding tool (a light at the tip of the robot arm): {off, on}
- Status of 5 switches, associated with tactile sensors on the robot: 2 stop-switches that can be on or off, and three click switches.

For testing the robot and for analyzing malfunctions, workers can request that the tip of the robot move in any physically reachable three-dimensional position. The control program internally divides the request into a sequence of control commands steering the individual motors.

A general authoring question with respect to generalizing this very specific communication scheme for this particular robot is raising its head here: what would be a general syntax for an arbitrary robot control program to communicate with a standardized visualization component? How many, and what kinds of primitives are required?

The Tracking Component

Tracking is required both for determining the current worker position and for determining the current physical shape of the machine. For their first implementation, the students used the AR-Toolkit [14] because it was easily available and provided them with a jump start.

The scene was augmented with two sets of markers:

- Scene-stabilized "base" markers that would not move. These markers formed the basis for determining the current viewer position.
- Object-stabilized "robot" markers at the tip of the robot arm (at the welding tool). These markers were used to determine the current robot pose. In a sense, these markers were redundant, since the object pose could also be determined from the sequence of control commands issued by the control program. The discrepancy between the virtual, computable position of the robot tip and its actually measured position due to marker locations is one of the cues that students expected to be critical to help workers diagnose malfunctions.

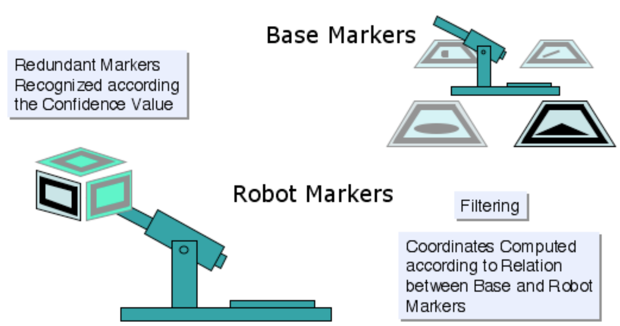


Figure 3: Marker Placement

The Visualization Component

Suitable visualization schemes are required to transform control state information into scene augmentations.

As a first requirement, the system needs to have an internal model of the real robot, animated to account for any potential robot movement. To this end, a VRML model of the Fischer Technik robot was built (see Figure 4). The model was composed from a library of basic models describing individual Fischer Technik parts.

In addition to visualizing the entire robot model, the visualization component is also able to

- overlay the model transparently onto a picture of the real robot (assisting the tracking state),
- light up active robot parts according to recently issued robot control commands (assisting the malfunction diagnosis state).

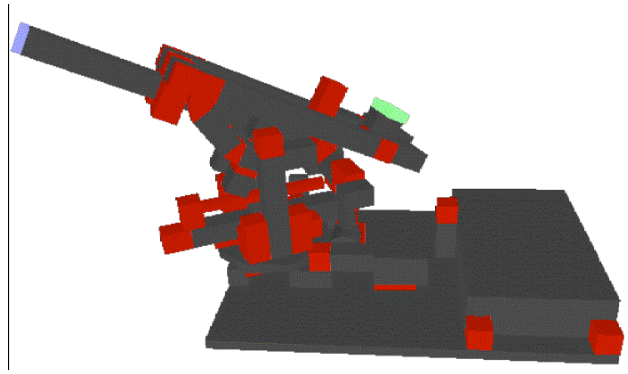


Figure 4: Virtual Robot

The visualization component receives input both from the robot control component and from the tracking component. Currently, it uses the robot control input to steer the animation of the virtual model (assuming that every issued command of the steering system has actually been executed correctly by the robot). It uses the tracking input to determine the current worker position and to render the augmentations accordingly. Thus, the tool does not yet exploit the redundancy between issued control commands and visually tracked robot positions to automatically identify malfunctions.

The stream of robot control commands also determines which parts of the virtual robot model are rendered transparently (i.e., are invisible to the worker), and which parts are actually rendered opaquely – and are thus pointed out to the worker.

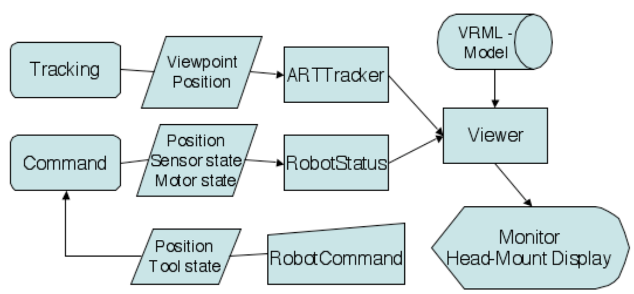


Figure 5: Information Flow

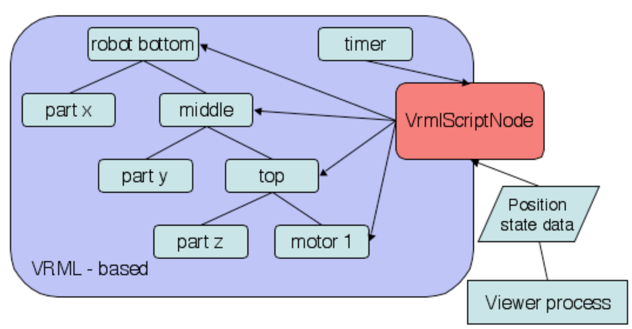


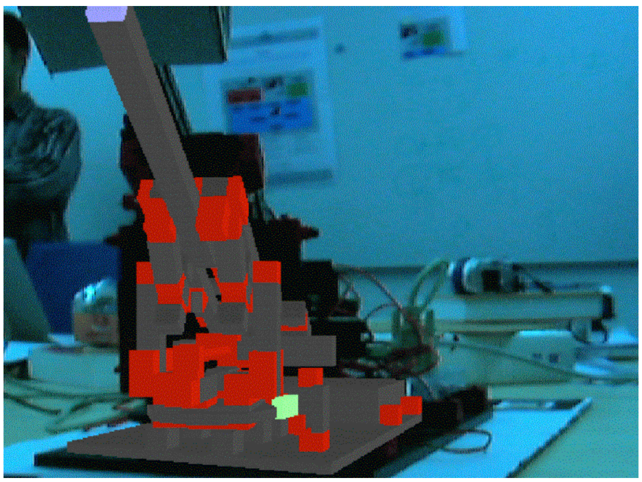
Figure 6: Model Updates

Results: A Demonstration System

Using the Distributed Wearable Augmented Reality Framework (DWARF) [10], students were able develop and build a first demonstration system within one semester to overlay the control state of a Fischer Technik robot onto its actual physical incarnation.

The system uses a camera to view the robot scene. Since the scene is carefully set up to contain a number of markers on the base plate, the camera can move freely within the scene: camera pictures are continually updated to superimpose the current robot view onto the video image.

According to recently issued robot control commands, currently active units of the robot (e.g. some motors or tactile sensors) are highlighted to indicate their active status to the worker. Such highlighting works well even when the robot moves, since the system tracks mobile robot parts independently of the scene-stabilized markers on the base plate.



Discussion

The augmentation of a robot in action provides a lot of potential. This is often alluded to in machine maintenance and repair scenarios.

Yet, the actual consequences of implementing a system within which physical components really do move according to a control program and thus need to be followed, have not yet been explored much.

Especially in the case of diagnosing malfunctions, there is an interesting discrepancy between the internal (virtual) state of a machine and its real (physical) state. If due care is being taken to formalize this discrepancy and to determine it from physical measurements (tracking data) and control status information, it can form the basis for very powerful diagnosis aids for machine repair personnel.

Yet, the discrepancy alone will not suffice. It needs to be supplemented with program context data which indicates what the robot control program is trying to achieve while issuing a certain sequence of robot control commands.

The formalization of such program context, as well as the standardized indication of a program control state (for an arbitrary machine) and the visualization of discrepancies between the virtual and real robot state are important visualization issues that need to be addressed in order to generate authoring systems that will be suitable as augmented debugging aides for larger sets of robots.

Within the current three-month student project, we have been able to lay the ground work towards exploring these and other exciting issues related to an online diagnosis of malfunctions of machines while they are in operation.

Acknowledgments

Many thanks go to all students who attended this class and contributed to the design and implementation of the demonstrator. We are also grateful to Asa MacWilliams, Martin Wagner, Martin Bauer, Christian Sandor and Thomas Reicher for making the DWARF system available to the class and for continually supporting the class in using it. The entire Lehrstuhl for Applied Software Engineering (Prof. Brügge) was involved in supporting the successful execution of the lab project, providing us with needed utilities and advice – as well as the Fischer Technik robot.

References

1. T.Caudell and D. Mizell. Augmented Reality: An Application of Heads-up Display Technology to Manual Manufacturing Processes. *Proc. Hawaiian International Conference on System Sciences (HICSS'92)*, pp. 659-669.
2. A. Janin, D. Mizell, and T. Caudell. Calibration of head-mounted displays for augmented reality applications. *Proc. of the Virtual Reality Annual International Symposium (VRAIS'93)*, 1993, pp. 246-255.
3. D. Curtis, D. Mizell, P. Gruenbaum, and A. Janin. Several Devils in the Details: Making an AR App Work in the Airplane Factory. *Proc. 1st International Workshop on Augmented Reality (IWAR'98)*, San Francisco, Nov. 1998, pp. 47-60.
4. R.T. Azuma. A survey of augmented reality. *Presence, Special Issue on Augmented Reality*, 6(4):355-385, August 1997.
5. International Workshop/Symposium on (Mixed)Augmented Reality (IWAR, ISAR, ISMAR), 1998-2003.
6. Friedrichs W (2002) ARVIKA Augmented Reality for Development, Production, and Service. In: Proc Int Symp on Mixed and Augmented Reality (ISMAR'02), Darmstadt, Germany, IEEE. See also: http://www.arvika.de/www/pdf/IVIP_Forume.pdf.

7. Tamura H (2000) What happens at the border between real and virtual worlds – the MR project and other research activities in Japan. In: Proc Int Symp on Augmented Reality (ISAR'00), Munich, IEEE.
8. F. Echtler, F. Sturm, K. Kindermann, G. Klinker, J. Stillla, J. Trilk, and H. Najafi. (2003). The Intelligent Welding Gun: Augmented Reality for Experimental Vehicle Construction. In: Virtual and Augmented Reality Applications in Manufacturing. S.K. Ong and A.Y.C. Nee, eds. Springer Verlag 2004.
9. B. Bruegge and A. Dutoit. *Model-based Software Engineering: A Project-Oriented Approach*, Prentice Hall 1999.
10. M. Bauer, B. Brügge, G. Klinker, A. MacWilliams, T. Reicher, S. Riß, C. Sandor, and M. Wagner. Design of a Component-Based Augmented Reality Framework. ISAR '01.
11. G. Klinker, O. Creighton, A. Dutoit, R. Kobylinski, C. Vilsmeier and B. Brügge: Augmented maintenance of powerplants: A prototyping case study of a mobile AR system. Proc. IEEE International Symposium on Augmented Reality (ISAR'01), New York, 2001.
12. G. Klinker, A. Dutoit, M. Bauer, J. Bayer, V. Novak and D. Matzke, "Fata Morgana -- A Presentation System for Product Design", Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'02), Darmstadt, 2002.
13. Fischer Technik. http://www.fischertechnik.de/
14. H. Kato, M. Billinghurst, I. Puopyrev, K. Imamoto and K. Tachibana. Virtual Object Manipulation on a Table-Top AR Environment. *Proc. 1rst International Symposium on Augmented Reality (ISAR'00)*, Munich, Oct. 2000, pp. 111-119.