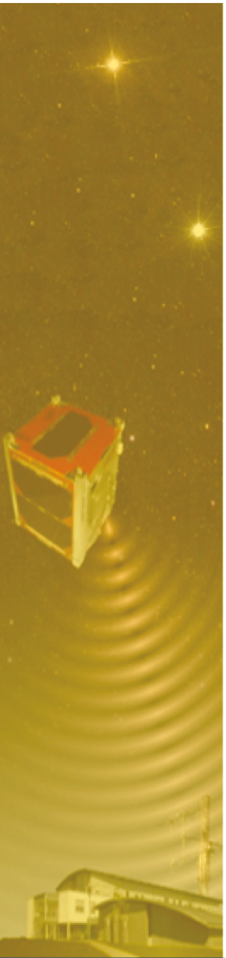


Institut für Informatik
Lehrstuhl für Robotik und Telematik
Prof. Dr. K. Schilling

Würzburger Forschungsberichte in Robotik und Telematik

Uni Wuerzburg Research Notes in Robotics and Telematics

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Band 13

Doris Aschenbrenner

Human Robot Interaction Concepts for Human Supervisory Control and Telemaintenance Applications in an Industry 4.0 Environment

Die Schriftenreihe

wird vom Lehrstuhl für Informatik VII: Robotik und Telematik der Universität Würzburg herausgegeben und präsentiert innovative Forschung aus den Bereichen der Robotik und der Telematik.

Die Kombination fortgeschrittener Informationsverarbeitungsmethoden mit Verfahren der Regelungstechnik eröffnet hier interessante Forschungs- und Anwendungsperspektiven. Es werden dabei folgende interdisziplinäre Aufgabenschwerpunkte bearbeitet:

Robotik und Mechatronik: Kombination von Informatik, Elektronik, Mechanik, Sensorik, Regelungs- und Steuerungstechnik, um Roboter adaptiv und flexibel ihrer Arbeitsumgebung anzupassen.

Telematik: Integration von TelekommunikationInformatik und Steuerungstechnik, um Dienstleistungen an entfernten Standorten zu erbringen.

Anwendungsschwerpunkte sind u.a. mobile Roboter, Tele-Robotik, Raumfahrtsysteme und Medizin-Robotik.

Lehrstuhl Informatik VII
Robotik und Telematik
Am Hubland
D-97074 Wuerzburg

Tel.: +49 (0) 931 - 31 - 86678
Fax: +49 (0) 931 - 31 - 86679

schi@informatik.uni-wuerzburg.de
http://www7.informatik.uni-wuerzburg.de

Dieses Dokument wird bereitgestellt durch den Online-Publikationsservice der Universität Würzburg.

Universitätsbibliothek Würzburg
Am Hubland
D-97074 Würzburg

Tel.: +49 (0) 931 - 31 - 85906

opus@bibliothek.uni-wuerzburg.de
http://opus.bibliothek.uni-wuerzburg.de

ISSN (Internet): 1868-7474
ISSN (Print): 1868-7466
eISBN: 978-3-945459-18-8

Zitation dieser Publikation

ASCHENBRENNER, D. (2017). Human Robot Interaction Concepts for Human Supervisory Control and Telemaintenance Applications in an Industry 4.0 Environment. Schriftenreihe Würzburger Forschungsberichte in Robotik und Telematik, Band 13. Würzburg: Universität Würzburg.
urn:nbn:de:bvb:20-opus-150520

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Human Robot Interaction Concepts for Human Supervisory Control and Telemaintenance Applications in an Industry 4.0 Environment

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Julius-Maximilians-Universität Würzburg

vorgelegt von

Doris Aschenbrenner

aus

Coburg

Würzburg 2017

Eingereicht am: 06.09.2016

bei der Fakultät für Mathematik und Informatik

1. Gutachter: Prof. Dr. Klaus Schilling

2. Gutachter: Prof. Dr. Marc Latoschik

Danksagung

Die vorliegende Dissertation wäre nicht möglich gewesen ohne die tatkräftige Unterstützung von vielen Menschen, denen ich im Folgenden noch einmal herzlich danken möchte.

Zuerst bedanke ich mich bei meinem Doktorvater Prof. Dr. Klaus Schilling für die Möglichkeit eine Promotion unter seiner Betreuung durchzuführen, aber auch für die vielen spannenden Dinge, die ich die letzten Jahre erleben durfte, und die Erfahrungen, die ich dabei sammeln konnte. Er ermöglichte mir eine berufliche und persönliche Weiterentwicklung über die Promotion hinaus.

Prof. Dr. Latoschik einen herzlichen Dank für seine unkomplizierte Art und die zügige Erstellung des Zweitgutachtens, sowie die tiefgehend fachliche Beratung in Sachen Mensch-Computer Interaktion. Ein weiterer Dank gilt meinen beiden Prüfern Prof. Dr. Andreas Nüchter und Dr. Christian Herrmann. Beide habe ich schon während meines Studiums, aber auch jetzt während meiner Promotionszeit sowohl menschlich als auch fachlich sehr schätzen gelernt.

Ebenso möchte ich mich ganz herzlich bei meinem Mentor über das Mentoringprogramm der Universität Würzburg, Prof. Dr. Wilfried Kunde und meiner kompetenten Ansprechpartnerin außerhalb der Universität Würzburg, Prof. Dr. Ute Schmid aus Bamberg bedanken.

Durch die Arbeit während der Promotion am Lehrstuhl für Robotik und Telematik der Universität Würzburg und dem Zentrum für Telematik hatte ich die Gelegenheit mit einem außergewöhnlichen Team zusammenzuarbeiten, was sich auch in privaten gemeinsamen Unternehmungen außerhalb der Arbeit widerspiegelte.

Danke hierfür im Besonderen an Felix Sittner, Michael Bleier, Michael Albert, Kai Schwedhelm und Florian Leutert, sowie Ulrike Göbel, Heidi Frankenberger und Dieter Ziegler.

In den letzten Jahren durfte ich im Rahmen dieser Arbeit auch mit einer Vielzahl von Studierenden zusammenarbeiten. Stellvertretend für alle diese Studenten möchte ich mich insbesondere bei meinen studentischen Hilfskräften Manuel Seitz, Daniel Kneip, Paul Barth und Nicolas Maltry bedanken. Weiterhin ein Danke an Johannes Kimmel und Michael Rojkov, die genau wie ein Großteil meiner studentischen Hilfskräfte ihre Abschlussarbeiten bei mir absolviert haben.

Vielen Dank an meine Eltern Peter und Karin und meinen Brüdern Franz und Claus für die uneingeschränkte Unterstützung und den Rückhalt in all den Jahren. Das Bewusstsein über diese Unterstützung und diesen Rückhalt gab mir die Sicherheit, die in vielen Situation für diese Arbeit wichtig war.

Zu guter Letzt möchte ich mich bei dem wichtigsten Menschen in meinem Leben bedanken - Julian Scharnagl. Du bist immer für mich da. Danke für alles!

Abstract

While teleoperation of technical highly sophisticated systems has already been a wide field of research, especially for space and robotics applications, the automation industry has not yet benefited from its results. Besides the established fields of application, also production lines with industrial robots and the surrounding plant components are in need of being remotely accessible. This is especially critical for maintenance or if an unexpected problem cannot be solved by the local specialists.

Special machine manufacturers, especially robotics companies, sell their technology worldwide. Some factories, for example in emerging economies, lack qualified personnel for repair and maintenance tasks. When a severe failure occurs, an expert of the manufacturer needs to fly there, which leads to long down times of the machine or even the whole production line. With the development of data networks, a huge part of those travels can be omitted, if appropriate teleoperation equipment is provided.

This thesis describes the development of a telemaintenance system, which was established in an active production line for research purposes. The customer production site of Braun in Marktheidenfeld, a factory which belongs to Procter & Gamble, consists of a six-axis cartesian industrial robot by KUKA Industries, a two-component injection molding system and an assembly unit. The plant produces plastic parts for electric toothbrushes.

In the research projects "MainTelRob" and "Bayern.digital", during which this plant was utilised, the Zentrum für Telematik e.V. (ZfT) and its project partners develop novel technical approaches and procedures for modern telemaintenance. The term "telemaintenance" hereby refers to the integration of computer science and communication technologies into the maintenance strategy. It is particularly interesting for high-grade capital-intensive goods like industrial robots. Typical telemaintenance tasks are for example the analysis of a robot failure or difficult repair operations. The service department of KUKA Industries is responsible for the worldwide distributed customers who own more than one robot. Currently such tasks are offered via phone support and service staff which travels abroad. They want to expand their service activities on telemaintenance and struggle with the high demands of teleoperation especially regarding security infrastructure. In addition, the facility in Marktheidenfeld has to keep up with the high international standards of Procter & Gamble and wants to minimize machine downtimes. Like 71.6% of all German companies, P&G sees a huge potential for early information on their production system, but complains about the insufficient quality and the lack of currentness of data.

The main research focus of this work lies on the human machine interface for all

human tasks in a telemaintenance setup. This thesis provides own work in the use of a mobile device in context of maintenance, describes new tools on asynchronous remote analysis and puts all parts together in an integrated telemaintenance infrastructure. With the help of Augmented Reality, the user performance and satisfaction could be raised. A special regard is put upon the situation awareness of the remote expert realized by different camera viewpoints. In detail the work consists of:

- Support of maintenance tasks with a mobile device
- Development and evaluation of a context-aware inspection tool
- Comparison of a new touch-based mobile robot programming device to the former teach pendant
- Study on Augmented Reality support for repair tasks with a mobile device
- Condition monitoring for a specific plant with industrial robot
- Human computer interaction for remote analysis of a single plant cycle
- A big data analysis tool for a multitude of cycles and similar plants
- 3D process visualization for a specific plant cycle with additional virtual information
- Network architecture in hardware, software and network infrastructure
- Mobile device computer supported collaborative work for telemaintenance
- Motor exchange telemaintenance example in running production environment
- Augmented reality supported remote plant visualization for better situation awareness

Thanks to

Awesome Technologies

Contents

1 Introduction **1**
1.1 Overview Industry 4.0 . . . 1
1.2 Motivation . . . 3
1.3 Topic description . . . 4
1.4 Summary of contributions . . . 5
1.5 Thesis outline . . . 6

2 Fundamentals **7**
2.1 Overview of the field . . . 7
2.1.1 Telematic communication in an industrial internet setting . . . 7
2.1.2 Definitions . . . 8
2.2 M2M communication . . . 12
2.2.1 Robot telematics . . . 13
2.2.2 Fault detection and diagnosis . . . 18
2.2.3 Big data and data mining . . . 20
2.3 HCI - Human Computer Interaction . . . 25
2.3.1 HCI for industry . . . 25
2.3.2 Mental workload . . . 27
2.3.3 Mental model . . . 27
2.3.4 Conceptual model . . . 28
2.3.5 Measuring usability and user experience . . . 29
2.3.6 The user experience process . . . 31
2.4 H2H communication . . . 33
2.4.1 Computer supported collaborative work . . . 33
2.4.2 Communication . . . 34
2.4.3 Common ground and grounding . . . 34
2.5 M2H communication . . . 36
2.5.1 Human perception and reaction . . . 36
2.5.2 Situation Awareness . . . 38
2.5.3 Spatial notation . . . 39
2.5.4 Visuospatial thinking . . . 40
2.5.5 Information space . . . 43
2.5.6 Augmented and Virtual Reality . . . 47
2.6 H2M communication . . . 54
2.6.1 Human in the maintenance process . . . 55
2.6.2 Human control of robots . . . 55
2.6.3 Extended Human Supervisory Control . . . 57
2.6.4 Summary . . . 58

3 Project MainTelRob: Initial situation and statement of the problem 63
3.1 Project setting . . . 63
3.1.1 Motivation and project goals . . . 63
3.1.2 Project concept . . . 65
3.2 Work flow . . . 66
3.2.1 Current state . . . 68
3.2.2 Preliminary interviews and telemaintenance . . . 68
3.2.3 Specifications . . . 69
3.2.4 Documentation of the use cases . . . 69
3.2.5 Infrastructure setup . . . 69
3.2.6 Fields of study . . . 70
3.2.7 Design informing models . . . 70
3.2.8 Design process . . . 70
3.2.9 Functional prototypes . . . 71
3.2.10 Pluralistic walkthrough . . . 71
3.2.11 Final prototypes and evaluation . . . 71
3.3 General results . . . 72
3.3.1 Contextual analysis: the plant . . . 72
3.3.2 Design informing models . . . 73
3.3.3 Scenarios . . . 77
3.4 Survey . . . 81
3.4.1 Mobile device for local maintenance . . . 83
3.4.2 Intuitive visualization of complex data . . . 84
3.4.3 Telemaintenance as human supervisory control . . . 84

4 Mobile device for local maintenance 85
4.1 Survey . . . 85
4.1.1 Motivation . . . 86
4.1.2 Goals . . . 87
4.1.3 Related Work . . . 88
4.2 Maintenance with a mobile device . . . 89
4.2.1 Contextual inquiry and analysis . . . 89
4.2.2 Maintenance checklist . . . 93
4.3 Inspection: Visual control . . . 94
4.3.1 Contextual analysis . . . 94
4.3.2 Prototypes . . . 95
4.3.3 Evaluation . . . 99
4.4 A comparison of two hand-held devices for industrial robotic control . 104
4.4.1 Methods . . . 105
4.4.2 Results . . . 108
4.4.3 Suggestions for improvement . . . 109
4.4.4 Self-criticism and methodical review . . . 110
4.4.5 Discussion . . . 111

4.5 Repair with Augmented Reality . . . 112
4.5.1 Contextual analysis . . . 112
4.5.2 Failure diagnosis . . . 112
4.5.3 Instruction prototypes with Augmented Reality support . . . 115
4.5.4 Evaluation . . . 127
4.5.5 Discussion . . . 136
4.6 Summary . . . 137
4.6.1 Maintenance . . . 137
4.6.2 Visual Control . . . 138
4.6.3 User study PHG/reisPAD . . . 138
4.6.4 Repair study Eyestruction . . . 139
4.6.5 Economical benefits . . . 139
4.6.6 Conclusion . . . 140

5 Asynchronous remote teleanalysis 141
5.1 Survey . . . 141
5.1.1 Motivation . . . 142
5.1.2 Goals . . . 143
5.1.3 Related Work . . . 145
5.2 Condition Monitoring . . . 145
5.2.1 Concept . . . 145
5.2.2 Implementation . . . 147
5.3 Machinery optimizer . . . 149
5.3.1 Contextual analysis . . . 149
5.3.2 Prototype . . . 151
5.3.3 Evaluation . . . 153
5.4 Complex analysis tool (CAT) . . . 158
5.4.1 Concept . . . 158
5.4.2 Data sources . . . 161
5.4.3 Application scenarios . . . 167
5.4.4 CAT architecture . . . 174
5.4.5 Data mining results . . . 179
5.5 Process visualization (ProcVis) . . . 197
5.5.1 Concept . . . 197
5.5.2 Implementation . . . 198
5.5.3 Results . . . 202
5.6 Summary . . . 204
5.6.1 Condition Monitoring . . . 204
5.6.2 Machinery optimizer . . . 205
5.6.3 Complex analysis tool (CAT) . . . 206
5.6.4 Process visualization (ProcVis) . . . 206
5.6.5 Economical benefits . . . 206
5.6.6 Conclusion . . . 207

6 Telemaintenance as Human Supervisory Control 209
6.1 Survey . . . 209
6.1.1 Motivation . . . 209
6.1.2 Goals . . . 210
6.1.3 Related work . . . 211
6.2 Telemaintenance architecture . . . 212
6.2.1 Contextual analysis . . . 212
6.2.2 Infrastructure . . . 214
6.2.3 Timing . . . 222
6.2.4 Adaptive Management and Security System (AMS) . . . 226
6.3 Telemaintenance interfaces and video interaction . . . 234
6.3.1 Contextual analysis and InteractionLayer . . . 234
6.3.2 Implementation and ApplicationLayer . . . 236
6.3.3 Video interaction . . . 238
6.4 Proof of concept: Motor exchange . . . 252
6.4.1 Procedure . . . 253
6.4.2 Evaluation . . . 254
6.4.3 Summary . . . 261
6.5 ARTab . . . 262
6.5.1 ARTab Concept . . . 262
6.5.2 Architecture . . . 263
6.5.3 Frame calculations . . . 264
6.5.4 User study . . . 268
6.6 Summary . . . 289
6.6.1 Telemaintenance Architecture . . . 290
6.6.2 Interfaces and Video Interaction . . . 290
6.6.3 Proof of concept . . . 291
6.6.4 AR Tab . . . 291
6.6.5 Economical benefits . . . 292
6.6.6 Conclusion . . . 292

7 Conclusion 295
7.1 Summary . . . 295
7.2 Innovation overview . . . 297
7.3 Economical benefits . . . 298
7.4 Suggestions for future work and further applications . . . 300

List of Own Publications 303

List of Figures 305

List of Tables 313

References 315

Chapter 1

Introduction

1.1 Overview Industry 4.0

Digitization changes the game of communication. While this applies for everyday life as much as for industry, the adaption speed is different. Industrial automation, on the one hand, uses high investment goods, like expensive plants with industrial robots with a lifespan of over 15 years. On the other hand, the information and communication technology (ICT) has evolved rapidly in the last years and brought forward a broad range of trends in nearly everyone's private life. The aim of Germany's so called "Industry 4.0" process is to bridge this gap.

This so-called "4th Industrial Revolution" is simplified in Figure 1.1: It aims to integrate the state of the art in ICT into Automation industry and to synchronize the innovation cycles in both areas. The recent ICT trends, like cloud based internet applications or mobile devices, can provide a benefit for production industry. But those approaches need to be adjusted in order to meet the high requirements in the production industry, especially regarding safety, security, availability and robustness.

But why is it called "Industry 4.0"? According to the definition given by the German government [1], the "classical" industrialization at the end of the 18th century counts as the first industrial revolution, which has been caused by the innovation of steam powered engines. Counting is continued by means of other great inventions and the accompanied innovation pushes. So mass production and electricity at the beginning of the 20th century is labeled the 2nd industrial revolution. As IT and electronics have already led to large changes in production industry in the 70s of the 20th century, this has been counted as the 3rd industrial revolution. The number 4 is addressed to the current innovation push, which is caused by internet technologies and cyber physical systems – a term described in the next paragraph.

The international perspective does not distinguish yet between those different industrial revolutions. But there is an ongoing discussion about "Internet of Things", "Industrial Internet" and "Cyber Physical Systems", which are closely related to the German discussion. As there are many different definitions in a field with a high influence of marketing buzzwords, a brief summary of the understanding of those terms in this publication is included. The term "Internet of Things" refers to "the networked interconnection of everyday objects, which are often equipped with ubiquitous intelligence." ([2] p. 1101) and is used in several different application fields. Applied on the Automation industry, the "Industrial internet" is even described as

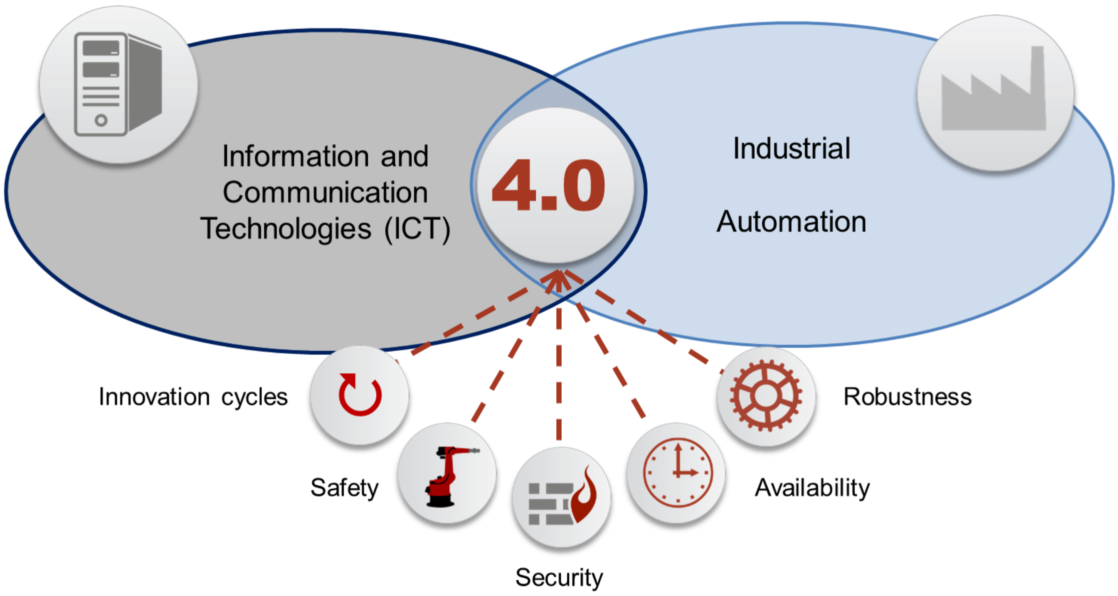


Figure 1.1: *Schema for industry 4.0*

"Industrial Internet of Things", but also by the promise "to bring the key characteristics of the Web – modularity, abstraction, software above the level of a single device – to demanding physical settings [...]." ([3] p. 1). Finally, the term Cyber Physical Systems (CPS) "usually comprise a network of physically distributed embedded sensors and actuators equipped with computing and communicating capabilities." ([4] p. 1). This new line of thought wants to understand software, hardware and communication technologies rather as unity [5] than as different research areas. The term "Cyber Physical System" is used for process modules, whole plants, but also for single individual intelligent products and is referred to widely in the actual German discussion.

The general requirements of the "Leitbild Industrie 4.0" of the German government [1] describe the final state of a factory with a holistic Industry 4.0 concept with a focus on individualized production. The goal is, to reach "batch size 1" while every product steers its production on its own in modularized plants. This should even include the whole supply chain: If a customer selects specific features of a consumer mass product, the production starts to reach exactly this desired configuration. This is possible with "flexible supply chain networks" and a "digital product memory" inside the product [1].

Of course, there are branches operating already very closely at this goal description, for example the automotive industry. But beyond theoretic scenarios visualized in exhibition stands, this might only be the long term perspective, because the necessary investments are not affordable for small and medium sized enterprises (SME).

But there is no "go big or go home". Individualized production does not make sense for every production branch, but they can still profit from the Industry 4.0 process.

According to Prof. Zühlke [6], there are a lot of "lighthouse projects", but too few everyday "streetlights" showing the first steps of Industry 4.0 for small and medium sized enterprises. Instead of talking too much about individualized production, he identifies three major areas of applied research for Industry 4.0: networking, smart objects and human machine interfaces. Those three fields will lead the research presented in this publication.

Zühlke [6] emphasizes, that working internet-based services in automation are the precondition for Industry 4.0. He also sees a huge factor in "smart objects" by which he means a broader definition of Cyber Physical Systems. But a central requirement of Industry 4.0 can be derived from the member survey of the Germany's Association for Automation (GMA) in 2012 [7]: On the question "Which technical or socioeconomic developments will give the biggest momentum in the next three years?" (translated) the "human machine communication" has been judged as one of the most important factors for future productivity. Despite ongoing automation, humans will still be present and also highly necessary in tomorrow's production factories: 96.9% of the questioned enterprises judged the importance of human work in five years still as "important and very important" [8].

Alltogether, this motivates a closer look into the dynamics between humans and machines in the context of the forth industrial revolution which is provided in this thesis.

1.2 Motivation

While the last section gave an overview of the theoretical discussion, this thesis covers a tangible Industry 4.0 application in praxis. In the research project "MainTelRob", the Zentrum für Telematik e.V. (ZfT) and its project partners, KUKA Industries and Procter & Gamble, have developed novel technical approaches and procedures for modern telemaintenance. We wanted to show, that the integration of computer science and communication technology in automation industry has a huge potential for productivity. We chose telemaintenance, because there is a high demand for telematics solutions there.

Special machine manufacturers, especially robotics companies, sell their technology worldwide. Some factories, for example in emerging economies, lack qualified personnel for repair and maintenance tasks. When a severe failure occurs, an expert of the manufacturer needs to fly there, which leads to long down times of the machine or even the whole production line. With the development of data networks, a huge part of those travels can be omitted, if appropriate teleoperation equipment is provided.

Typical telemaintenance tasks are for example the analysis of a robot failure or difficult repair operations. The service department of KUKA Industries is responsible for the worldwide distributed customers who own more than one robot. Currently such tasks are offered via phone support and service staff which travels abroad. They want to expand their service activities on telemaintenance and struggle with the high demands of teleoperation, especially regarding security infrastructure.

In addition, the facility of P&G in Marktheidenfeld has to keep up with the high international standards of Procter & Gamble and wants to minimize machine downtimes. The quality of a molding process with involvement of industry robots is influenced by multi-dimensional parameters and breakdown times come with very high costs. Like 71.6% of all German companies [8], P&G sees a huge potential for early information on their production system, but complain about the insufficient quality and the lack of currentness of data.

There is always a need for one or more humans to perform the different tasks mentioned above. If software is used, the human machine interface plays a key role for the success of the maintenance action. When it comes to telematics applications, the human machine interface for the external expert is even more important. The key goal for the external expert is to understand "what is going on" at the remote plant (situation awareness). But also the local worker need software support to function in such a scenario. Despite of these facts and several findings on the importance of reducing human errors, usability methods and user centered design processes are currently not used widely in the field of industrial automation.

We therefore regard the introduction of such methods for industrial applications as a necessary step towards the "4th industrial revolution" and want to share our experiences in this thesis.

1.3 Topic description

This thesis describes the development and evaluation of a new telemaintenance system. As reference situation serves the research project "MainTelRob: Maintenance and Telematics for Robots". The specific results can be used as a starting point for projects which have one or more questions similar to the following list:

- Which research disciplines are necessary in order to derive future models for modern telemaintenance in an industrial internet context?
- How can a modern telemaintenance architecture look like?
- Which components are required for a modern telemaintenance infrastructure?
- Which actual trends from ICT can be used in the context of telemaintenance?
- Which technologies can be used locally at the plant?
- Which demands on the communication infrastructure does a telemaintenance scenario raise?
- How can the use of the communication link be optimized?
- How can you provide teleoperation while maintaining safety?
- Which security questions arise in the context of telemaintenance?

- Which Augmented Reality approaches on are useful for local maintenance and repair scenarios?
- How can the interfaces be designed, so that the operator is not confused and has to perform complicated mental transformations?
- Is it possible to optimize the plant process further? How can this task in general be supported for remote access?
- Which technologies can be used in the telemaintenance center? Which methods can be used without direct plant access?
- Which Augmented Reality approaches are useful for the application of telemaintenance?
- How can we provide the feeling of "standing beneath the plant" to the external operator? Which camera positions are required?
- How to use user centered design in research projects in the production industry? Which methods can be applied?
- How can the state of the art in telematics be of use in the automation industry?

This thesis shows the development process of a telemaintenance system, explains different methods and solutions and draws qualitative and quantitative comparison to the state of the art.

1.4 Summary of contributions

In order to enable a performant teleoperation for a plant with an industrial manipulator, a human centered design of all system elements in the telemaintenance chain is required. Therefore, this thesis aims at optimizing the different elements of a telemaintenance scenario from an overall system performance point of view including the human as design factor. The project itself was structured in six scenarios [9] integrating control theory, networking, security and human factors.

This work contributed to the international knowledge in the following points:

- Application of human centered design methods for the entire spectrum of telemaintenance tasks [10]
- Implementation of an all-in-one concept for a mobile device [11], together with a user study [12] for the new tablet PC based robot programming device compared to the old teach pendant
- User study on local repair guided by Augmented Reality [13]
- Teleoperation of an industrial robot [14] with video interaction, enabled by an innovative networking approach [15] [16] [17] and networked control [18]

- Remote teleanalysis for a single cycle [19] or a multitude of cycles [20]
- Analysis of situation awareness in a telemaintenance scenario [21] and further investigation of situation awareness in dependence on viewpoint and Augmented Reality [22]

1.5 Thesis outline

The remaining part of this monograph is structured as follows. Chapter 2 describes the theoretical fundamentals, on which the work is based. It includes a detailed analysis of related work on machine-to-machine, human-to-machine, human-to-human and machine-to-human communication, as far as it is applicable to a telemaintenance setup. This supports and strengthens the understanding of the overall telemaintenance system including the human. Based on this, design criteria for telemaintenance systems are derived. Finally, the knowledge is put together in an extended human supervisory control model.

Chapter 3 describes the "MainTelRob" project context as far as it is required for the understanding of this work. The initial situation of the plant and the design process are described in general. Also, the six scenarios with their specific use cases are explained in the context of Human Supervisory Control which helps to understand the following chapters.

Chapter 4 investigates the locally performed tasks, like inspection and repair, realized in an integrated all-in-one strategy in a mobile device. The new strategy of KUKA Industries of using a tablet PC teach pendant is examined under a human factor perspective. Finally, Augmented Reality is used to support repair tasks at the switch cabinet of the manipulator.

Chapter 5 introduces the concept of situation awareness for remote analysis tasks like optimization and failure detection. The chapter describes the developed architecture for Condition Monitoring and sensor data transmission, as well as the human machine interfaces used for single cycle analysis, multi-cycle analysis and Virtual Reality visualization.

Chapter 6 focuses on the developed concepts for the entire telemaintenance setting. It describes the architecture used for remote access on the plant and the robot, as well as the new developed video interaction principle. In a proof-of-concept experiment, the different parts of the architecture succeed in a remotely supervised motor exchange. Further research involved using Augmented Reality for understanding the plant setting and process as a first essential step to telemaintenance.

In Chapter 7 a short summary, final conclusions, and directions of future research based on this work are presented.

Chapter 2

Fundamentals

In the ongoing discussion about Industry 4.0, the remote maintenance (= telemaintenance) and the teleoperation of factories is a important topic. In order to develop software for this situation, a deeper understanding of this setting is needed. Also some fundamentals need to be discussed.

This chapter starts with an overview of the research field where the specific research questions will be situated. After briefly outlining the main fields of research and introducing some basic theories, we will summarize the knowledge in a theoretical synthesis which will guide us through the rest of the work.

2.1 Overview of the field

2.1.1 Telematic communication in an industrial internet setting

Very early, the societal importance of the combination of telecommunication, automation technology and computer science was discovered and defined as "Telematics" [23], as a of the three disciplines, mentioned initially in [24]. The Internet is one of the key technologies of the digital change and internet-based telematics will most certainly make the difference between "Industry 3.0" and "Industry 4.0".

The telematics settings in "Industry 4.0" or "Industrial Internet" introduced in Chapter 1 can be generalized in Figure 2.1, which gives a good base to a general theoretical discussion. It depicts schematically different exemplary communication partners, each representing one or more similar entities. The different communication paths are named according to communication source and recipient.

In Figure 2.1 we see a plant labeled "Machine 1" which contains an industrial robot and several sensors on the upper left. The plant is connected to the central cloud icon representing the Internet or, more generally, a generic network. There are other machines involved, like "Machine 2", a server with several clients or software agents depicted on the lower left. Between the server and the plant, the communication through the network is labeled "M2M", which is short for "machine-to-machine-communication". This term is widely used in a variety of senses (compare for example [25]), but we come to exact definitions later. There are also several humans connected to the same network – with specific computer environment on the upper right or mobile devices on the lower right. There are more possible communication paths. They are labeled accordingly to "M2M" with "M2H" for machine-to-human

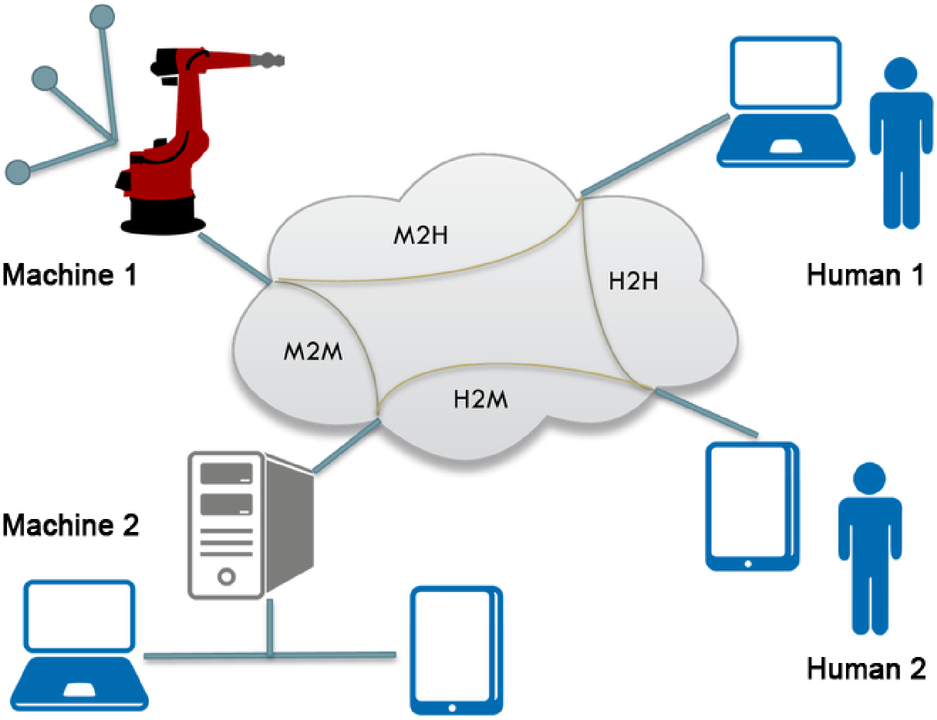


Figure 2.1: *Communication in a general industrial internet setting*

communication, "H2H" for human-to-human communication and "H2M" for human-to-machine communication.

Although all communication is digitally transferred through the network, the communication types differ substantially. While M2M needs specific predefined protocols and interfaces, the inter-human communication is structured completely different. A machine can display its status, e.g. malfunction to the user in warning signals or status interfaces (M2H). There is a huge difference to the other way, H2M, where humans for example look deeply into the machine inwards or reprogram the robot. As the rules of those communication paths differ, also the scientific methods to analyze the communication are extremely different.

2.1.2 Definitions

In this work we focus on a specific telematics application, the telemaintenance, which comprises the spatial separated access to IT systems for maintenance and repair. This is more than remote administration. After a brief discussion on maintenance in general and the special field of e-maintenance we derive the definition we will use in this work.

2.1.2.1 Maintenance

There are several international norms, which specify the field of maintenance. In the anglophone countries, it combines the field maintenance, repair and operations (MRO), which may be defined as a "combination of all technical and management actions intended to retain an item in, or restore it to, a state in which it can perform as required" (following the International Electrotechnial Commission IEC [26]).

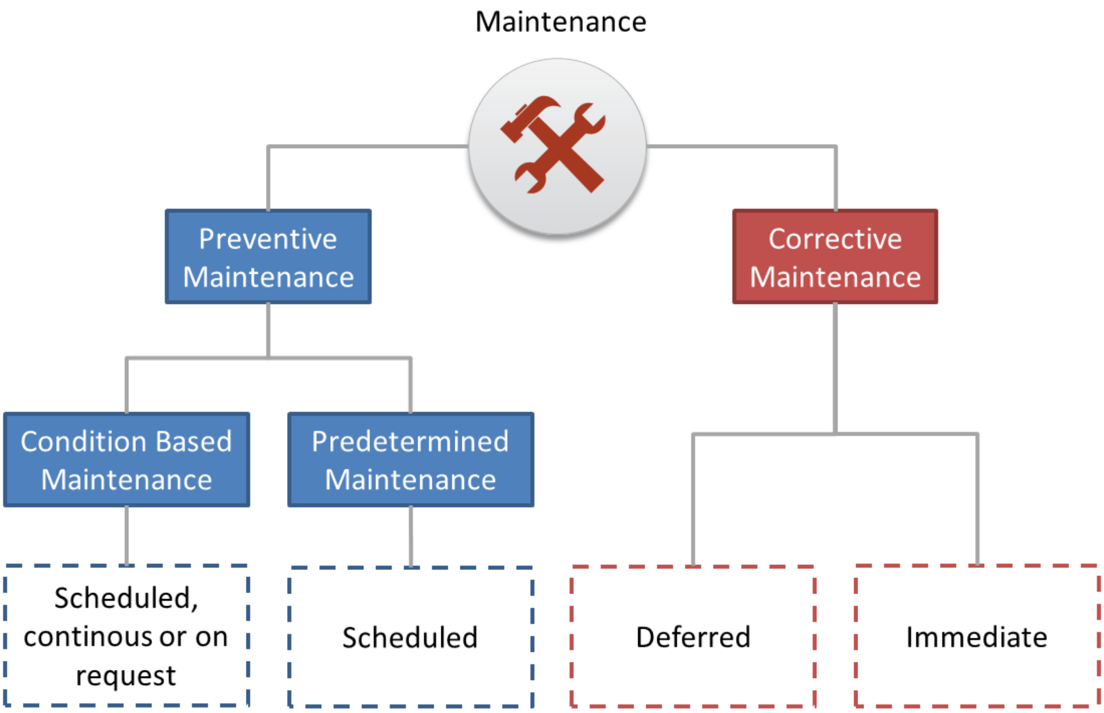


Figure 2.2: *Different maintenance types (SS-EN 13306:2001 based on IEC 60050-191 [27] as referenced in [28])*

As displayed in Figure 2.2, several types of maintenance can be classified. Whereas corrective maintenance involves the restoration of failed equipment after a fault, the preventive maintenance is used to prevent faults from occurring.

In Germany the term maintenance covers the basic measures inspection ("Inspektion"), maintenance ("Wartung"), restoration or repair ("Instandsetzung"), overhaul ("Verbesserung") and weak spot analysis ("Schwachstellenanalyse") (DIN 31051 [29]). These terms can be understood with the help of Figure 2.3, which arranges the terms around the life cycle of a machine.

At the initial operation, the working supply is per definition 100% and the machine works in its target condition. During the operation, the working supply sinks due to abrasion. This is not necessarily a continuous process as depicted. By performing inspections, technicians can ascertain the current condition of the working supply. It will sink further rapidly, if no maintenance measures are taken. With the help of maintenance actions, the abrasion can be reduced, so that the working supply

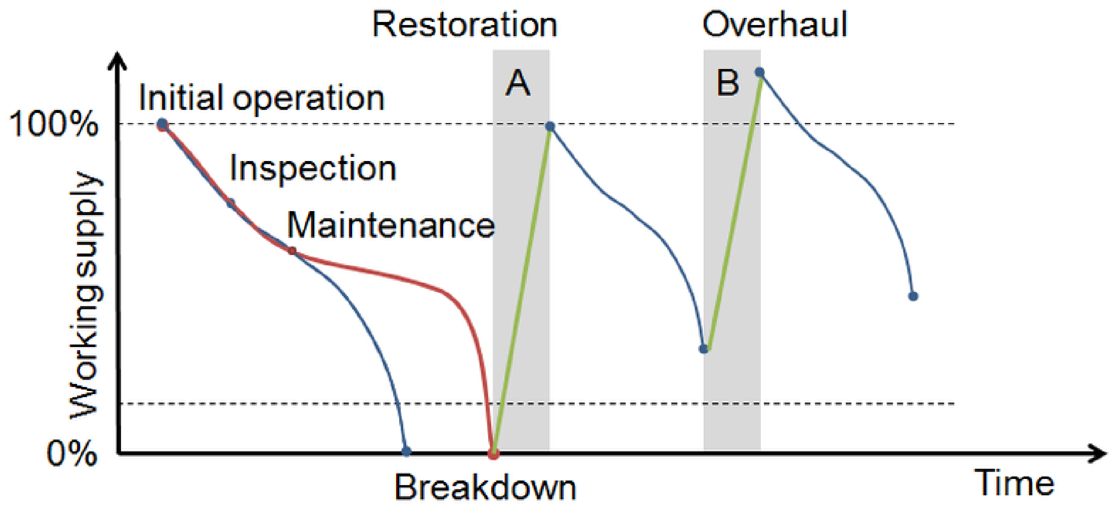


Figure 2.3: *Visualization of an asset history and its working supply (with changes out of [30])*

sinks less steep. Both maintenance and inspection can be considered as preventive actions, which are scheduled periodically in the classic case.

Despite maintenance actions, the working supply sinks nevertheless, so that it will one day fall below the threshold level of damage and the machine has a breakdown. The working supply can be restored (in theory) completely with a restoration (A). Then the decline of the working supply starts again. This is a corrective maintenance action. Such restoration processes can alternatively take place before a breakdown (preventive maintenance) and also include improvement or overhaul measures (B), so that a breakdown can be prevented and the working supply can be raised across 100% [30].

The best constellation for maintenance is determined with a cost-efficiency analysis. Breakdowns are extremely costly, as every minute without production is very expensive. This applies especially, if the plant is on a critical path in the production line. In case of such an unexpected fault, already the diagnosis can take a long time. Thus, in the best case, breakdowns should not happen at all. The same is true for so called immediate corrective maintenance measures which are performed in the case of a breakdown. Other maintenance measures can be deferred to a better instant of time and are called deferred corrective maintenance. The corrective maintenance types are displayed in Figure 2.2.

In general, corrective maintenance measures tend to be more costly then preventive maintenance measures. In the first case, something is actually broken and needs to be mended. Hence, the less corrective maintenance actions are undertaken the better. If a lot of preventive maintenance actions are performed, breakdowns can be prevented (see Figure 2.3). But this is also expensive. As a result, it is optimal, to drive the plant as near to the threshold level of damage as possible [28]. This qualitative cost-efficiency analysis can be summarized: The goal is to perform as much maintenance as necessary and as little as possible.

2.1.2.2 E-maintenance and predictive maintenance

In our opinion, telemaintenance is a part of the field of e-maintenance, although both are used synonymously in some publications because they are closely related. E-maintenance has been defined as "the integration of the information and communication technologies (ICT) within the maintenance strategy or plan" (p. 1 [31]) and has been an interesting field of research for years.

Particularly, e-maintenance means replacing traditional maintenance processes, which are mainly based on human experiences, by technological systems. It is also referred to as "predictive" maintenance. The condition of the equipment (and the working supply) can be determined with a condition maintenance system very precisely, so that maintenance tasks are performed only when justified. The key is "the right information at the right time" for being able to transform unplanned stops like breakdowns into planned stops in which e.g. deferred corrective maintenance actions can be performed. Also remote diagnostic functions can be used for the estimation of the optimal time. With such measures, nearly zero downtime performance can be reached. There are other potential advantages like increased equipment lifetime or improved plant safety, which are discussed in the context of predictive maintenance.

Telematics and telemaintenance have only been established in the case of high-value investment goods, because of the special demands of the services and the appropriate security concepts. These have not been manageable for small and medium-sized enterprises (SME). For example only a small amount of the SMEs are informed about the security requirements and the law frame conditions of cloud computing although a lot of them are already using that technology [32].

As a summary, all those fields lay the base for the developments of new applications. But also the connection to other parts of the company needs to be addressed. A newer article [33] states that an "Enterprise 2.0 model" has been developed, which influences the maintenance. This shows that the "Industry 4.0" discussion has been of huge interest in the context of maintenance.

2.1.2.3 Definition of telemaintenance

The International Electrotechnial Commission (IEC) defines "remote maintenance" as "maintenance performed without direct personnel access to the item" [26]. It was important for us, to emphasize the idea of "Industry 4.0" and telematics in line with the e-maintenance movement described in the last section. We also needed to connect to the German maintenance terms, because the project was established with German industry partners.

For this thesis we define telemaintenance as the combination of electronic maintenance actions with telematic analysis and teleoperation as depicted in Figure 2.4. It involves the four main tasks inspection, repair, optimization and teleoperation, while all of the tasks need to be performed remotely in cooperation with a local supporter. The optimization task summarizes overhaul and weak spot analysis.

In most service cases, the remote expert needs to get an understanding of the specific plant and the concrete problem. Thus, optimization and teleoperation need an analysis phase beforehand (teleanalysis). This can be based on inspection and repair history, but also on specific inspection measures performed at the beginning of a remote maintenance action.

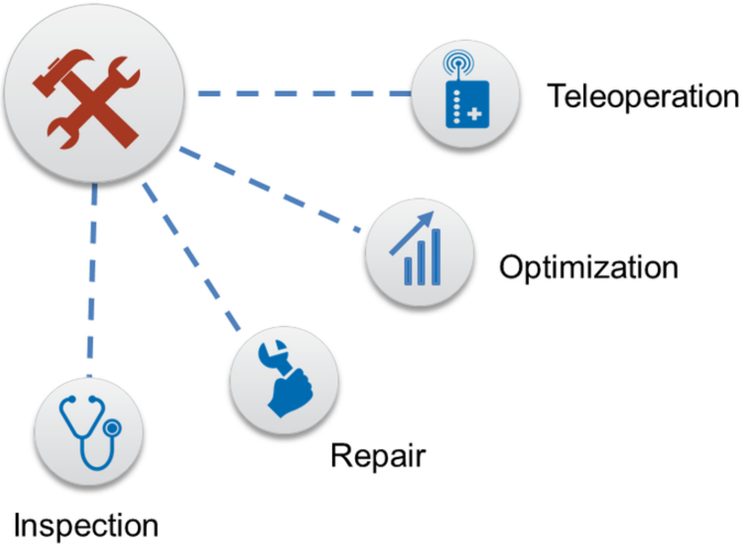


Figure 2.4: *Different tasks for telemaintenance*

2.2 M2M communication

As mentioned above, M2M is considered to be "Machine-to-Machine" in this publication. This is one of the hype-terms; some see it as the next technology revolution after the computer and the Internet. This results in a multitude of definitions circulating in the literature. Part of the confusion can be attributed to the fact, that M2M is not something new.

This publication follows the definition given in [25], which is visualized in Figure 2.5. The M2M communication in its most basic form (Figure 2.5 a) is the bidirectional data exchange between a device and a business application via a network. But there can also be a group of devices connected directly to the network (Figure 2.5 b) or communicating via a gateway (Figure 2.5 c). There are several characteristics which describe the M2M applications, like multitude, variety, invisibility and intrusiveness [25]. The communication network has a key role, so M2M can also be interpreted as a shortened synonym for M2(CN2)M: Machine-to(Communication-Network-to)Machine [25].

This M2M model includes industrial instrumentation: The depicted "device" is then a sensor or meter which records data (such as temperature, etc.) and sends it to the application software. Also several sensors can communicate with the ap-

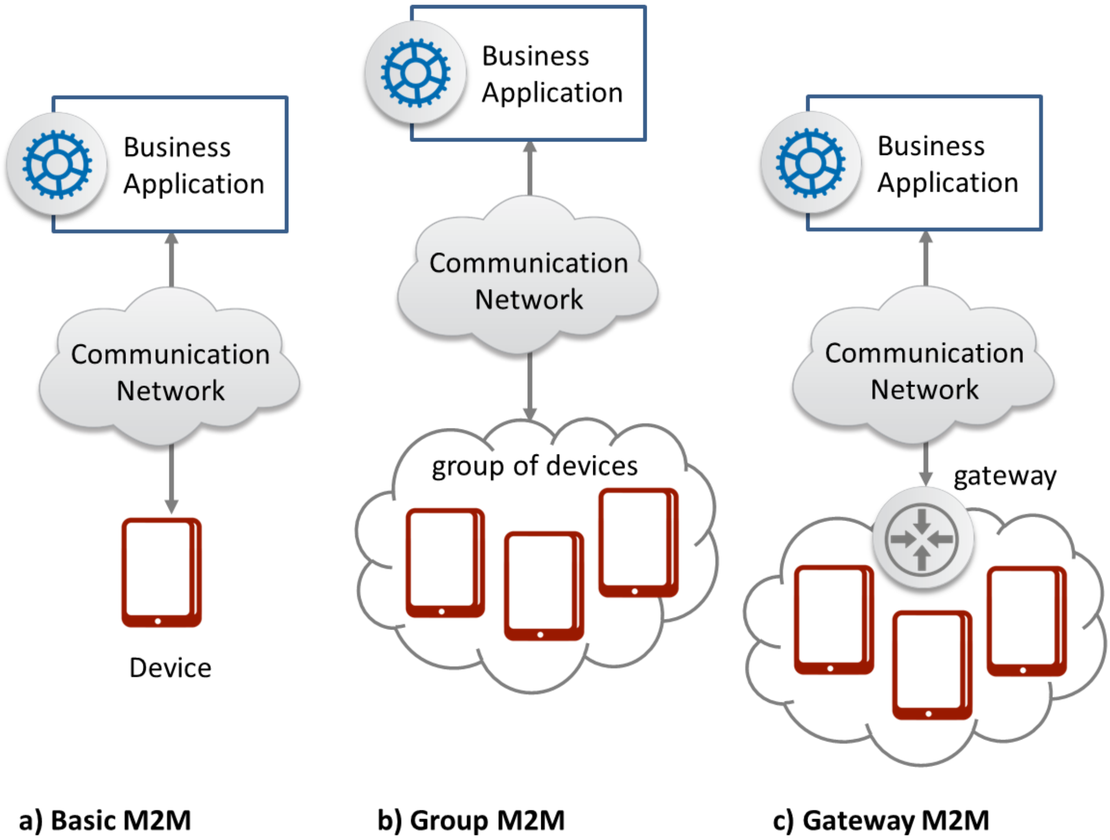


Figure 2.5: *Concept of M2M communication (following [25])*

plication. This aligns with the Industry 4.0 topic "Cyber Physical System" (CPS) which "usually comprise a network of physically distributed embedded sensors and actuators equipped with computing and communicating capabilities" ([4] p. 1).

2.2.1 Robot telematics

In our case, the large field of M2M and CPS needs to be limited on industrial manipulators, sensors and surrounding plant components connected via the Internet with processing servers (compare Figure 2.1). Of course, additional sensor equipment can also independently connect to the network. In this section, we briefly cover some of the basic concepts which will be used throughout this work.

2.2.1.1 Industrial automation

In order to understand the industrial production process, we use the scheme depicted in Figure 2.6. The production process can be described as a function with input u and output y. The closed loop has different layers, in order to alter the input u so that the required output is achieved.

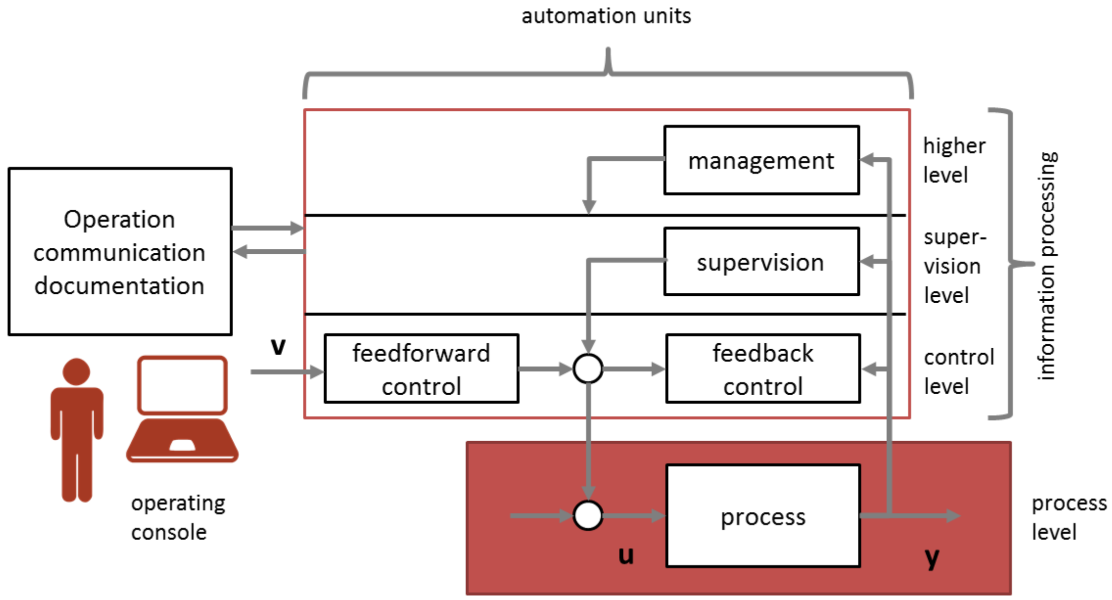


Figure 2.6: *Simplified scheme of process automation (following [34])*

This view perfectly connects with the so-called "Automation pyramid", a level oriented classification based on the IEC 62264 [35], an international standard for enterprise-control system integration. It covers five levels displayed in Figure 2.7, which may have fluent boundaries based on the facility.

- Level 4 is the enterprise level with the Enterprise Resource Planning (ERP) system
- Level 3 covers the site level with the Manufacturing Execution System (MES)
- Level 2 is the process supervision with the Supervisory Control and Data Acquisition (SCADA) system
- Level 1 involves process control on Programmable Logic Controllers (PLC) together with their inputs and outputs
- Level 0 is the process level

2.2.1.2 Data sources: manipulator and injection molding

A industrial robot is defined according to ISO 8373 [36] as follows: "An automatically controlled, reprogrammable, multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications."

The manipulator is provided with a movement program, which usually consists of manually taught positions. A schematic visualization of a robot is depicted in Figure

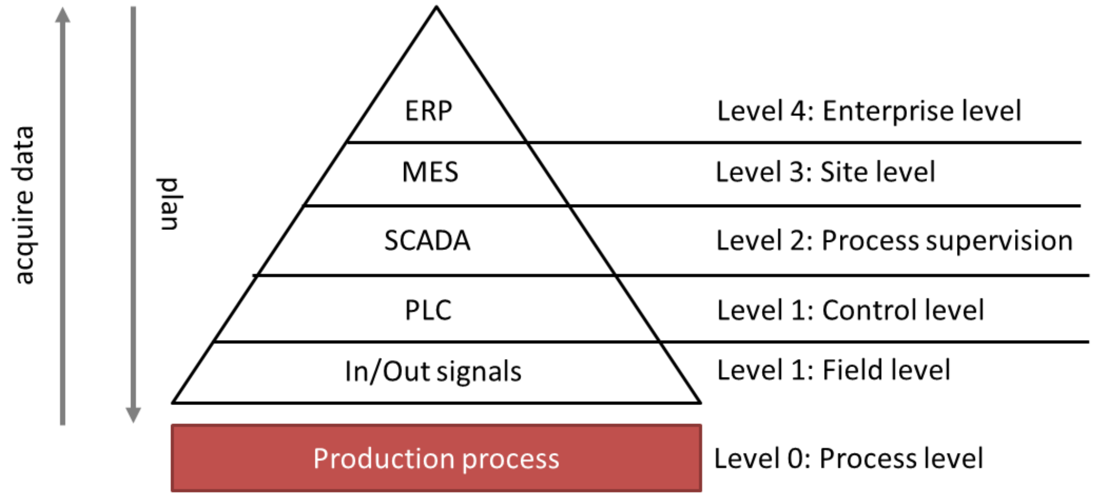


Figure 2.7: *Automation pyramide*

2.8 a). Robots are categorized on base of their configuration – the composition of the axes (or links) and whether they are connected with prismatic or revolute joints. For further reading see i.e. [37].

In this publication, we consider a special type of Cartesian robot (schema in Figure 2.8 b) with three axes forming a coordinate system with prismatic joints.

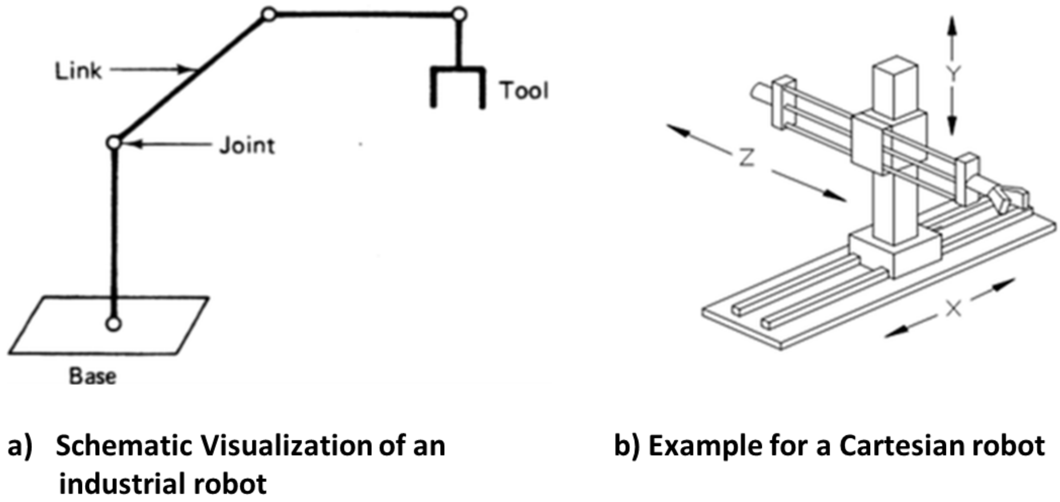


Figure 2.8: *Industrial manipulators*

The manipulator has several internal sensors used for position control and status diagnostics. It can use additional external sensors and communicate with different other parts of the plant, i.e. the injection molding machine.

Injection molding is the most important forming technology for plastics processing. Ready formed parts with complex geometry can be produced in one work step in almost arbitrary size. The quality of the produced parts needs to be optimized on a multi-dimensional parameter space [38]. This is why injection molding machines also have numerous internal and external sensors, i.e. for measuring the temperature in different parts of the machine.

2.2.1.3 Signals

In order to describe the signals produced by devices, we distinguish between continuous and digitized data as depicted in Figure 2.9. This distinction applies respective to the time axis and also to the value axis. An analog sensor for example generates a time-continuous and value-continuous signal. The process of converting a continuous signal into a numeric sequence is called sampling. The Nyquist-Shannon-Kotelnikov theorem (sampling theorem) states, that a specific minimal sample-rate needs to be used in order to reconstruct the signal perfectly. "If a function $f(t)$ contains no frequencies higher than W cps [counts per second], it is completely determined by giving its ordinates at a series of points spaced $\frac{1}{2}W$ seconds apart." ([39] p. 448)

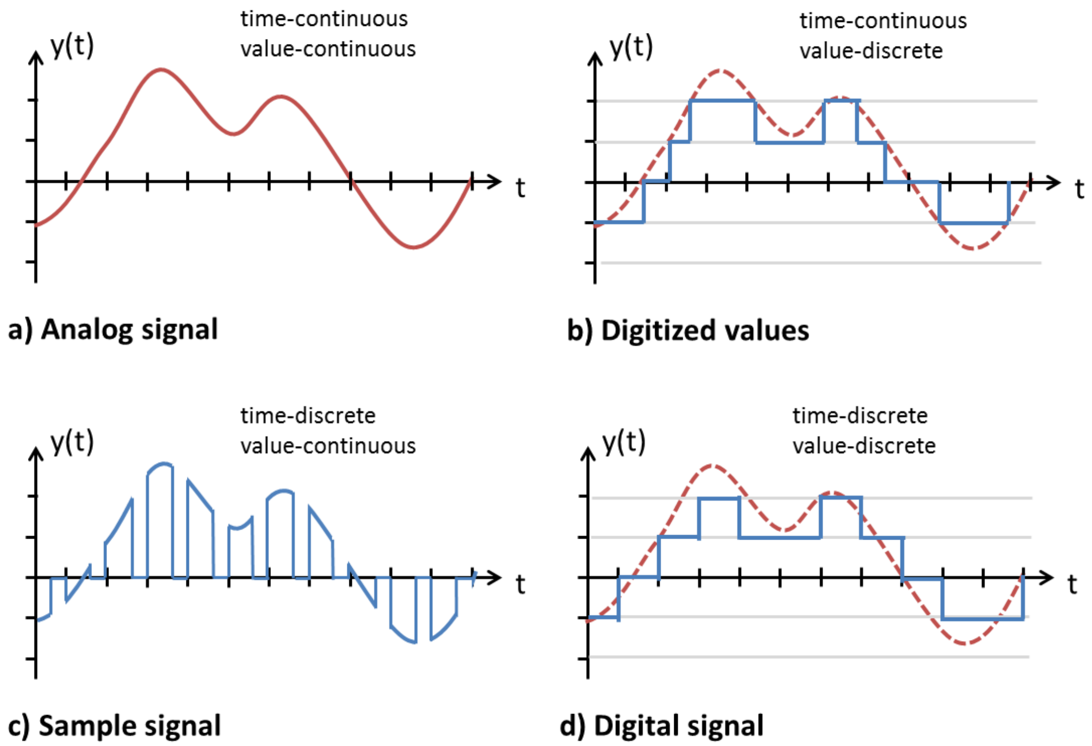


Figure 2.9: *Visualization of quantization*

Another important factor is time synchronization: Data processing and data transmission needs time. If a M2M application uses more than one device, a time synchronization of the signals is necessary. Otherwise two signals received at the same time in the information processing may refer to different instances of time.

2.2.1.4 Information transmission

The sender-receiver model, also known as Shannon Weaver model, describes communication as transmission of a message from one entity to another. The message is encoded and transferred as a signal via an information channel. Successful communication means that the receiver gets the same message the sender has sent. Thus, sender and receiver need to use the same codification, so that the decoded message is the same as the original one. The encoded message also can be altered by noise, so this needs to be detected and prevented. The original model of Shannon [39] is depicted in Figure 2.10.

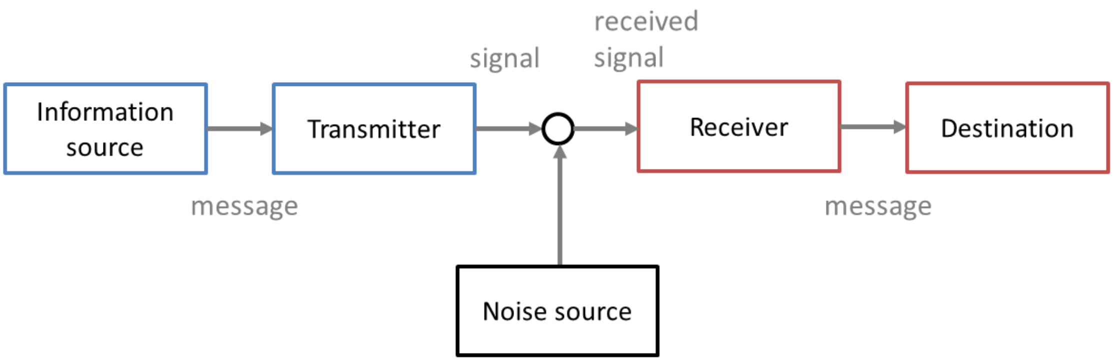


Figure 2.10: *Schema for information transmission [39]*

Based on this general model, a multitude of questions need to be solved in order to receive an appropriate network connection. First, the information channel can consist of different physical media like copper cables or air (for radio transmission). Second, following the definitions in the last section, we also distinguish between analog transmission and digital transmission. Third, both partners need to agree upon several things: They need to negotiate who is sender and who is receiver in a bidirectional connection, which code they use, how much information they can send and so forth. Even more complicated, the Internet is a network of networks with a multitude of communication partners, different structures and transmission media.

To decrease complexity, most networks are organized in layers. Every layer provides specific services to the layer above, but hides details, so that the highest layer can focus on the message content. In order to explain some basic concepts, which characterize connection quality, we use a simplified model depicted in Figure 2.11 for further reading see [40], [41] or [42].

It is important to understand that the internet is based on packet switching. Messages are fragmented in packets of the same size and sent individually. In our example in Figure 2.11, a server is located in central Europe and has a message. Signals travel with limited speed, so that the packets have a delay between the sender in Europe and the receiver in China. As the packets are sent individually, they can take different paths and have different delays. The variation of the packet arrival times is called jitter. Packets may arrive in a different order or may not arrive

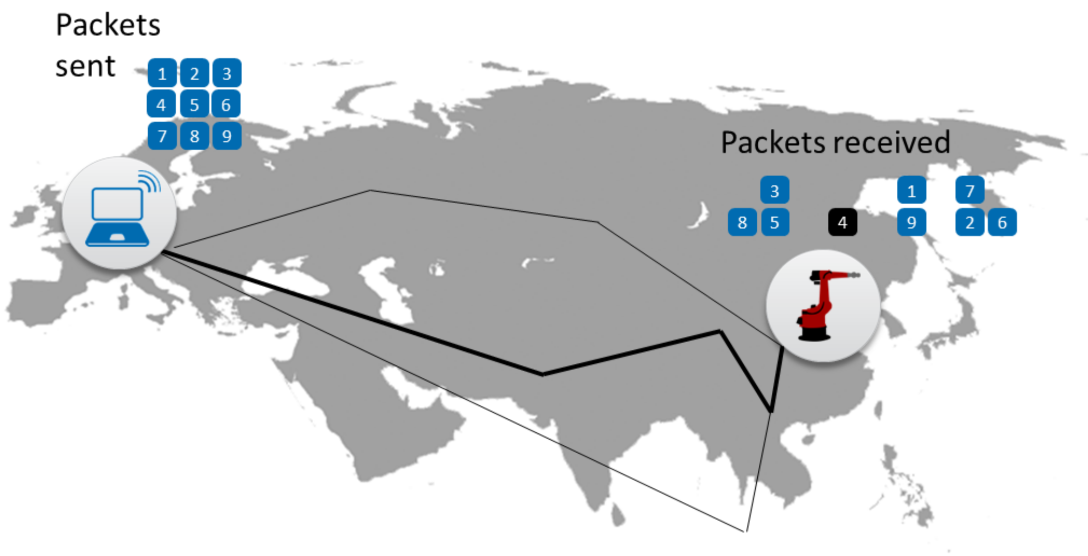


Figure 2.11: *Schematic visualization of packet switching: A message is sent from Europe to China in packets, which will arrive on different transmission paths, in a different order, in different time or not at all (visualized in packet 4)*

at all. This is called packet loss and visualized with a missing package (packet 4).

The last network characterization covered in this brief survey is the amount of data, which can be transferred, the throughput. There is a theoretical limit to the maximum rate at which data can be transferred over a communication channel of a specified bandwidth in the presence of noise (channel capacity) [39]. The term bandwidth describes the range of frequencies that make up a signal [41] and is used for characterization. But the amount of user data or payload is further shortened dependent on the used codification (overhead).

2.2.2 Fault detection and diagnosis

When we define maintenance as restoring an item to a state in which it can perform its required function, the precondition for that is a proper fault detection and diagnosis. In a plant, a multitude of faults can happen. For example, a sensor fault might result in a biased value, an actuator fault may lead to a parameter change and a plant fault can cause a leak. In most cases, the symptoms of a fault are observable effects, but also those may be altered by noise and disturbances.

There are several different definitions of faults as several disciplines deal with this matter. In this thesis, we follow the definition of [34], which describes a fault as "an unpermitted deviation of at least one characteristic property (feature) of the system from the acceptable, usual standard condition." ([34] p. 17)

There are many different types of faults, which are also called errors, especially if directly caused by humans. Faults may develop abruptly (stepwise) or incipiently (driftwise), as depicted in Figure 2.12. Some faults may not affect the correct functioning of a system, but others may initiate a failure or a malfunction. Those terms have also been defined precisely in [34]: "A failure is a permanent interruption of a system's ability to perform a required function under specified operating conditions." ([34] p. 18) and "A malfunction is an intermittent irregularity in the fulfillment of a system's desired function." ([34] p. 18) Both failure and malfunction result from one or more faults. The relation between the terms is depicted in Figure 2.12.

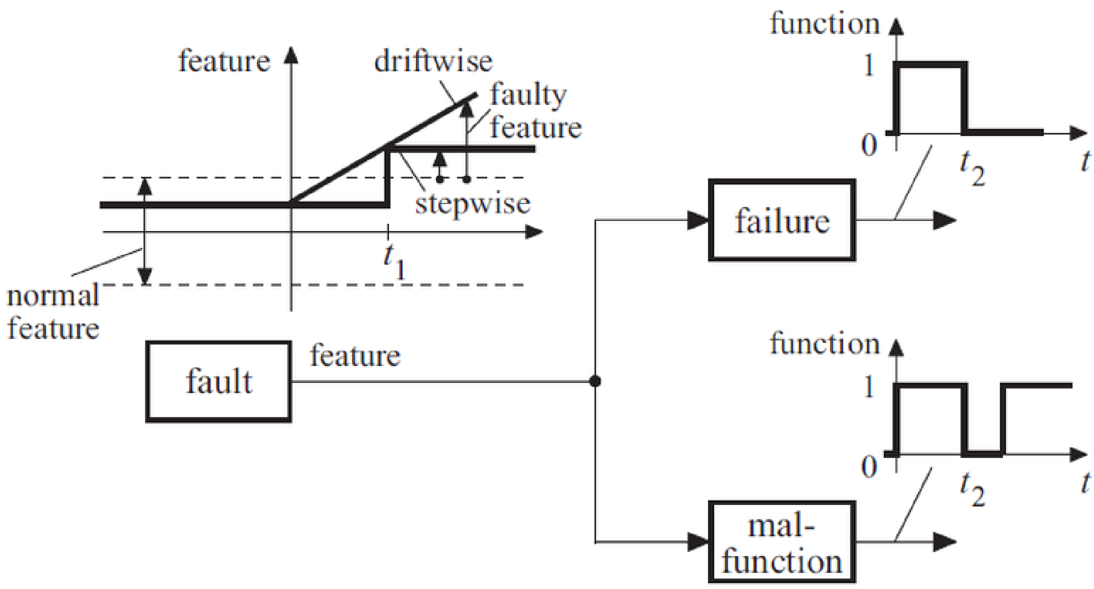


Figure 2.12: *Development of the events "failure" or "malfunction" from a fault which causes a stepwise or driftwise change of a feature (out of [34])*

If we talk about fault diagnosis, the first task is to detect whether there is a fault (fault detection). The second task is determining where the fault is (fault isolation). A deeper analysis allows the determination of the behavior of the fault (fault identification).

2.2.2.1 Fault diagnosis

As described above in Section 2.2.1.1, the production process can be regarded as a controlled process P. It is depicted in open loop in Figure 2.13 a) and we follow the definitions and parameter descriptions out of [34]. As usual, $U(t)$ is the input and $Y(t)$ the output signal. As described above, internal faults F_i and external faults F_e can appear. These faults $F(t)$ affect internal process parameters $\Theta(t)$ by $\Delta\Theta(t)$ or internal state variables $x(t)$ by $\Delta x(t)$ which change the measurable output $Y(t)$. But also natural process disturbances and noise $N(t)$ have influence on the output. The overall change is specified by the term $\Delta Y(t)$. In the closed loop version depicted in Figure 2.13 b) a Controller C is added. As outlined in [34] most importantly not only $Y(t)$ but also $U(t)$ should be monitored, because faults may be compensated by the closed loop and can not be detected until they are very large.

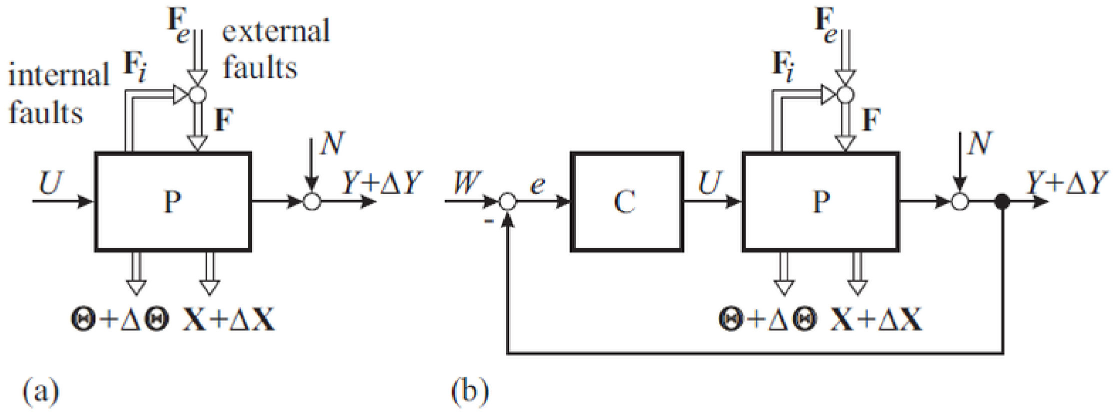


Figure 2.13: *Scheme of a process or product P influenced by faults F: a) process in open loop; b) process in closed loop (out of [34])*

2.2.2.2 Condition monitoring

Condition monitoring describes the "acquisition and processing of information and data that indicate the state of a machine over time. [...] The machine state deteriorates if faults or failures occur." (ISO 13372 [43]). As this definition is consistent with the definition for monitoring used in [34], the latter also introduces the term "supervision", which describes the "monitoring [of] a physical system and taking appropriate actions to maintain the operation in the case of faults" ([34] p. 322).

Following the classification in Figure 2.2, condition based maintenance is a preventive maintenance type. The International Electrotechnical Commission defines it as "preventive maintenance based on the assessment of physical condition" [26]. It utilizes condition monitoring technologies in order to determine the condition of an item, such as a machine, and thereby plan the maintenance schedule [28].

Figure 2.14 visualizes the condition based maintenance system architecture proposed by [44]. They divide the necessary actions in diagnostics and prognostics, while the first covers classical fault diagnosis described above and the latter tries to forecast the process on a larger model in order to derive maintenance decisions. Here also data from several similar plants can be used, a phenomenon often described as "Big Data", which will be described in the next section.

2.2.3 Big data and data mining

The term "Big Data" has been mentioned a lot recently, some even call it the "oil of the 21st century". McKinsey & Company has even nominated it as "the next frontier for innovation, competition and productivity" [45]. In the year 2012 4.6 billion euro have been invested in Big Data related projects and services worldwide [46].

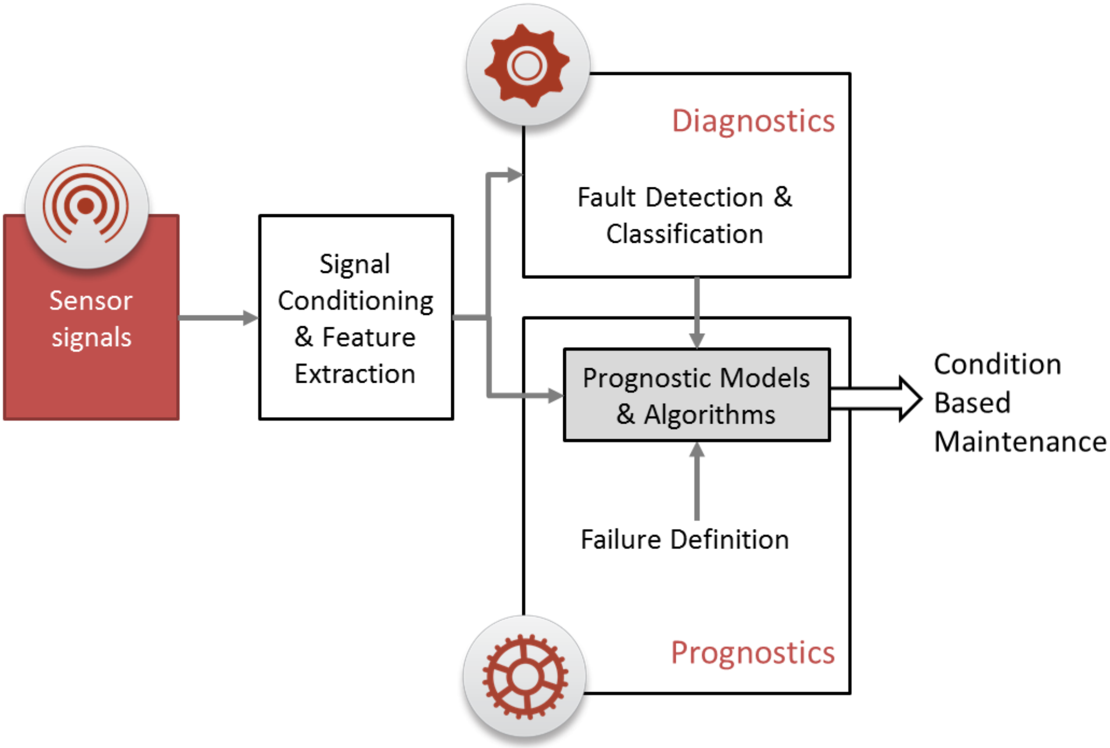


Figure 2.14: *Basic elements of diagnostics and prognostics for Condition Based Maintenance*

Although different definitions are still in use, scientific literature agrees, that the distinctive feature of Big Data and its challenges and benefits evolve from the increasing amount, velocity and variety of data [47].

From a scientific point of view, Big Data is not a new concept, but a new dimension in the cooperation of different scientific domains. The whole process is based upon the knowledge discovery from various data sources displayed in Figure 2.15. The "big data" is located in various data sources on the left, while the "data mining" process with its prestages tries to discover interesting and useful patterns and relationships. Data mining is a interdisciplinary field of artificial intelligence, machine learning, statistics and database systems.

2.2.3.1 Data

Data mining works on a data set of examples which are often called instances. This can be structured or unstructured data. In the first case, we can think of it as structured in a table. In the latter, we need to perform e.g. some search actions to transform it into a structured form. Each instance consists of numerous variables, which are called attributes.

If there is a specially designated attribute we can think about it as a column header. Data of this kind is called labeled data. Data without any specially designated attributed is called unlabeled data.

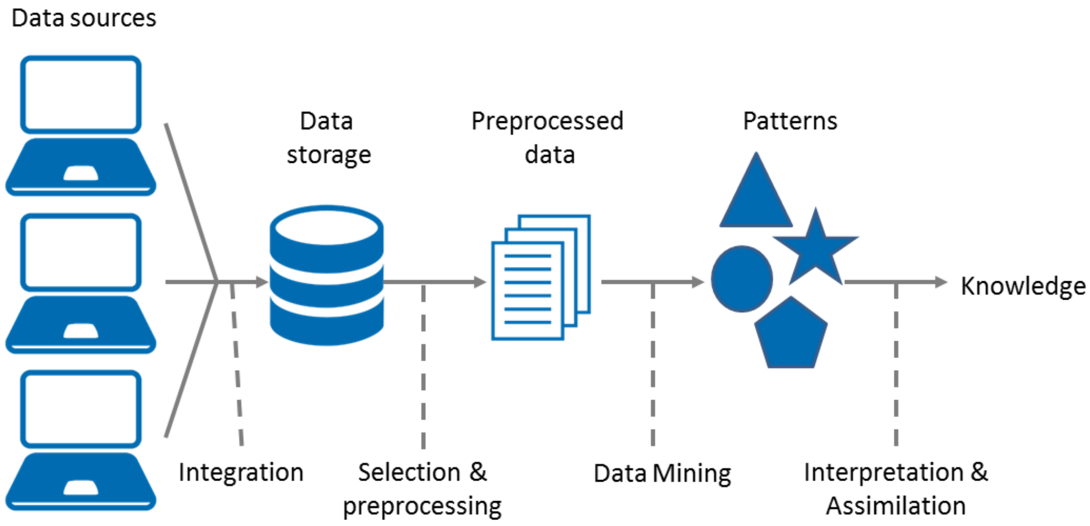


Figure 2.15: *Knowledge discovery schema from [48]*

Normally, a data set is partitioned into training data and validation data. The goal is to detect patterns in the training data. The associated algorithms are then used on the validation data to "sort" them according to the patterns. This task is different for different types of labeled data.

It is further divided in attribute types: It can be categorical, that means it describes a category like "good" or "house". Then the task of dividing data with respect to the attributes is called classification. Otherwise, the attributes can be numerical. Then the task is called regression [48].

Generally, recorded data needs to be verified [49]. Through different circumstances data can be altered, so that a result is faulty or the plotted data is visualized in a wrong way. Some errors are caused by programming errors or hardware errors in data acquisition. But of course the values may also have been generated by the observed object in that way, which tends to be the most interesting part of the data [50], [51].

It must be clarified, if data shall be visualized only or also explored. Users often see patterns in visualized data. Whether those patterns are recognized correctly can only be estimated with one or several confirmation steps afterwards [52].

2.2.3.2 Data mining methods

There are a lot of different data mining methods, which can be categorized in the following overview.

- **Statistical analysis**

  With statistical analysis methods, like linear and non-linear regression, a prediction of data streams can be derived. In many situations, a critical system state cannot be detected out of a single parameter, but out of the combination of several measurements. Thus, mathematical procedures are applied to enable a combined consideration of the data.

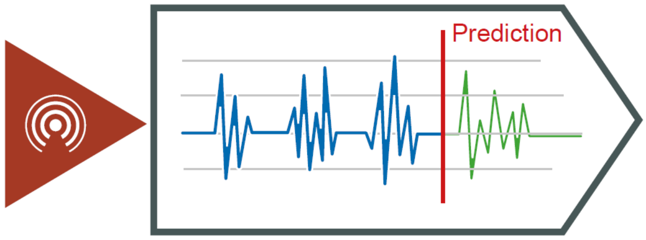


Figure 2.16: *Statistical regression for prediction of machine state*

- **Semantic rules**

  Often experts and technicians are able to describe connections between signals, states and events in the form of rules. If the desired behavior of several components is described with semantic, domain specific rules, a system deviation can be detected.

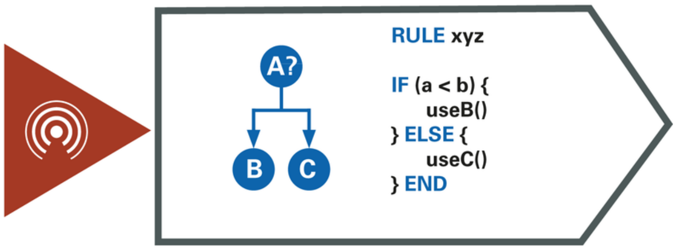


Figure 2.17: *Rule-based fault detection*

  It is important that experts and technicians define the rules as simply as possible. The use of a domain specific language is necessary to explain abstract concepts and interrelations.

- **Use of reference cases**

  Next to rule based connections, there is also a huge treasure trove of experience in form of reference cases. Data of the past of the plant describe on the one hand the target behavior of the plant and on the other hand show indications on the factors that led to a breakdown.

  With the help of Case-Based Reasoning (CBR) a "signature" of the current state of the plant can be derived. This can be compared to the huge amount of reference cases in the historic data as shown in Figure 2.18.

  Thus, critical situations and fault symptoms can be estimated by classification based on historic data.

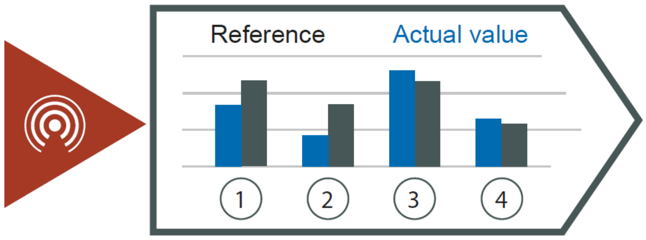


Figure 2.18: *Diagnosis with reference cases*

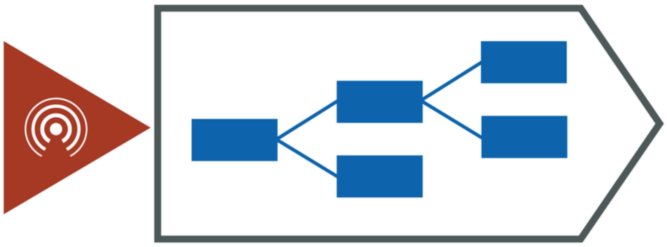


Figure 2.19: *Use of decision trees*

- **Decision trees**

  Besides, for the classification of data, decision trees can be designed (as shown in Figure 2.19), which decide on single attributes of a data set, in which category it needs to be sorted. Additionally nearest neighbor methods can be used to determine as much mutuality to other data sets as possible [48].

- **Combination of different methods**

  For recognizing complex system connections or detect faults, the methods described in the previous sections need to be combined as depicted in Figure 2.20. Each of the methods has specific advantages and disadvantages why an integration and interconnection is necessary.

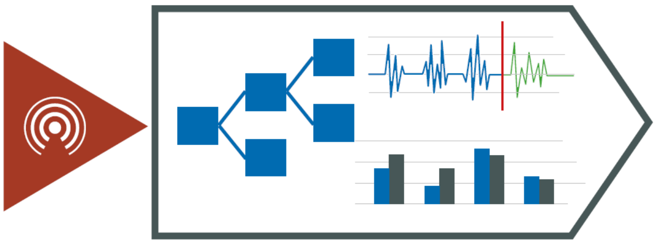


Figure 2.20: *Integration of different methods*

2.3 HCI - Human Computer Interaction

As explained initially, we regard the communication between (industrial) machines and humans instead of the large field of Human Computer Interaction in general. Nevertheless, a mutual understanding of some terms of the HCI research field is necessary.

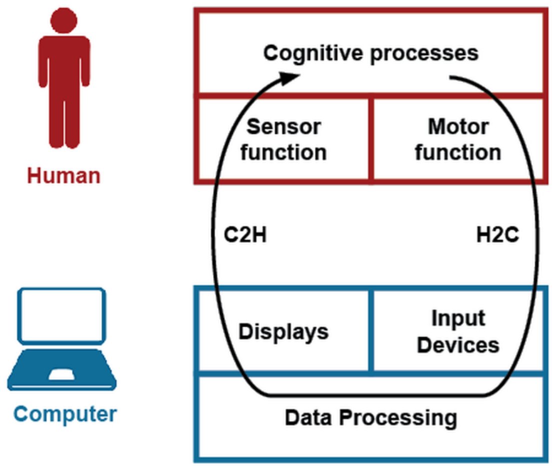


Figure 2.21: *Socio-technical system*

In order to analyze Human Computer Interaction, it is important to regard the "socio-technical system" as a whole, the "combination of organizational, technical, educational and cultural structures and interactions" ([53] p. 60). Human and computer both have inputs and outputs, which are connected in a feedback loop. Although we can distinguish between the human to computer (H2C) channel and the computer to human (C2H) channel, the complete feedback loop needs to be considered. This section serves as a base for the latter sections focusing on the channels.

2.3.1 HCI for industry

There are a lot of reasons for considering Human Computer Interaction elaborately during software development, especially in R&D projects. One of the most important reasons is the reduction of working memory load, if the software is easier to use. Most users want to solve a problem and do not especially plan to use any software. Computers and software are becoming unobtrusive and unremarkable, but the need for a high quality user experience does not disappear [54], [55].

In applied science projects, we often come across software displaying the characteristics of what can be called a "dancing bear" software, a term taken out of [56]. The author describes the occasion of a great idea triumphing over poor design: "Just having a bear that can dance leads one to overlook the fact that it cannot dance very well." ([54] p.7).

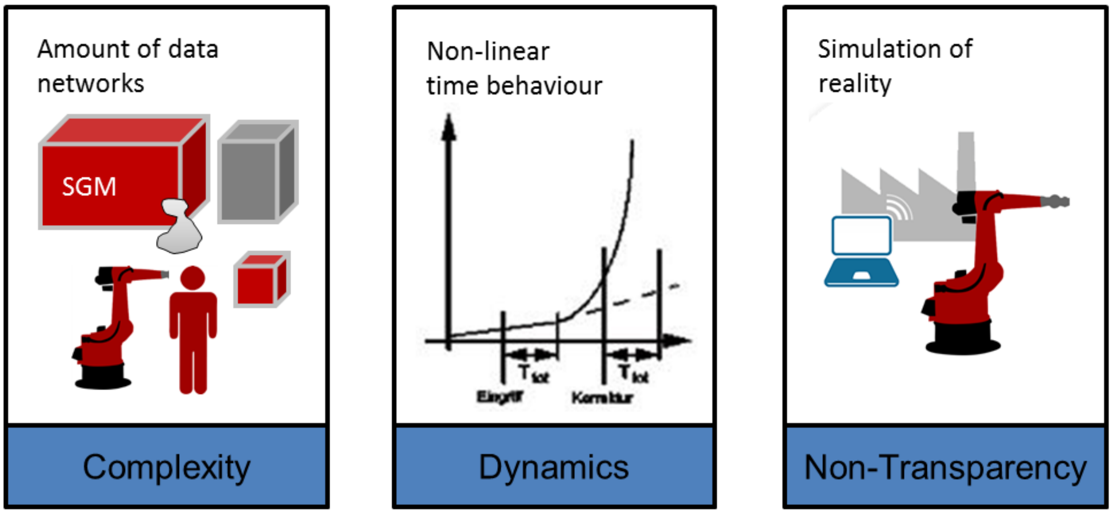


Figure 2.22: *Human error sources*

The role of a human-machine-interface is to "make sure that (1) the user can figure out what to do, and (2) the user can tell what is going on." ([57] p.188). Especially in the field of industrial automation, there seems to be an "operation crisis" [58], because studies showed, that the average machine operator only knows 50% of the function volume of his plant and needs two to three weeks of training. This might be reduced dramatically, if usability engineering is performed throughout the development process.

Another reason for the application of HCI methods is the reduction of human errors, which are extremely expensive and also possibly dangerous in the field of industrial automation. There are three major problem fields (depicted in 2.22):

- A **complex system** has a lot of inter-dependent features. The entirety is recognized by humans, but they lack knowledge of the individual interactions. Thus, the human is likely to draw wrong conclusions, which may lead to severe errors [58].

- **System dynamics** also impede the prediction of future system performance, because humans tend to anticipate linear time behavior, while a lot of technical systems show strongly nonlinear behavior. Without knowledge of the dynamic correlation, the human can produce errors [58].

- Modern technical systems are **not transparent** and act like a black box for the human, who cannot see, feel or hear the processes directly. He has to decide on the basis of incomplete information, which leads to problems especially in the case of sudden state shifts [58].

All those fields depicted in Figure 2.22 even increase within the 4th industrial revolution. The telemaintenance setting has high complexity, high dynamics and possibly also intransparency. Thus, this section covers several principles used to design and evaluate a system in order to prevent human errors.

2.3.2 Mental workload

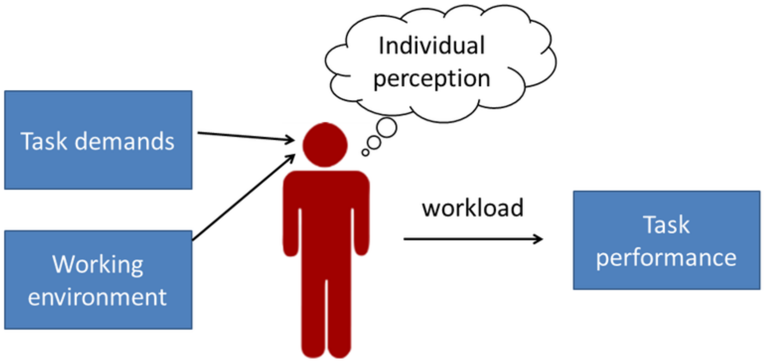


Figure 2.23: *Mental workload*

As depicted in Figure 2.23, the working environment and task demands affect the individually experienced workload, which leads to the human's task performance. Every person experiences workload differently, as the mental processing capacity varies from human to human. A high workload leads to stress, illness and an increasing error probability. But also a very low workload increases errors, as the operator experiences boredom, reduced vigilance, etc.. Workload covers mental and cognitive workload as well as physical and temporal workload [59]. For a human-centered design of human-machine-systems the human should be provided with a sufficient and adequate amount of information and tasks and considers himself as an important part of the process [58]. The task load can be measured with the NASA Task Load Index questionnaire [60] and provides a good scale for the overall rating of the system. This questionnaire is validated and widely used. Another method is provided by the SEA Scale [61].

2.3.3 Mental model

In order to understand the processes inside the human, we need to introduce the concept of mental models. The theory is explained in Figure 2.24: A person sees a blue box and generates a mental model in his or her mind. We will use this kind of visualization in the next sections.

This concept has been widely discussed in research community. The term "mental model" is believed to have originated in 1967 [62]. In the context of usability, Norman defined them as "the models people have of themselves, others, the environment, and the things with which they interact. People form mental models through experience, training, and instruction" ([57] p.17). They are imperfect, unstable and cannot be distinguished [63].

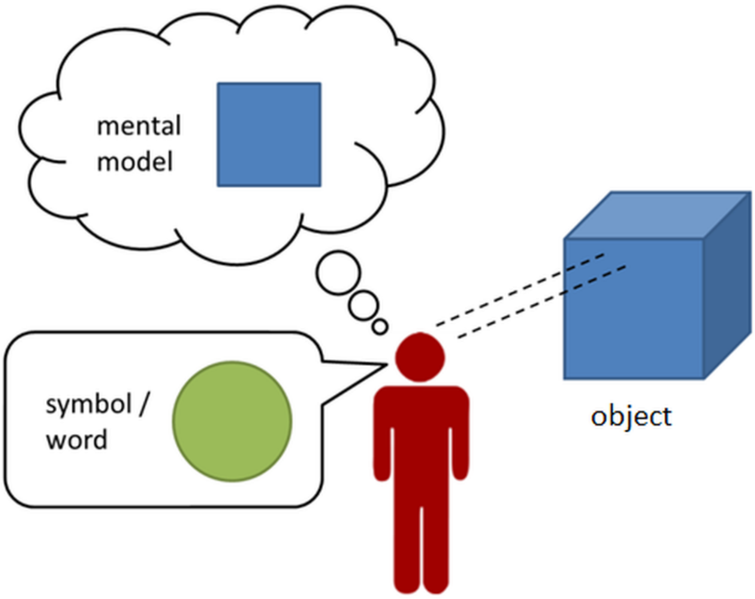


Figure 2.24: *Visualization of the concept of mental models and the semiotic triangle of symbol, mental model (thought or reference) and real object (referent)*

According to Sheridan the mental model "represents what a person thinks about some set of objects or events" ([64] p.75). He distinguishes between two forms: The first form is a qualitative mental model, which describes the interrelationships of a particular set of objects or events experienced, like a diagram how a particular device works. The second form is a quantitative cause-effect relation, like a model of how water in a tapered vase rises as it is being filled from a constant-rate faucet.

The symbols and words used to describe an object (like "blue box") are not directly connected to the object, they are connected via the mental model ("semiotic triangle" [65]), which means that two different people can have a different understanding of the term "blue box".

2.3.4 Conceptual model

As everyone knows from daily life, a special dynamic happens, when humans talk to each other about their mental models. Even more so, when they do not interact with each other directly. In our example depicted in Figure 2.25, a designer (red person on the left) has a specific mental model in his or her mind (blue box). He or she implements a system on base of that model (designer's conceptual model). The system has a "system image" (green circle) resulting from its physical structure including documentation, instruction and labels. The designer thinks that the system image is the same as his design model [57], which is not necessarily true. Now someone uses the system (blue person). He or she builds a mental model (red triangle) by using the system. "The mental model of a device is formed largely by

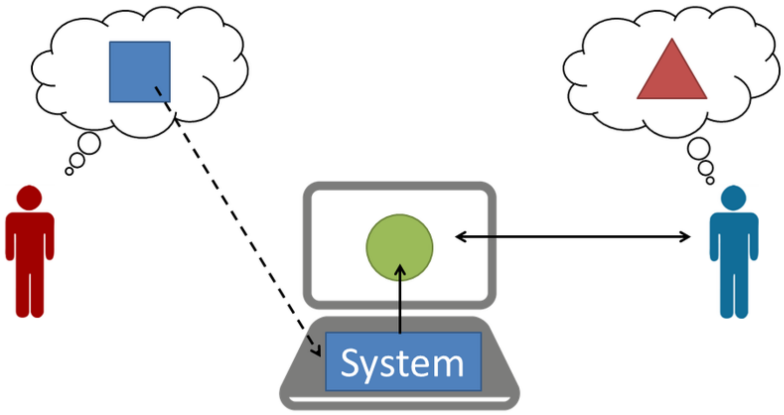


Figure 2.25: *Visualization of conceptual models*

interpreting its perceived actions and its visible structure" ([57] p. 16). If the system image does not make the design model clear, then the user's mental model will differ from the design model. This is according to [58] the main reason for usage errors.

2.3.5 Measuring usability and user experience

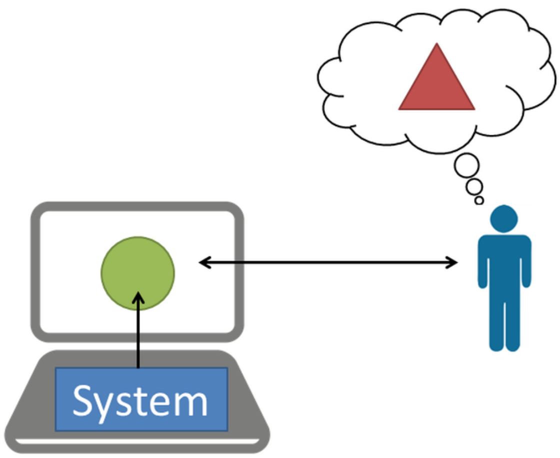


Figure 2.26: *Visualization of usability measuring*

According to the concept of conceptual models described above, the system image can differ from the mental model of the user. This difference depicted in Figure 2.26 could serve as a benchmark for usability. But those models can not be measured directly, thus a lot of research has been dedicated to measurement of usability and user experience (UX).

Some of the methods collect quantitative data, which is defined as "numeric data, such as user performance metrics or opinion ratings" ([54] p. 429). Others collect qualitative data, defined as "non-numeric and descriptive data, usually describing a

UX problem or issue observed or experienced during usage" ([54] p. 429), which can be just as informative as quantitative data. According to [54] those methods can be distinguished further between formative evaluation (evaluation and improvement of the process) and summative evaluation (evaluation of the result).

There is a huge number of evaluation methods. For a good summary of summative evaluation methods see [66]. So-called "informal summative evaluation" and formative evaluation methods are covered in [54], which provides an engineering approach.

Nevertheless, I describe three classical usability measurements methods briefly: Heuristic Evaluation, Keystroke-Level Modeling and Usability Testing.

- With **Heuristic Evaluation**, problem factors concerning effectivity, efficiency and intuitivity shall be estimated. An commonly used approach are the Ten Heuristics by Nielsen [67]. Several usability evaluators explore the software and classify the perceived problems according to those principles (like fault prevention or visibility of system state). The errors are rated in five categories from 0 (no problem for usability, rather cosmetic) to 4 (usability disaster, which should be fixed before release).

- The **Keystroke-Level Modeling** captures predefined times for specific phases of human machine interaction (like pressing a button, realizing a new loaded system screen, move finger to new button, and so on) [68]. The time for a mechanic keystroke is for example 0.39 s [69]. As every use case has a ideal path, this method allows for an estimation of the duration of each task.

- A **Usability Test** measures effectivity, efficiency and contentment (ISO-9241-110 [70]) with a sufficient amount of probands in order to state statistical assertions. The exact study design depends on the software and the tasks evaluated. For example, effectivity can be measured with the number of solved part tasks, efficiency can be measured with the required time and the individually experienced task load (measured with questionnaires), and contentment can be measured with questionnaires like the ISONORM questionnaire [71], and intuitive use can be measured with the QUESI questionnaire [72].

Also, Human Robot Interaction (HRI) research shows the importance of human factors for HRI like for HCI. [73] provides an approach how heuristic evaluation can be adapted to robots. It is very important to involve the user in the whole design process. Especially the final evaluation has to be performed with end users, who will use similar systems in the future. Due to the complexity, HRI systems are very difficult to evaluate and to be compared quantitatively. Human factors research provides a good background and models to explain different effects of systems, their elements and functionalities on human robot interaction. [74] suggests a summarized set of common metrics for HRI, while the focus lies on task-oriented mobile robots.

2.3.6 The user experience process

As described above, a high quality user experience can be ensured by applying a consequent process as described in [54]. The described "UX cycle" (see Figure 2.27) mainly consists of the steps analysis, design, prototype and evaluation, while the repeating of one step or jumping back to the last step is encouraged.

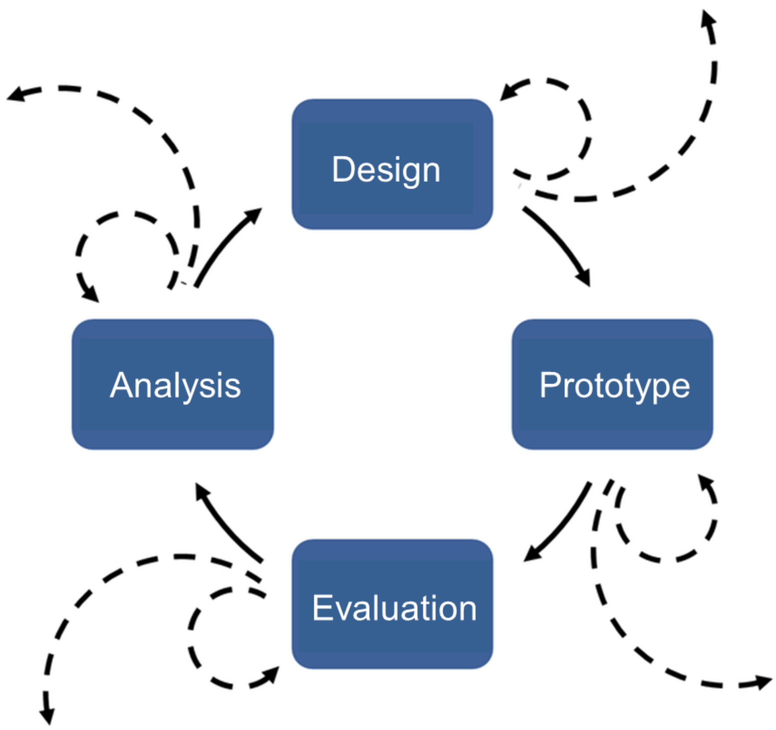


Figure 2.27: *The user experience development cycle (following [54])*

In Figure 2.28 the design process is visualized with different important steps, although there is the possibility for an iterative procedure as described in the "UX cycle". The first step is contextual inquiry, a early system activity "to gather detailed descriptions of customer or user work practice for the purpose of understanding work activities and underlying rationale. [...] Contextual inquiry includes both interviews of customers and users and observations of work practice occurring in its real-world context." ([54] p.89). It is followed by contextual analysis, "a systematic analysis - identification, sorting, organization, interpretation, consolidation, and communication - of the contextual user work activity data gathered in contextual inquiry [...]" ([54] p.129). In Figure 2.28, the raw data from contextual inquiry is interpreted in work activity notes. Those are "used to document a single point about a single concept, topic, or issue as synthesized from the raw contextual data." ([54] p. 131). Afterwards, they are clustered in a so-called work activity affinity diagram (WAAD) and are used to extract software requirements and design-informing models. The latter are "design-oriented constructs, such as task descriptions or user personas, that turn raw data into actionable items as design ideas, as elements to consider or take into account in the design." ([54] p.183). There are different kinds of design informing models like flow models (overview of work process), user models (different work roles or personas) or social models (the relations between different work roles). Depending on the task and complexity of the work domain, different models can be chosen and should be generated in team work and iterated with domain experts, if possible (for a detailed description see [54]).

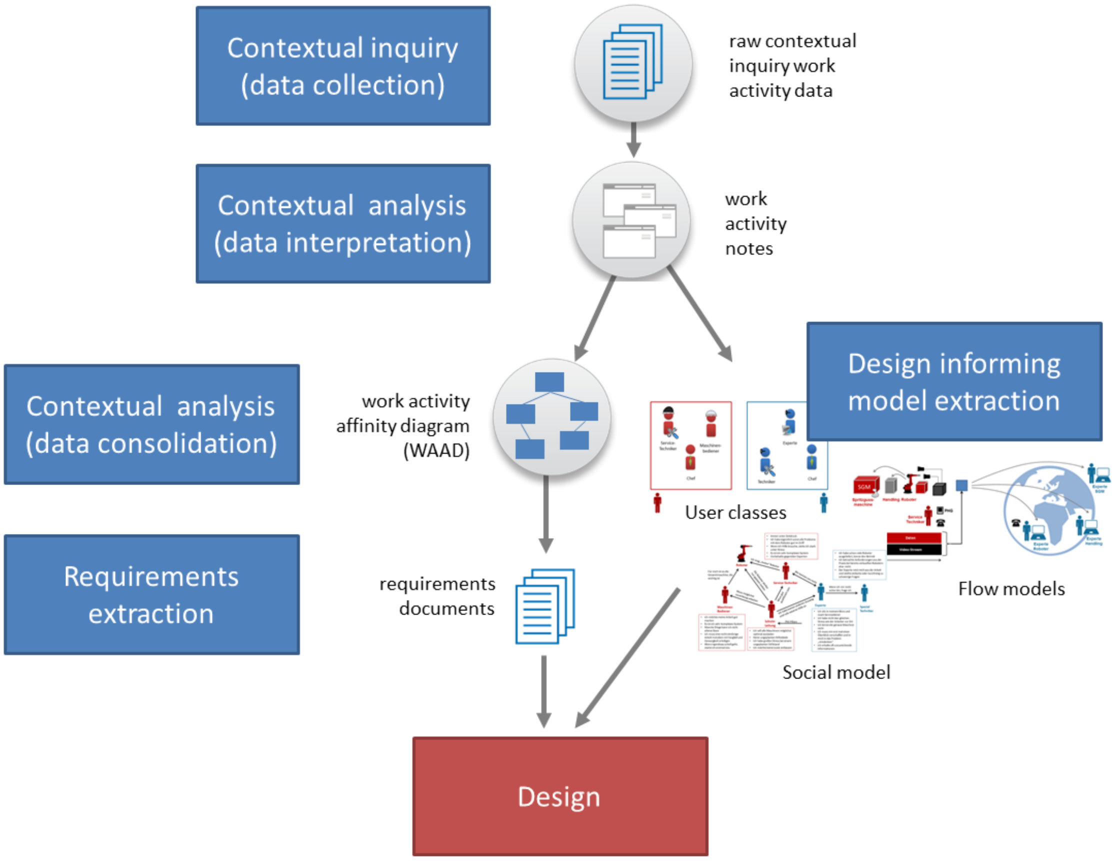


Figure 2.28: *The user experience development process (according to [54])*

The most important role of the models and the detailed inquiry is to depict the actual state of the work environment and to reason about an envisioned future state which shall be reached with the new product. This is also called the "design gap" [54] which can be bridged with the methods described above.

The design process can be categorized in the steps "Ideation and sketching", "Conceptual design", "Intermediate design", "Detailed design" and "Design refinement" [54]. Some steps overlap in practice, while different mediums like sketches, storyboards, illustrated scenarios, wireframes and visual comps are used. They are evaluated in the team at first, but later also with end users.

2.4 H2H communication

On base of the considerations about mental processes above, the human-to-human communication will be analyzed in this section. In many cases a telematics system is used by two or more humans in order to work together via a network. Thus, the dynamics between humans will be of importance for the development of such an application, as the cooperation can be systematically supported by the interfaces. When such principles are disregarded the application can also hinder high quality cooperation.

2.4.1 Computer supported collaborative work

The research field associated with human-to-human communication via network is called "Computer Supported Cooperative Work" (CSCW). There are numerous definitions of this term. We use the following definition translated from [75]: "CSCW is research on possibilities and effects of technological support for humans to work and communicate in groups and work processes".

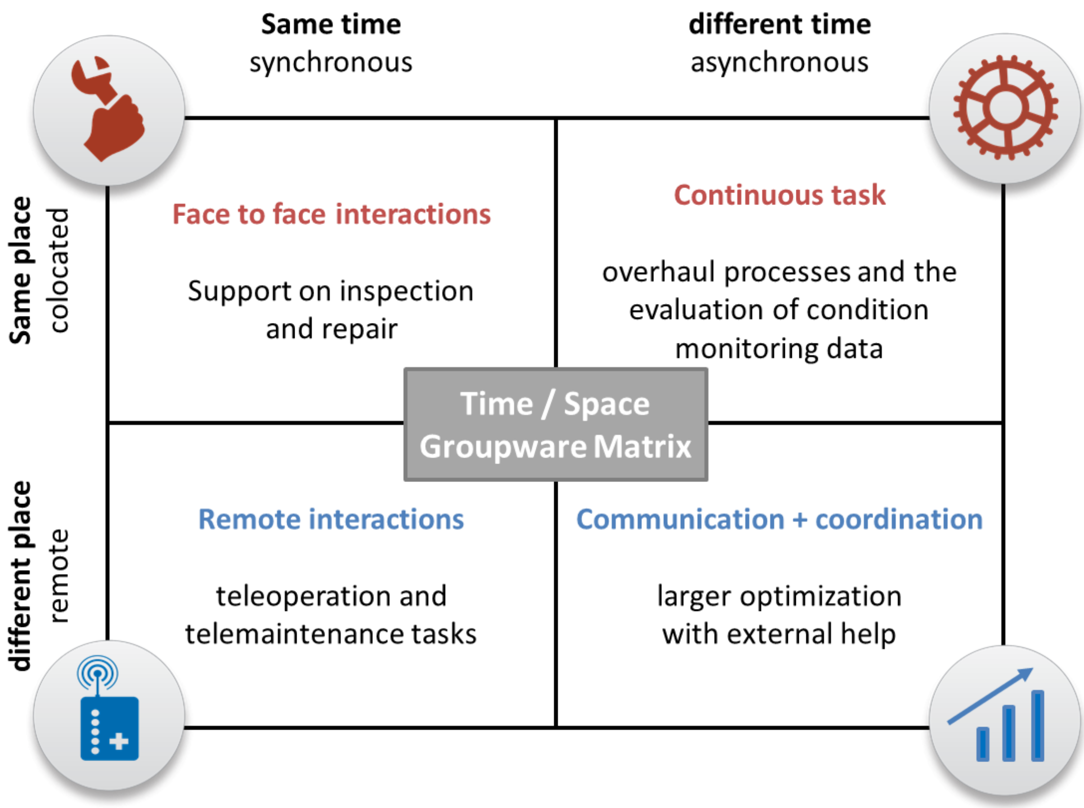


Figure 2.29: *Time-space matrix for CSCW (following [76])*

CSCW systems can be characterized according to Figure 2.29 on the two axes time and space. The systems in the depicted sections have different requirements.

We try to allocate the associated maintenance tasks: colocated and synchronous tasks are the support on inspection and repair. Asynchronous, colocated tasks can be the discussion on overhaul processes and the evaluation of condition monitoring data on a longer time period. The classical telematics applications are designed for cooperation at different places: The teleoperation and telemaintenance tasks take place in synchronous time, while larger optimization with external help is performed asynchronously.

2.4.2 Communication

The Shannon-Weaver model mentioned above (see Section 2.2.1.4), does not handle the "sense" of a communication, as we understand a communication between people. Social interaction always happens, when people are in purposeful and meaningful contact with other people. The communication psychology (e.g. [77]) expands the Shannon-Weaver model to social interaction between people. The encoded message between sender and receiver is the verbal and nonverbal communication between two persons. It can be altered by interferences. One can simply put it, that communication fails, if both sides do not use the same encoding.

A newer communication theory [78] states that a message contains four sides: fact, self-revealing, relationship, and appeal. If the receiver listens "on the wrong ear" he or she might misinterpret the message. Figure 2.30 visualizes this theory. Based on its mental model of the blue box (see Section 2.3.3), a sender encodes a message. Due to interferences, the receiver cannot decode the message correctly – he or she gets a mental model out of the message which does not match with his or her mental model of the blue box (which is again different from the sender's model). The possible interferences increase, if they both use a telematic connection. So this considerations are vital for the case of computer supported collaborative work we cover in telemaintenance scenarios.

2.4.3 Common ground and grounding

As described above, people have mental models of objects, circumstances and dependencies. They also use words and symbols to describe the objects, which are connected to the mental models ("semiotic triangle" [65]). Naturally, people do not have the same mental models and also the understanding of words and symbols differ, i.e. in different languages. In order to speak about the same object, humans need to have a common understanding of the terms, also called "common ground" [79].

Common ground is needed for any coordination of a process between independent entities. In Figure 2.31 both humans have slightly different mental models, but they use the same terms in order to speak about the object. But common ground also means a mutual knowledge, mutual beliefs, and mutual assumptions. As the mental models and the connected words are formed by education and society, people with the same cultural background have similar connections.

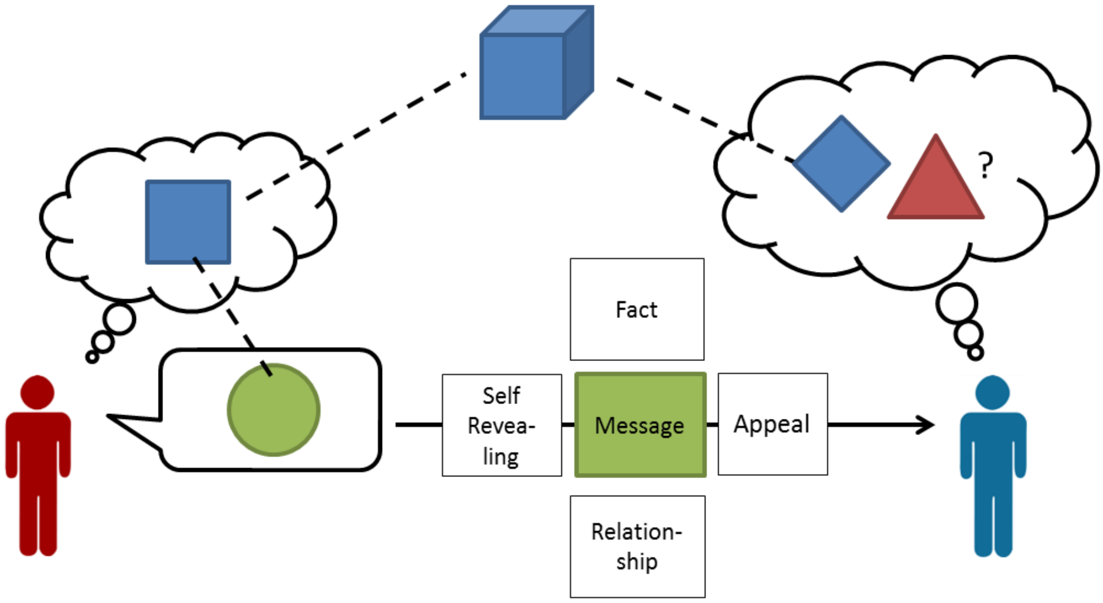


Figure 2.30: *Model for H2H communication based on [78]*

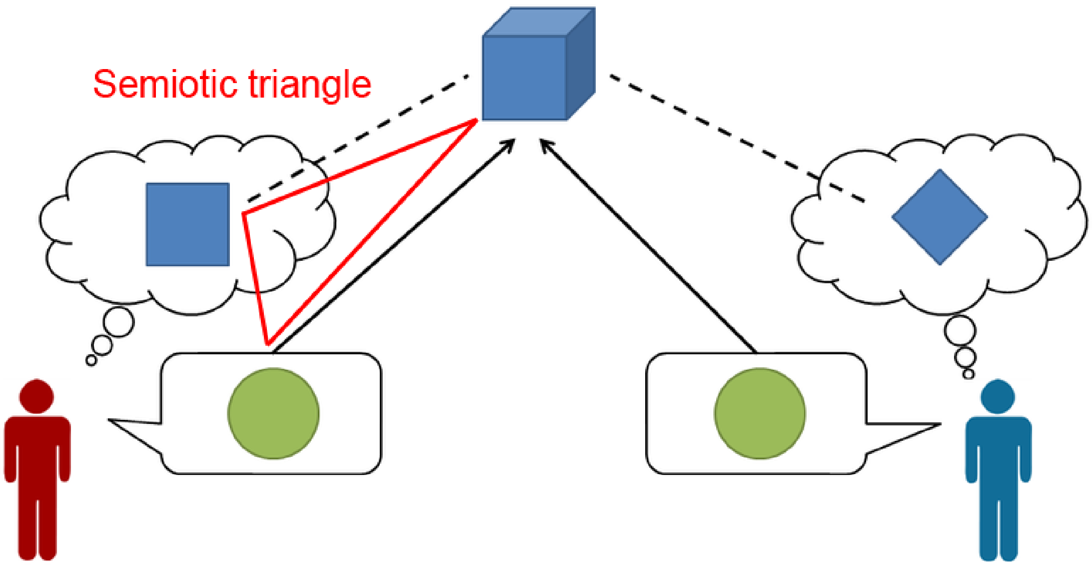


Figure 2.31: *Visualization of common ground*

When two people from different backgrounds interact, they need to update their common ground. This process is called "grounding process". The entities try to reach a mutual belief on the situation and the objects. This is especially important, if those two persons need to work together. In order to coordinate their tasks, they need to actualize their mutual knowledge constantly [80]. This can be expressed in two phases (following [81]):

1. **Representation phase**: Person A gives a remark to Person B and expects B to reacts on that remark. If a reaction takes place, A assumes, that the remark has been understood.

2. Acceptance phase: Person B accepts the remark and replies with an implication. If Person A accepts the implication, B can assume, that he or she has understood the original remark correctly.

The specific kind of grounding process depends on the context of the conversation. But two factors are essential to understand grounding:

- **Purpose:** What are the communication partners trying to achieve with the conversation?

- **Medium:** The techniques presented with a communication medium and the cost involved.

Especially the choice of the communication medium is an important choice for the designer of a telematic system. An experimental study with different communication models and their effect on the cooperation quality has been carried out by [82] and has been extended by [81]. They found different constraints that a medium may impose on a grounding process between to people: Copresence, visibility, audibility, cotemporality, simultaneity, sequentiality, reviewability, and revisability. "When a medium lacks one of these characteristics, it generally forces people to use alternative grounding techniques" ([81] p. 230). There are also different costs resulting from this absence for speakers and addressees. For the case of telemaintenance presented in this work, this means for example: A telematic connection cannot generate real copresence. It therefore needs to amplify the other factors systematically.

2.5 M2H communication

When we discuss the "machine-to-human" communication, it covers mainly the question of information visualization and display. In order to optimize this for human perception, we need to cover human information processing as well as visualization possibilities like Augmented Reality.

2.5.1 Human perception and reaction

The socio-technical system (compare Figure 2.21) shows the human side as information input, processing and output. But to be precise, the human side is more complex than that. Figure 2.32 shows a more detailed version of the information processing on the human side.

Starting with the sensor function on the left of Figure 2.32, the human eye receives beams of light. But only with the complex perception process [83], these stimuli can be compared with memorized patterns and models. In order to make this happen, the sensory short-term storage needs to contain a precise and complete picture of the

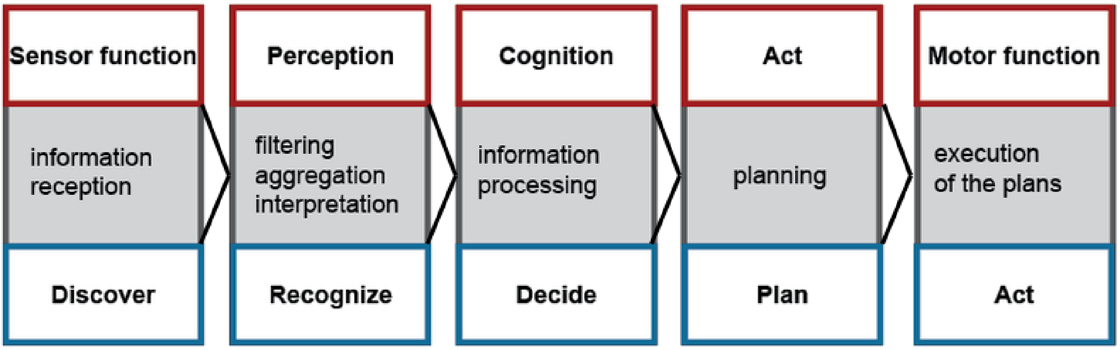


Figure 2.32: *Human perception adapted from [58]*

environment for the actual problem [84]. The human then plans his or her actions and executes them with the use of the motor functions (talking, pressing a button, and so on).

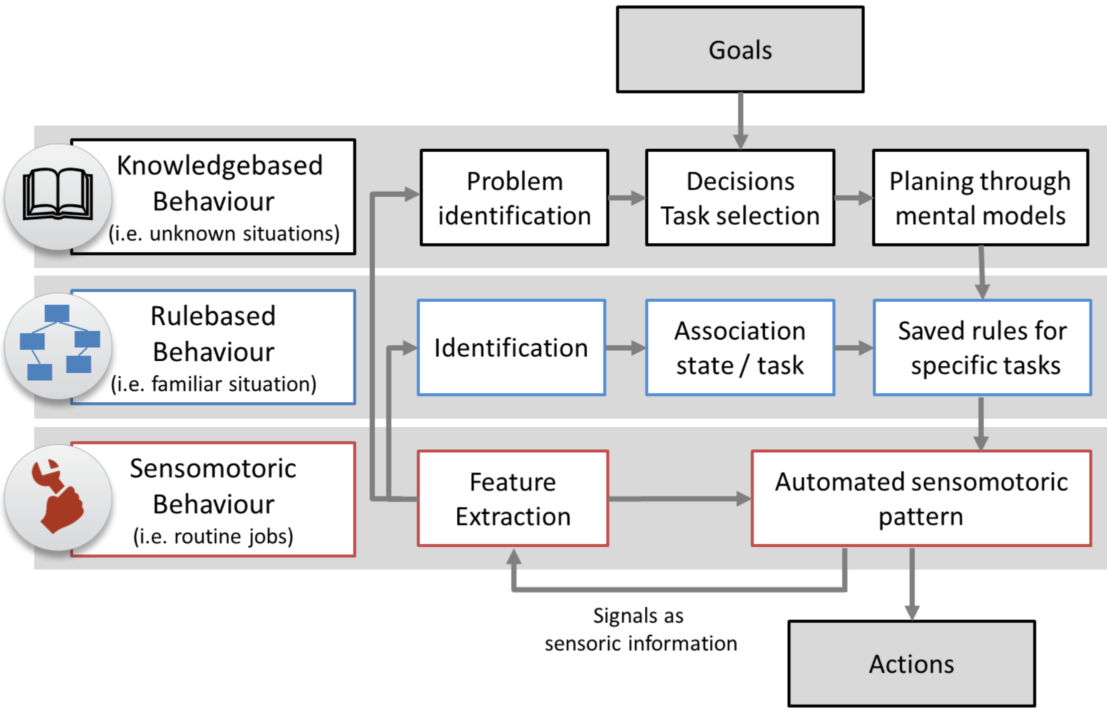


Figure 2.33: *Regulation layers for human behaviour from [58]*

When we look deeper into the cognition part, there have been different models that describe the process of reasoning and action. We use the model by Rasmussen [85] displayed in Figure 2.33, which has been translated for an industrial context [58]. The model describes a multi-layer architecture. Control functions in normal situations and simple problem solving strategies like pushing the emergency stop button are located on the sensor-motoric behavior level (lowest layer). The rule-based behavior level (second layer) covers learned activities. The most decision and problem solving activities needed in the industrial domain are located in this level. Only if no suitable rule exists, the human needs to use the knowledge based behavior level (upmost layer).

In critical situations, humans need to perform the right rule-based or knowledge-based behavior. But due to high information demand, high information supply and stress, the user might act stereotypically on low behavior levels. This can be omitted by a computer system which provides exactly the right information combined with simulation training with fix "ready-to-use" rules. But not all situations can be trained specifically. Systems optimized for routine operation may be very poorly suited for dealing with system failures, as they require knowledge-based behavior. Especially the higher behavior levels rather need a training of problem solving strategies, which can also be supported with software (compare [58]).

2.5.2 Situation Awareness

As described above, the requirement for a proper perception of the situation is sufficient sensor information. It has already been mentioned, that more data does not automatically mean more information. "The problem with today's systems is not a lack of information, but finding what is needed when it is needed" ([86] p.4).

An example for this problem gives Figure 2.34. Although a lot of information is provided on the left side, the information can be recognized in an ordered form by the red human through direct observation. The small checkmarks indicate that the person thinks he or she has enough information. The displayed thought balloons contain a "situation model" [87], which is a "parameterized mental model" ([86] p. 277) or "one's situation awareness and one's understanding of a specific situation" ([86] p. 277).

The ability to get information about a situation is especially important, if the person is not standing near the objects, but perceives them via a telematics system, as the blue person in Figure 2.34 does. A major challenge is to provide sufficient information through a remote interface to compensate for the cues perceived directly. The figure also visualizes the source of the knowledge for the remote user. The operator gains information through the system knowledge which is expressed in the interface knowledge. Thus, the operator builds his or her situation model out of the conceptual model of the interface. He or she is also able to gain additional information from other people.

A general definition of situation awareness describes it as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future" ([88] p. 97). This already indicates three levels of the situation awareness: The perception (Level 1), the comprehension (Level 2) and the projection (Level 3).

It is possible to measure situation awareness with numerous methods (for detailed information see [66]) including standardized questionnaires like SART (Situation Awareness Rating Technique) [89]. The relation between situation awareness and

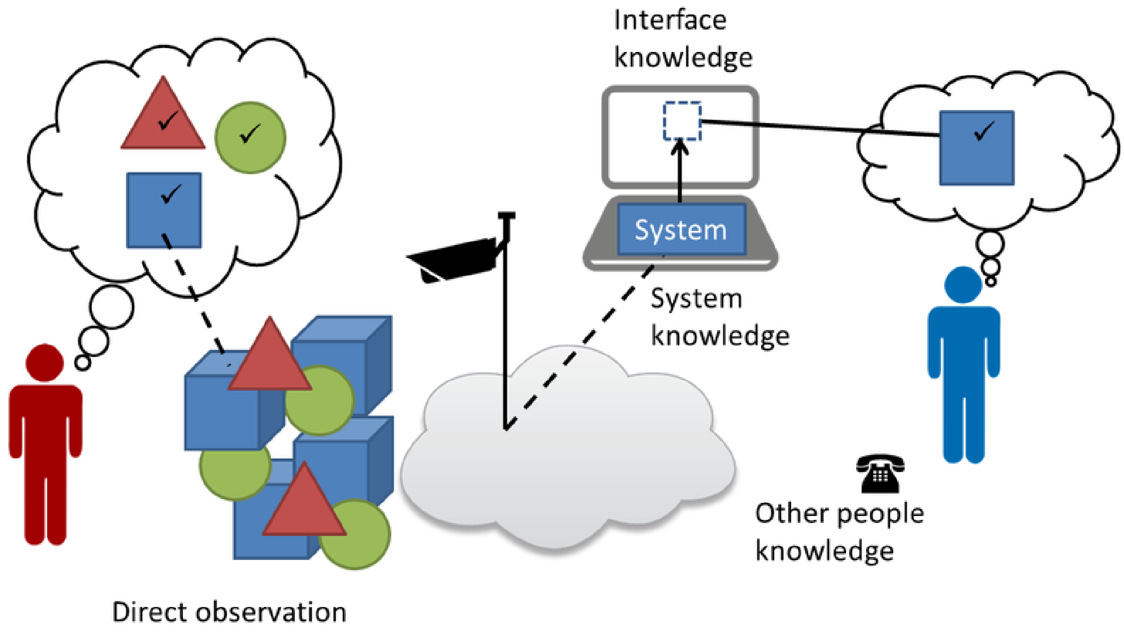


Figure 2.34: *Situation Awareness*

workload is important, too. As performance and situation awareness are only linked probabilistically, there is no threshold of situation awareness that guarantees a specific performance level [86]. But it is possible to measure differences between direct observation and different system designs.

Situation awareness can play an important role in a group of people. Collective situation awareness can be seen as the intersection of the individual situation awareness of each group member. [90] shows the importance of situation awareness in maintenance teams.

2.5.3 Spatial notation

In order to describe movement in space, we use a Euclidian 3D space together with the three dimensions of angular orientation (roll, pitch, yaw) depicted in Figure 2.35. The specifications of a system's parameters within its own coordinate space defines the system's frame of reference.

Transformations $f : F \mapsto M$, $p_f \mapsto p_m$ can be used to map one coordinate system to another and can be displayed as transformation matrices ${}^F T_M$ with

$$p_m =^F T_M \cdot p_f$$

There are two basic types of transformations: translations and rotations.

There is the generic translation matrix with offset $p = (x, y, z)$ and the three fundamental rotation matrices around the three coordination axes (x, y and z). Composite transformations consist of translations and rotations which are performed successively.

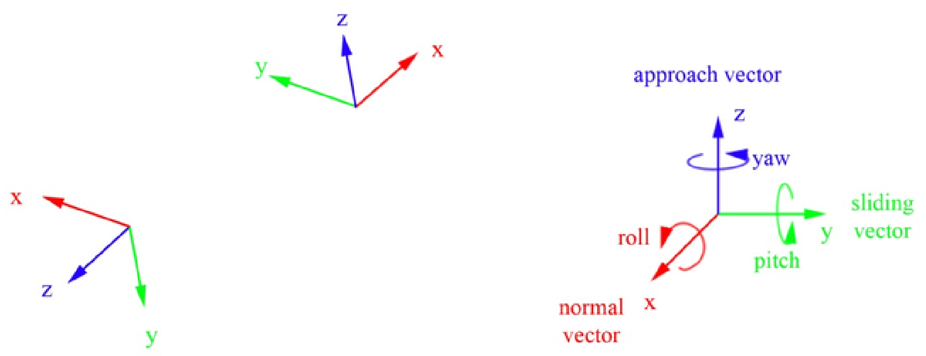


Figure 2.35: *Spatial coordinate system*

$$Trans(p) = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad Rot(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\theta & -S\theta & 0 \\ 0 & S\theta & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rot(y, \theta) = \begin{bmatrix} C\theta & 0 & S\theta & 0 \\ 0 & 1 & 0 & 0 \\ -S\theta & 0 & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad Rot(z, \theta) = \begin{bmatrix} C\theta & -S\theta & 0 & 0 \\ S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.5.4 Visuospatial thinking

Humans usually do not think along the Euclidian space, as this is an artificial external representation. Intuitively, humans use their own body and the space surrounding the body as reference. Human tend to have mental model about the world around them. For navigation, they use for example wayfinding relative to landmarks [91]. By using those, humans demonstrate different spatial abilities like spatial visualization and spatial memory [91].

If we try to enhance situation awareness on a situation, where spatial dimensions between objects matter, we need to keep in mind, that humans use different coordinate systems. For a proper spatial cognition, they need to translate between their internal coordinate system and the coordinate system presented in the application with mental transformations. Following [91] and Figure 2.36 we distinguish between the following six basic frames of reference:

- World frame. This is the global frame of reference in which the operation occurs.

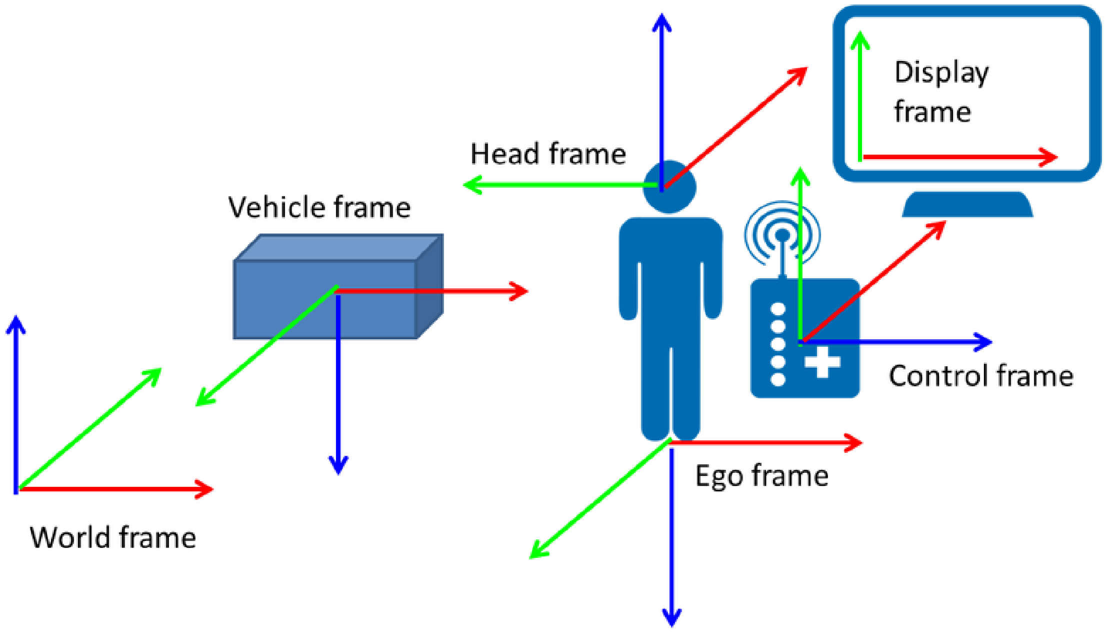


Figure 2.36: *Frames of reference*

- Ego frame. Typically the location of the observer/operator.
- Head frame. The orientation and location of the operator's head with respect to the trunk.
- Vehicle frame. The frame associated with an operated vehicle.
- Display frame. The frame of reference of information on a display.
- Control frame. The frame of reference of control input devices.

It can easily be reasoned, that a huge number of possible configurations between those frames of reference is possible, even complicated by the fact that each frame of reference can move in all of its six degrees. Every task carried out in the system must be accomplished by a transformation in the human's head. This is called visual spatial thinking [91].

The goal is to setup a user interface with the demanded information while minimizing the needs for mental transformations, which have different costs dependent on the task. For example mental lateral rotations beyond 45 degrees have higher cognitive demands than translations [91].

In order to understand the symptomatic between this, we regard the possible viewpoints for airplane teleoperation depicted in Figure 2.37 on the upper half and the schematic representation of the perspective the supervisor would see on the lower half. In the first picture (a) the pilot's direct view out of the cockpit, an egocentric perspective, is chosen. This needs minimal transformation for the supervisor, but the amount of information does not suffice. In the second picture (b) the view

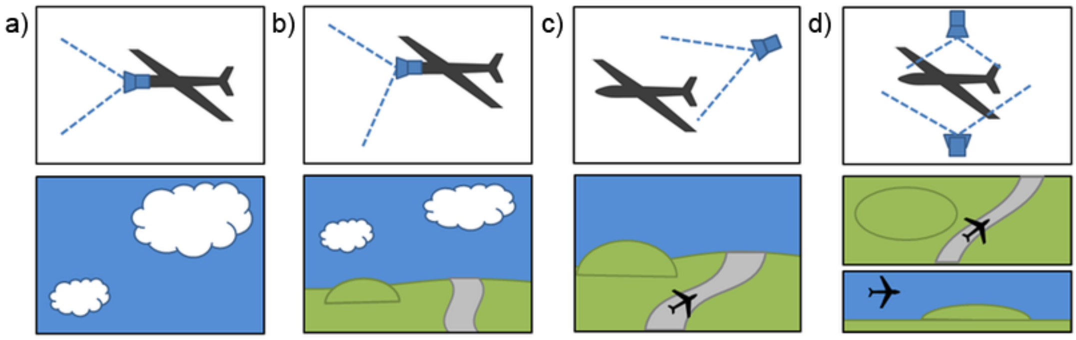


Figure 2.37: *Possible viewpoints and interfaces adapted from [92]*

covers a greater angle which requires only a modest mental transformation. The same applies for a 3D exocentric or "tethered" viewpoint in the third picture (c). The 2D coplanar view with a top lateral view and a vertical profile view covers all information but requires two pictures and a large amount of mental transformation.

	3D		2D
	Immersed (b)	Tethered (c)	Coplanar (d)
Cost of scanning	Low	Low	High
Cost of Cognitive Integration Across Planes (Axis Pairs)	Low	Low	High
Principle of Pictorial Realism	Confirmed	Confirmed	Violated
Line of Sight Ambiguity	Cost	Double cost	
Keyhole View	Cost		

Figure 2.38: *Cost and Benefits of Different Display Perspectives adapted from [93]*

There has been a huge research interest in the best way of representing information in an airplane cockpit. Figure 2.38 summarizes the results of several studies of transformation costs. It can be summarized following [91] as follows: The more natural a viewpoint is (like Figure 2.37a or b) the better it can support control tasks like navigational control or tracking. If an exocentric perspective is chosen, it is even more difficult to decide, if the mountain in (c) is in front or below the airplane (line of sight ambiguity). For spatial judgement tasks the coplanar display (d) is superior because of the absence of that line-of-sight ambiguity. Yet the 3D exocentric viewpoint (c) can best support the situation awareness as it avoids the mentally demanding integration of information out of two views (d) and can see a larger angle like objects near the airplane (a and b only allow a narrow angle = "keyhole cost").

When several viewpoints are provided in a cockpit, the different viewpoints can compensate the shortcomings of the others. But on the one hand this requires a mental integration into one "large picture". On the other hand, research has shown that the immersed viewpoint is very compelling and may attract attention away from other viewpoints [94].

In order to address the mental integration problem, the principle of "visual momentum" has been presented: It means e.g. to highlight a common element in both (or all) display viewpoints and to highlight overlapping areas. This can improve various aspects of dual map performance [95].

When we come to moving reference frames, it is important to mention the principle of the moving part [96]. This principle asserts that the moving element on a display should correspond to (and move in the same direction as) the user's mental model of what moves in the real world [91]. But this may cause conflicts with other principles like the use of ego-referenced frames on a map: a right turn of the vehicle will trigger a left rotation of the map. According to [91] the principle of the moving part becomes more important the more rapid the speed of movement is.

2.5.5 Information space

In the opposite to the visualization tasks of the last sections, displays often do not represent real space but information. This involves the new research field data visualization, the idea of providing "the information behind the data" or "the structure of the data" (see [97] p 6.).

In this context of M2H communication, researchers [91] talk about the "information space". There are different kinds of information structures displayed in Figure 2.39: (a) a hierarchical relation, (b) a network relation with possible multiple links, (c) a matrix organization, and (d) a true Euclidian space (like graphs). The same information can be displayed in each space, but will not have the same effect in every space - different tasks require a different representation.

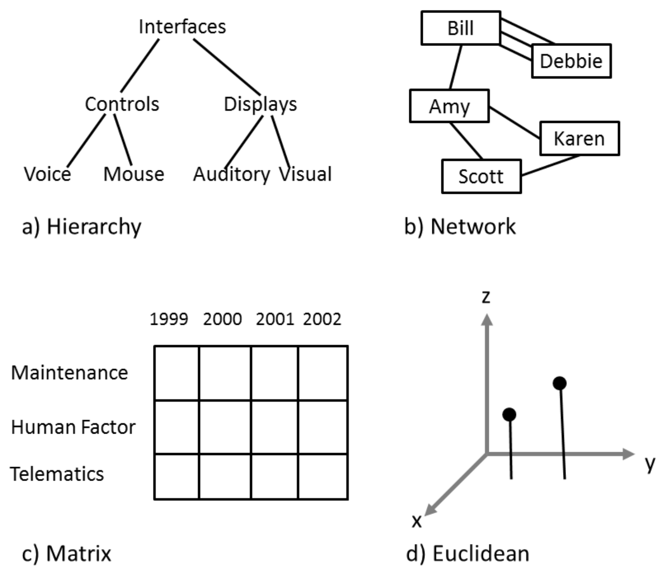


Figure 2.39: *Information spaces adapted from [98]*

There are different tasks which can be carried out on the representation of the information structure, like "retrieving an entity", "retrieve information about a node", "compare nodes" or "to understand the overall structure of the database" [91]. A good graphic is always "part of a larger whole, the context, which provides its relevance" (see [97] p. 58). The task defines the most appropriate visualization method.

A systematic characterization of different visualization tasks in a "design space" was performed [99] along the dimensions goal, means, characteristics, target and cardinality. Consistent with [97] the most important differentiation is the goal. The exploratory analysis tries to derive a hypothesis from an unknown dataset, the confirmatory analysis aims to test a hypothesis about a dataset and the presentation exhibits confirmed analysis results [99].

The momentary focus of attention in the information space can be regarded as "ego/icon" consistent to the considerations of the last section. More exocentric views support a better understanding of the structure of the information space and allow a more rapid location of items within the space. Also the keyhole view can lead to the phenomenon of getting lost [91].

As the application of this work is data from industrial plants, theoretically it would be possible to visualize it as a table or matrix. But although people are able to get single point values more accurately and quickly out of tables, graphs are easier to remember and allow better information integration like trend prediction [91]. In the following sections, we will focus on quantitative information displayed in graphs.

There are several psychological models that try to describe the understanding of graphs [91]. They all have in common, that they distinguish between the perceptual and conceptual processes and predict five factors to play a role: the display characteristics, the data complexity, the viewer's task, the viewer's prior knowledge and the viewer's graph-reading expertise [91].

2.5.5.1 Different graph types

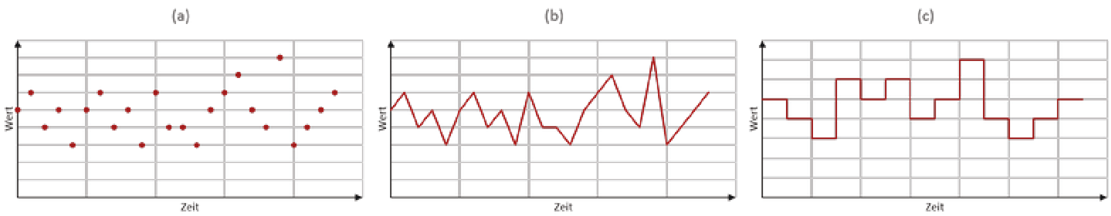


Figure 2.40: *x-y visualization of data in (a) scatterplot, (b) line graph (c) step chart*

Information is perceived very differently in various displays. If the data is displayed in a scatterplot (Fig 2.40 a), individual values can be identified accurately. But trends can be better identified with the help of line graphs, as the human always tries to interpret line graphs as trends [91]. Simply connecting data points linearly physically and mathematically not correct, because the data points mark sampling points of values which might keep changing between the samples (compare Section 2.2.1.3). A step-line graph can be used to visualize the time discrete sampling.

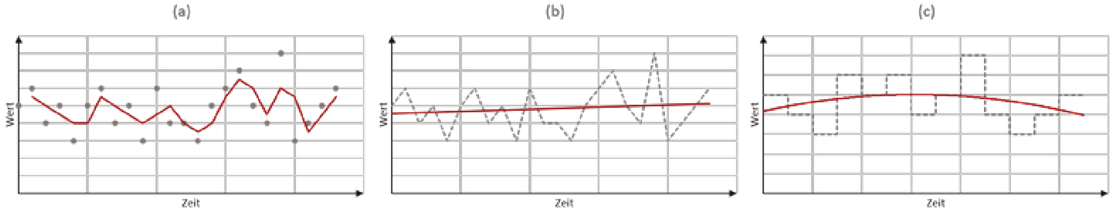


Figure 2.41: *x-y data with trend curve (a) scatterplot with fitted LOESS [100] curve, (b) line graph linear trend line, (c) step chart with second grade polynomial regression trend*

The human tendency of finding trends in data can be assisted by adding a trend curve with mathematically computed values. The goal of this curve fitting is to find a curve or mathematical function which represents the measured values best, possibly subject to constraints. For an exact fit, interpolation methods are used, for a trend visualization smoothing techniques are used. Figure 2.41 provides several possibilities. A function most commonly implemented in statistical toolboxes is the LOESS [100] method, which uses a polynomial fit to the data using weighted linear squares.

For proportional comparison, typically pie charts are used, as displayed in Figure 2.42. Values are displayed as angles that sum up to 360 degrees. Similar is the "donut" chart displayed in Figure 2.42 (b), in which not the whole circle but only a circle outline is used. Both displays work only for a small data amount, as it is pointed out by Figure 2.42 (c).

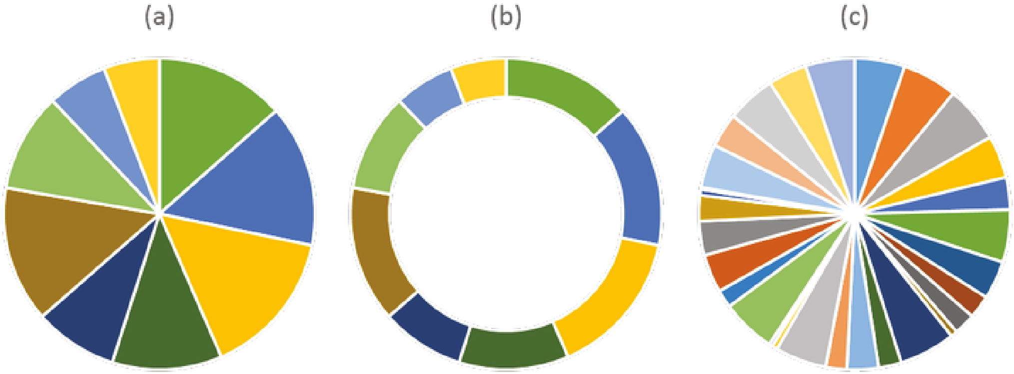


Figure 2.42: *Circle displays (a) pie chart, (b) "donut" chart, (c) pie chart with too many values*

Another possibility for the display of x-y data is the bar chart, which does not suffer from the limitation regarding the amount of data. As depicted in Figure 2.43 the bar charts can be used to display a time context or a proportional relation.

There has been a huge research interest on the question which display method works better. This means to investigate with which display the human can perceive and understand the information more quickly and more accurate. Then the following summary of several studies is provided by [91]: As line graphs are always interpreted as trends by the human brain, they serve better if an integrative task is demanded. Bar graphs provide a more separable format, thus they are better for discrete comparison or point reading. This is formulated in the proximity compatibility principle. When it comes to pie charts, they are, in general, more accurate for making part/whole judgements than divided bar graphs like Figure 2.43 (b). But if the goal is to communicate the proportions and not the accurate value, the divided bar charts may be better. The line graphs in Figure 2.43 (c) fail to communicate the proportions but can display the trends of the particular value.

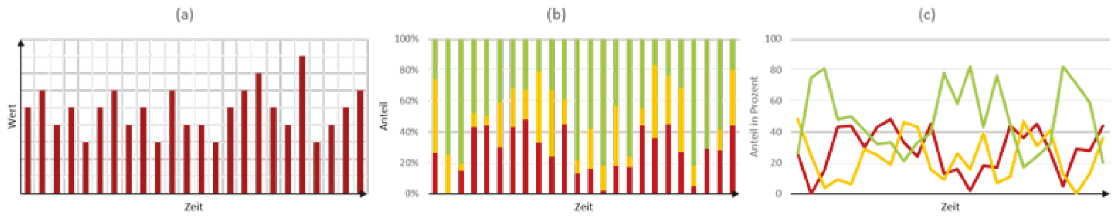


Figure 2.43: *Bar graphs in comparison: (a) bar graph, (b) divided bar graph, (c) same values with line graphs*

All of the graphs presented in this chapter have used colors. Studies confirmed that color is often interpreted as categorical and provides helpful cues, especially by helping to keep track of quantitative relationships. But color cannot accurately represent precise quantitative information [91].

There is a tendency to add a third dimension to a bar or pie chart. Although these measures were found to improve visual interest, those additional noninformative features are a distraction for the reader and may lead to errors [91]. Only if the task requires the integration of information of all three dimensions, three-dimensional displays has proven to be better. But the 2D projection of the 3D relationships can degrade or occlude information [91]. A similar finding has been made about animated data: There are potential benefits but the perception of the information behind the data is often difficult and error-prone [91].

As mentioned above, quantity can be estimated best by the human brain if it is displayed in height and length. Other visualization methods are possible, but may lack accuracy. Nevertheless they may serve good for other tasks than value estimation.

2.5.5.2 Visualization of complex (big) data

Especially when it comes to huge amount of complex data, human users are overstrained. That is the reason for a cautious user interface development. The problem is, that the models of simple graph comprehension provided in the last section "may not scale up to account for more complex data sets and may not provide useful guidelines for the development of new information visualization tools." ([91] Chapter 11). As data complexity increases, the time and errors on simple fact-retrieval tasks increase. But the tasks carried out in the "Big Data" domain are usually way more complex than fact retrieval and the tasks have a huge influence on the effectivity of visualization [91]. A survey on data analysts [101] clearly states that there is definitely a special demand for better user interfaces in the area of Big Data.

The exploration of large data sets is an important but difficult problem. Information techniques may help to solve it. Integration of visualization techniques would combine fast automatic data mining algorithms with the intuitive power of the human mind, improving the quality and speed of the visual data mining process. Visual Data Exploration usually follows a three step process: Overview first, zoom and filter, and then details-on-demand (which has been called the Information Seeking Mantra). Visualization technology may be used for all three steps.

For a statistical view on a huge amount of data, the distribution function and the density function can be displayed in order to detect patterns of known distributions like Gaussian distribution, Exponential distribution or Gamma distribution. A researcher states the hypothesis that the dataset follows a specific distribution and performs hypothesis tests.

1. Computing from the observations the observed value t_{obs} of the test statistic T.

2. Calculating the p-value. It is a function of the observed sample results (a test statistic) relative to a statistical model, which measures how extreme the observation is.

3. Rejecting the null hypothesis, in favor of the alternative hypothesis, if and only if the p-value is less than the significance level (the selected probability) threshold.

There is a lot of research on user interfaces for big data sets. One example is VisTrails [102], which visualizes a large amount of scientific data. It also stores and displays the history of the data. The development of database interfaces can be compared to the evolution of operation systems: They have been operated via consoles but are now providing a graphical interface [103]. It can be distinguished between atomic visualization, aggregated visualization and density plots.

2.5.6 Augmented and Virtual Reality

As described in [104], humans use visualization and drawings since primitive times in order to convey ideas and objects to other people. As society has advanced, more realistic visualizations have been evolved, so that 3D models are used in many scientific areas. Next to anatomic visualization of human bodyparts [105], visualization of machine parts [106], maps of caves or terrain, 3D models can be used in schools [107]. All application areas show that three-dimensional displays have advantages compared to two-dimensional displays. In industrial automation many processes can be optimized with interactive 3D models [104] presented in Virtual and Augmented Reality.

2.5.6.1 Virtual Reality

The first definitions of Virtual Reality (VR) refers to electronic simulation of environments as well as to the hardware used to perceived this world [108]. After a while, hardware as characteristic for VR has been replaced by presence in the simulation. The perception of virtual worlds happens with the help of nearly all sensory organs, allowing the feeling of presence in another world. To distinguish between the real world of the user and the virtual world, the term telepresence is used to describe the perception in the virtual world. The definition of Virtual Reality on base of telepresence illustrates the independence of hardware [108]. It also clarifies, that the analysis object of VR is the individual, who perceives the world. Because individual experience causes Virtual Reality, the development of virtual world needs to consider clarity and interactivity [108], [109].

The perception of virtual objects has been enabled by the long lasting development of computer graphics. Different functions can be displayed in a three-dimensional space. Mostly, virtual objects are displayed with polygons, while the planar polygons approximate the desired object surface. But there are also other approaches like the use of bi-cubic parametric patches or constructed solid geometry [110], [111].

The 3D scene must be rendered from a camera view with ray tracing methods [112]. While interacting with objects, it is important to implement, that the user means the foremost object, because he can only see that. There are several 3D development environments, that use the state of the art methods for 3D image display, rendering and interaction, because most games play in a virtual 3D nowadays.

2.5.6.2 Augmented Reality

While the term "Virtual Reality" (VR) is already widely known, the demarcation of "Augmented Reality" (AR) remains still unclear for many people, especially in the area of industrial applications [113], although it has been used since 1994. AR enables the enhancement of normal perception with artificial information, while VR is a complete artificial construction. Consequently there is a "reality-virtuality continuum" where a consistent transition between real and virtual environment takes place [114].

Augmented Reality (AR) describes the combination of virtual, computer-generated content with real-life objects. The virtual objects appear in a spatial context to the real objects so that the user perceives them as enhancement of the real world. In the opposite to VR, where the user is located in a completely artificial world, AR always uses the real world as medium.

In Figure 2.44 this "reality-virtuality-continuum" is depicted for an industrial robot context. On the left side is the real environment of the facility. For construction and programming, a pure virtual CAD model is already used, shown on

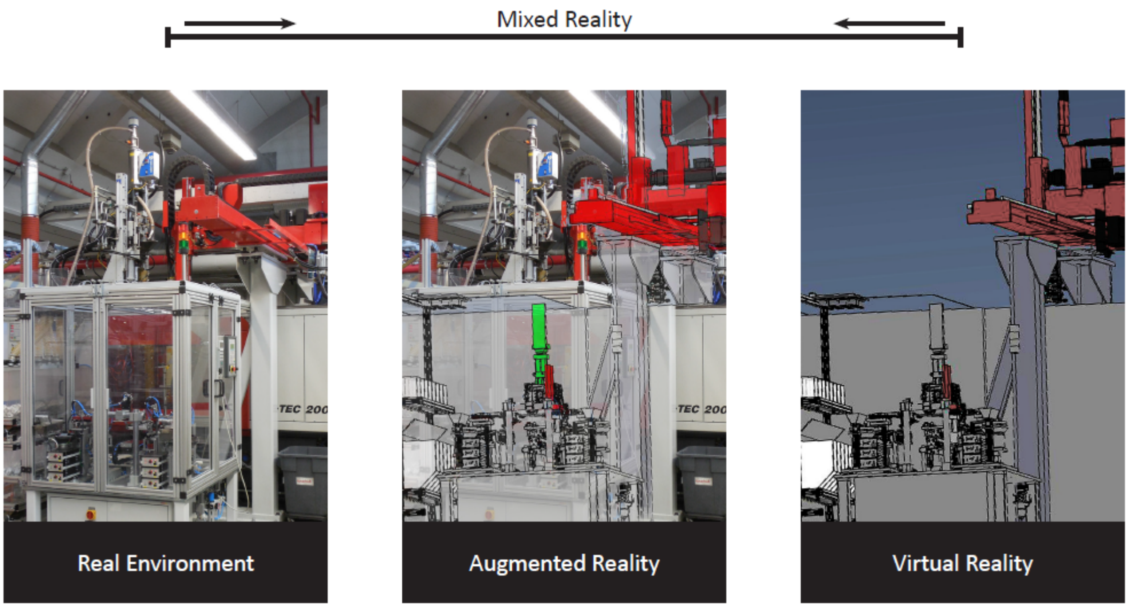


Figure 2.44: *Reality-Virtuality-Continuum following [114]*

the right side. The mixture of both worlds is augmented reality. The task for each AR application development is to determine where exactly a solution has to be located in that continuum. That means answering the question how the scenario can be supported with augmented reality in order to provide a measurable benefit for every day work.

In Figure 2.45 the essential components of each Augmented Reality application are displayed. At first there is the reality displayed on the upper left, perceived from a specific viewpoint. Depending on the technology, it can be a fixed on a moving viewpoint observed by a camera or by the user. AR applications with fixed viewpoints are called Spatial Augmented Reality [115], otherwise we talk about Mobile Augmented Reality. In both cases we need to estimate the transformation of the viewpoint relative to the object frame, in the latter case (of Mobile AR) we need to do this continuously by a reference or tracking system. On the upper right, the virtual model is displayed. This can be a model of the whole scene or only location information and small hints. The virtual content must present the same viewpoint as the real content. This can be done by rendering the scene from a virtual camera with the same transformation as the real camera. Next to the user on the lower left, the AR device is displayed on the lower right, where virtual objects in the real world are displayed. A tablet PC is visualized here, but there are several different possibilities for the device. The device can contain interaction device (H2M communication) as well as visualization device (M2H), but these both can also be separated. Position estimation via the tracking system, data processing and rendering can be done by the device, but can also be separated onto different hardware components.

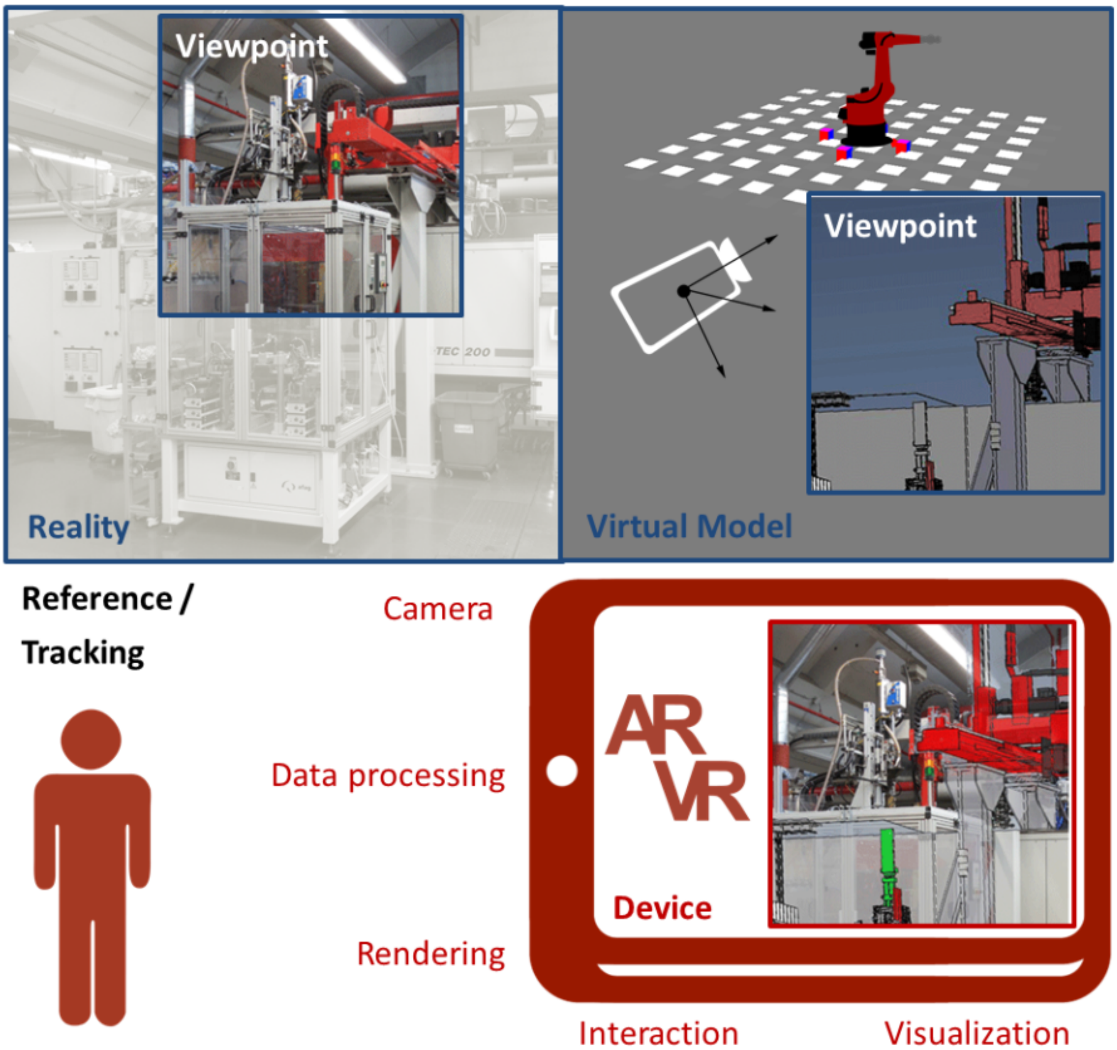


Figure 2.45: *General architecture for Augmented Reality applications*

2.5.6.3 AR and VR devices

The definition of AR is not intimately connected with the used technique. There are a lot of different devices used for the display of AR. It can be distinguished between categories "Head-worn", "Hand-held" and "Spatial", each containing several subcategories [116]. Especially hand-held devices with AR are already in use, for example apps using the built-in camera to include additional information on the smartphone or tablet PC screen.

Head-Mounted Displays (HMD) can be divided in Optical-See-Through and Video-See-Through displays. In the first case there is a blank screen, through which the real environment can be perceived. In the latter case, a video monitor shows a camera picture and the augmented content. The latter can also be used as VR device. The screen is placed directly in front of the eyes and offers (in the best case) different views for each eye to reach a 3D-effect. There are already some off-the-shelf devices available on the market.

2.5.6.4 Camera parameters

A camera projects the three-dimensional world into a two-dimensional image. The human brain interprets those images as three-dimensional. If the camera parameters are known, the computer can also use the 2D data to identify or locate objects, but for this additional information is needed.

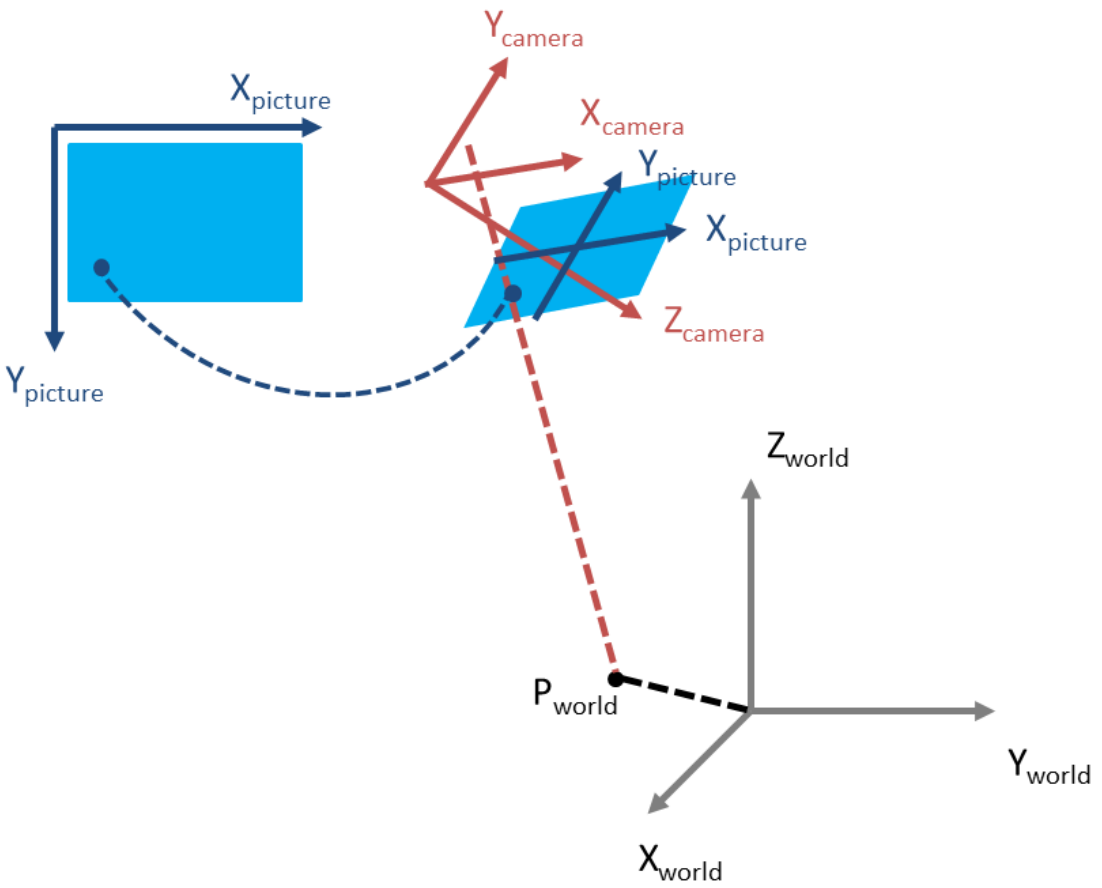


Figure 2.46: *Visualization of camera calibration*

The camera parameters define the transformation between the 3D and the 2D world. In our example in Figure 2.46, there is a Point P_{world} in the world frame. The extrinsic parameters define the transformation ${}^{world}T_{camera}$, which can be expressed in a Rotation R and a Translation T. Another transformation ${}^{camera}T_{image}$ with the intrinsic parameters is used to project into a 2D image while expressing the distortions caused by the camera lens. The intrinsic parameters can be expressed in matrix A (for more details see e.g. [117]).

2.5.6.5 Pose estimation

The problem of pose estimation in an industrial environment is visualized in Figure 2.47. Here a portable device like a tablet PC is used for a Mobile AR application. The position of this device and its onboard camera have to be determined very precisely to enable the fading of the virtual model. The tracking has to determine the projection of the relative coordinate system of the camera of the (stable) base coordinate system of the robot in real time.

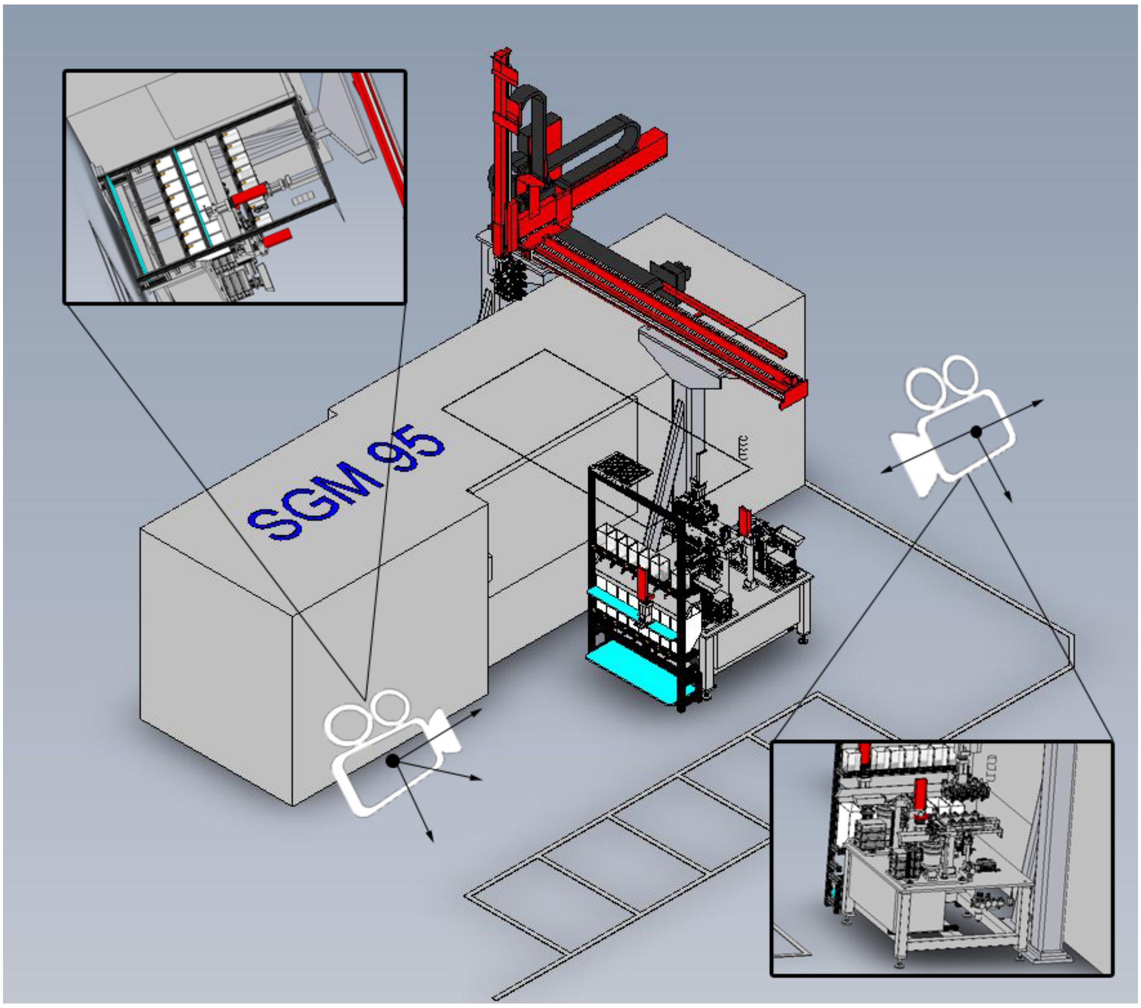


Figure 2.47: *Pose estimation problem*

2.5.6.6 Benefit for maintenance

In general, there are several benefits of Augmented Reality applications mentioned in [113] that also apply for Virtual Reality:

1. Computer support enhancement of human senses with additional digital objects.

2. Visualization of detailed and complex information is possible.
3. Support for complex and difficult tasks.
4. Minimization of time-to-content, the time needed for getting the required content.
5. Combination of haptic and digital experience.

But those benefits do not appear automatically. As mentioned above, the AR field is widespread, and the question arises, which techniques and devices achieve the best results. Also it is important to keep some constraints in mind:

- **Diminuation of work errors**. Research on AR for a production task has been performed with test persons who assembled a construct of Duplo cones [118]. One group used a paper-instruction, the others had the instruction on a LCD display, on a HMD or another AR application. The task consisted of 56 steps and was kept domain neutrally. The performance of the participants was estimated on the basis of the required time, the errors and follow-errors and the subjectively impressed effort. The results show that the probands with AR equipment show less errors using the AR application. There has been no difference regarding the used time.

  The research in [119] also sees a great potential for reducing errors. A user is able to see information to the actual task which may result in a higher quality. Also the training from apprentice to expert may be faster, if the additional information is provided on demand. The attention of a worker can be guided to specific points of interest. The user builds up a "mental map" from pairs of real content and virtual information, so that the understanding may grow and lead to less errors.

- **Acceleration of repair tasks**. Complex tasks with a high search effort can be supported with AR [120]. Performing maintenance or repair actions can be faster, more productive and more flexible due to the direct spatial context of the information.

- **Better building of mental models and better learnability**. AR is a good medium for training and learning new abilities, states [121] after using different AR methods in a maintenance context. But they also mention the problem, that workers may become dependent on the AR applications, because the cognitive effort is too low. They propose using visual cues instead of detailed text instructions. Under those circumstances they have found a lower number of unsolved errors during a maintenance task.

  There are also design criteria for the creation of AR applications [122], so that the learnability is not decreased. It has been shown, that AR helps to build mental models of a task or function. But information on the actual state of the work and about the context of the working step is necessary. They propose, that the application only provides hints, so that the own performance of the worker is not hindered and he can build mental models.

Altogether the evaluation of AR methods is difficult as many factors interact in each evaluation. Different device specifications and experiment conditions like delay may have a huge influence on usefulness of Augmented Reality in a specific task. This must not be underestimated in experiments.

2.6 H2M communication

In this publication, the term Human-to-machine communication is mainly used for the control process of a machine, especially a robot. In most cases, this happens in a closed loop, but can be distinguished from the questions arising from visualization and display, which have been discussed in the last section.

We use computers, machines or robots for different tasks. In Figure 2.48 on the left side we see a human with a specific workload. The amount of a typical work load for one human is visualized by the dashed line above the human's head. On the right, additionally to the human, a computer or machine is used for different tasks. The computer can help in the first case to extend the maximum amount of load, that the human is able to carry. In the second example, the human can be relieved by the machine while the same amount of work is performed. The machine can be used as a backup or can even replace human work - a discussion with a lot of controversy.

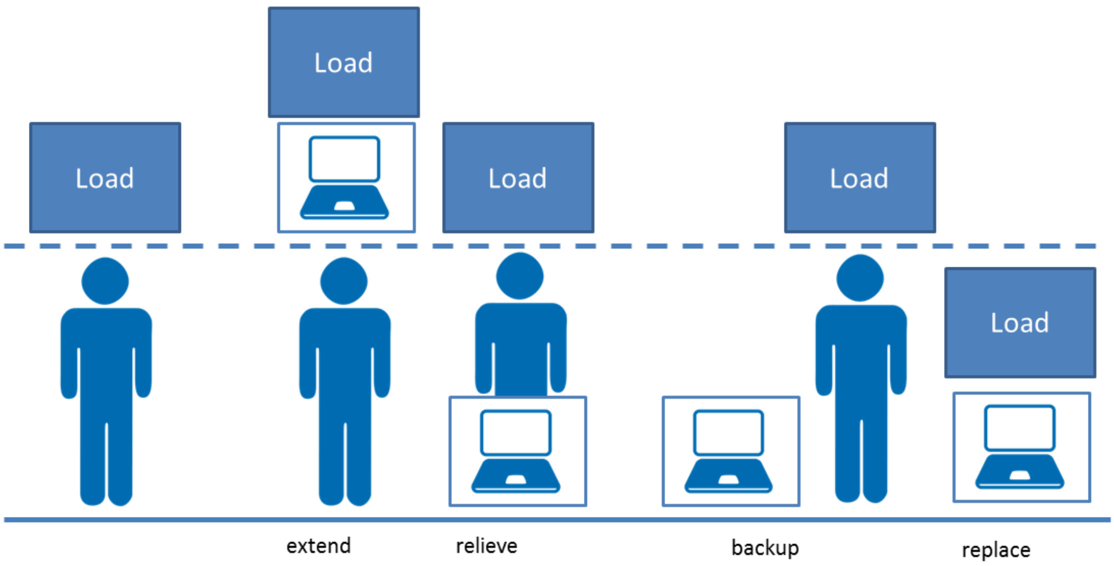


Figure 2.48: *Sharing and Trading Control after [64]*

In this thesis, the machine will be a robot in most cases. As mentioned before, humans will not disappear in modern production lines [3], but will most certainly work more closely to robots, since the strict separation of working spaces of humans and robots has experienced a softening due to recent robotics research. Taking this into account, the problem "to understand and shape the interactions between one or more humans and one or more robots" [4] will play an important role in industrial Internet applications. This problem is called Human Robot Interaction. As it is a child of the Human Computer Interaction field, the development and analysis of interfaces between machine and human is the key goal of research.

2.6.1 Human in the maintenance process

In the automation industry, especially in the area of maintenance or repair, the human operator has a key role as depicted in Figure 2.49. At the lower right of the figure the process is depicted, where faults can occur. Of course, there are the "classical" tasks of supervision like monitoring and automatic protection: the plant can produce alarms after a threshold violation. These are monitored by humans and an appropriate counteraction like a shutdown might be undertaken. If it comes to more complex plants, additional measurement is required: by measuring the input signals and the accessible state variables $x(t)$, it can be decided if a fault has happened or if a change of the output signal comes from measurable disturbances. This information is provided to the supervision level: a software toolchain derives features, symptoms and finally faults, which are then categorized and presented to the human operator. He or she is then deciding on the adequate measures depending on the hazard class of the diagnosed fault(s). That is called fault management or process-oriented asset management. The measures can lead to a stop, a change or a reconfiguration of the process of event maintenance or repair actions.

2.6.2 Human control of robots

In order to reconfigure a plant with an industrial manipulator for maintenance or repair actions, the direct control of the robot is necessary. In most cases mentioned in the last section, this is done directly while the operator is standing next to the plant. As we focus on telemaintenance, we need to cover the topic teleoperation as well as the superordinate topic human supervisory control.

2.6.2.1 Definition of teleoperation

Robotic teleoperation is a comparatively old and therefor very broad field of robotics. Generally "a robotic teleoperation system allows to reproduce the actions of a human operator and to interact physically with objects and environments placed at a distance" [123]. This is schematically visualized in Figure 2.50: The operator is controlling a master system, which is communicating over a communication channel with a slave system that can be influenced by the environment there.

For an overview of applications see [124]. There are several fields using teleoperation of robots. Traditionally they are used in areas where it is impossible or dangerous for human to access - such as in search and rescue [125] or planetary

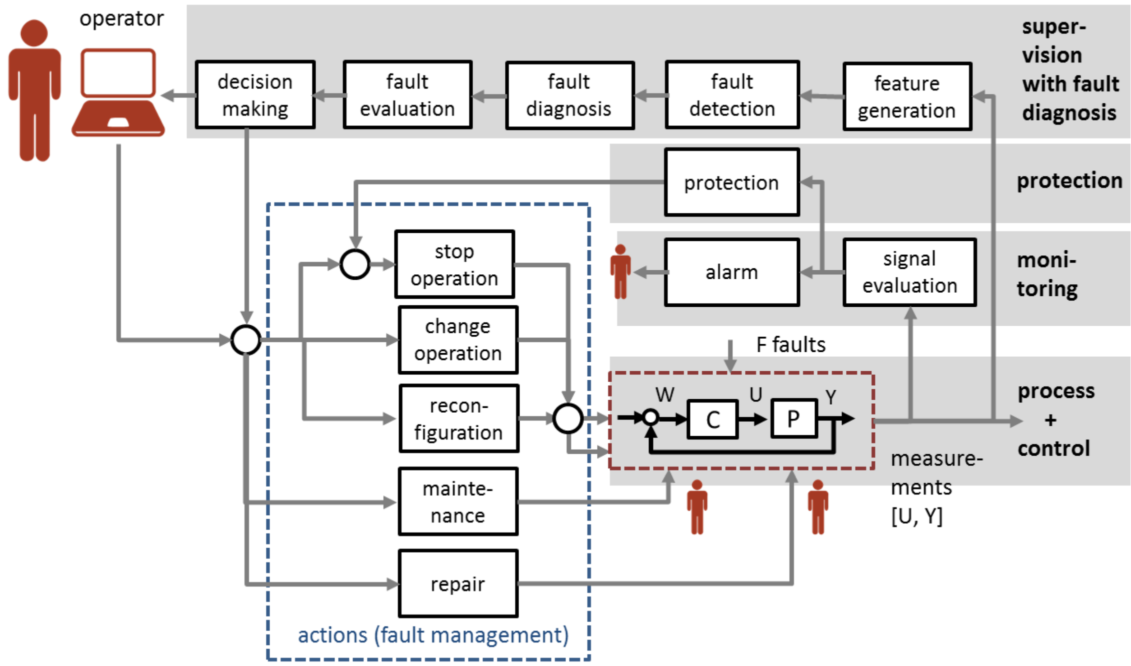


Figure 2.49: *General scheme of advanced supervision methods with fault management (supervisory loop) from [34]*

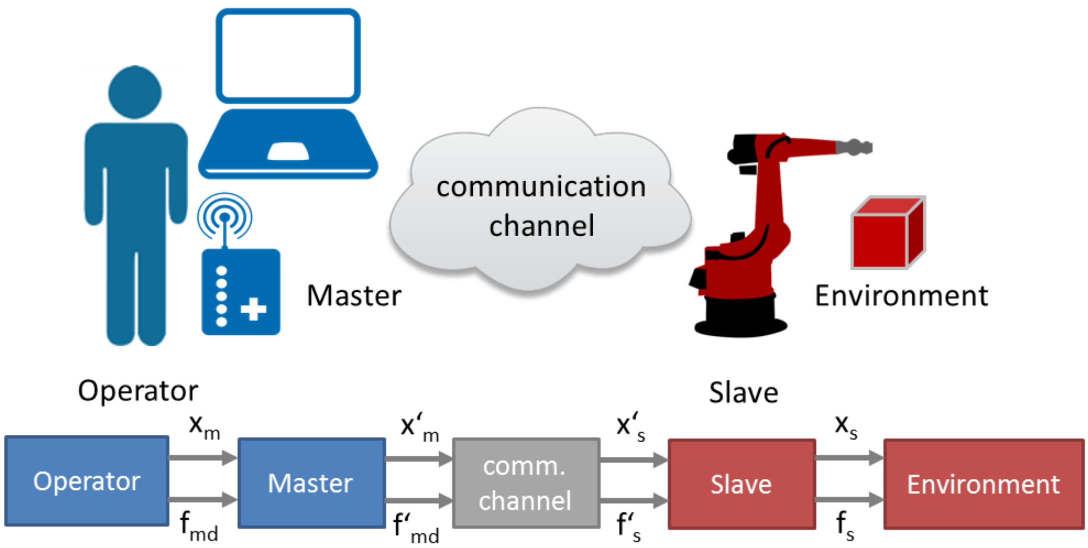


Figure 2.50: *A telemaintenance system and its block scheme reprentation. Subscripts m and s refer to variables at the master and slave site following [123]*

exploration [126]. Moreover, there are other applications for the remote scenario, such as e.g. human-robot teams [127] [128].

In teleoperation, all decisions of the operator are based on what he or she can perceive and understand from the information presented to him or her through the user interface. This is limited by the perception of the robot's sensors, limitations and disturbances of the communication channel, and how the information is pre-processed and presented to him or her. In order to operate the robot at remote efficiently it is necessary for the operator to understand the remote environment around the robot (situation awareness, see Section 2.5.2).

2.6.2.2 Human supervisory control

Theoretically speaking, the teleoperation task can be seen synonymously to human supervisory control as defined in a strict sense in the work of Sheridan [64]. Here the control loop is closed through the human and the control computer. The central visualization of that model is presented in Figure 2.51. Instead of speaking of "master" and "slave" unit, the model introduces closed control loops which are carried out through the two computer units: The remote human interactive system (HIS) and the task interactive system (TIS). It further draws the analogy between humans and robots regarding input and output stream. It also considers the nested control loops involved on the different levels of the teleoperation system.

There has been lot of interest in the control theory explanation of this model. Figure 2.52 depicts a simplified approach: The operator observes a system state vector x and tries to take action u to maximize some given objective's function $V(x, u)$. This figure can also be translated in a maintenance context: The estimation of system state is inspection $I(x)$, the measures u can be maintenance, repair, overhaul and control. All of the measures have the goal to maximize the working supply of the plant $V(x, u)$.

The human supervisory control approach integrates several of the theories mentioned above, such as the use of mental models and transformation costs. In Figure 2.53, these theoretical concepts are visualized in a teleoperation scenario: The operator has a mental model of the robot (1.), and sees displays representing another concept model of the robot (2.). Further, the understanding of this model is deeply connected with situation awareness. But the computer might use a different model representing the robot for machine-to-machine communication. The controls, like for example the teach pendant, present another model of the robot. Those models have different frames of reference, so that the user performs different mental transformations during control, each with different mental "transformation costs". This has to be considered in the design of a complex human supervisory control scenario.

2.6.3 Extended Human Supervisory Control

The classical approach of Human Supervisory Control can also be applied to the Industry 4.0 setting, but needs to be extended. This is visualized in Figure 2.54: Not only does the remotely connected expert perform Human Supervisory Control

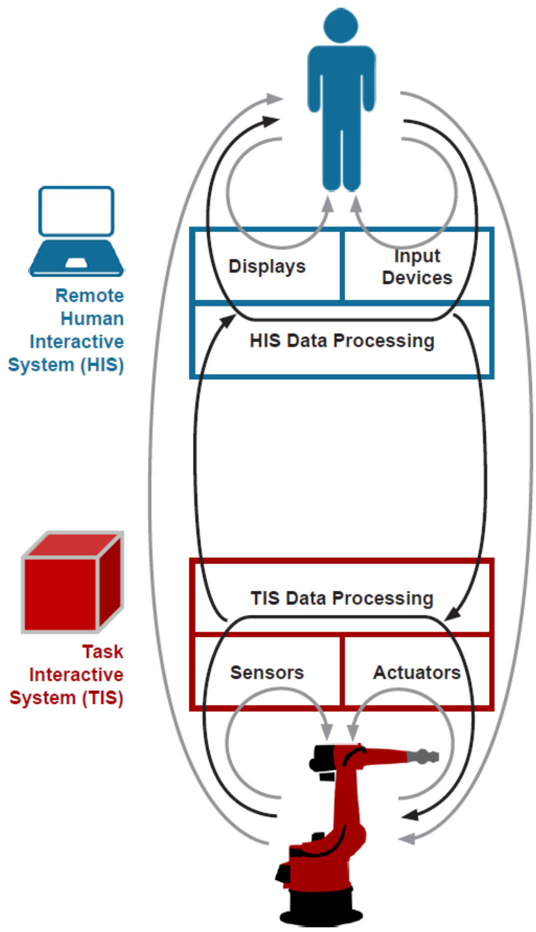


Figure 2.51: *Human Supervisory Control model as introduced in [64]*

(a), but also the local service technician (b). The model therefore needs to be extended as shown in (c).

In the European Union and most other parts of the world nobody gets the legal permission to teleoperate an industrial robot in an automation facility if there is no local supervisor present. Depending on the specific implementation, the service technician colocated with the plant has to press some kind of dead man's switch to enable the external control. Therefore, he or she also performs Human Supervisory Control and mirrors the conceptual structure of the main teleoperator (the expert). The entire model is depicted in Figure 2.55.

2.6.4 Summary

This chapter has brought up questions about communication between humans and machines in the industrial internet setting of telemaintenance. Telematics as a combination of telecommunication, automation technology and computer science is a large, multi-disciplinary field and this publication can only provide a glimpse on the huge amount of different research questions in this field. Along the categories

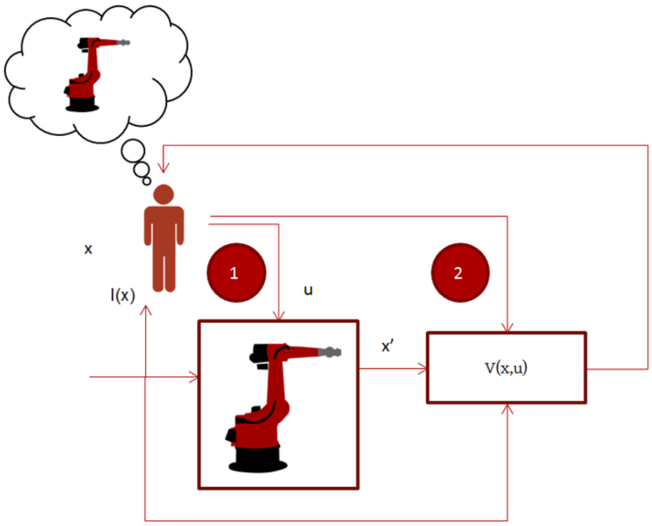


Figure 2.52: *Easy approach for Human Supervisory Control as a control model according to [64]*

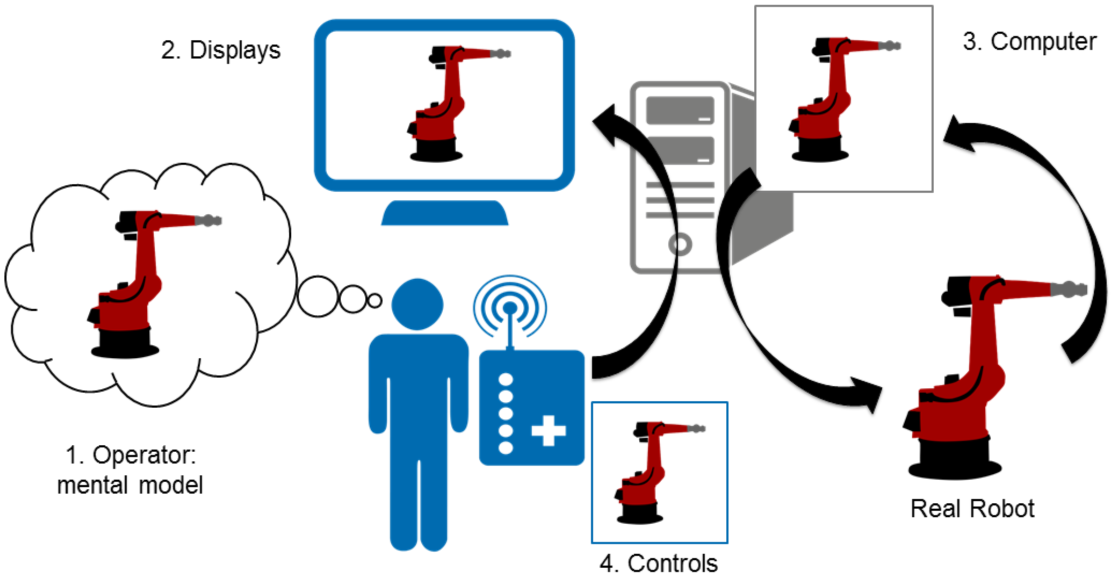


Figure 2.53: *Mental models in Human Supervisory Control according to [64]*

of who is communicating with whom, four major fields have been identified. Thus, this chapter explained the fundamentals of machine-to-machine (M2M), human-to machine (H2M), machine-to-human (M2H) and human-to-human (H2H) communication in the context of an Industry 4.0 setting. Those explanations will serve as fundamentals for the research presented later in this thesis.

After building up the complex communication scenario of a telemaintenance setting with all partners, we came to the central model of extended Human Supervisory Control, the model from [64] with two communicating humans. A synthesis of the

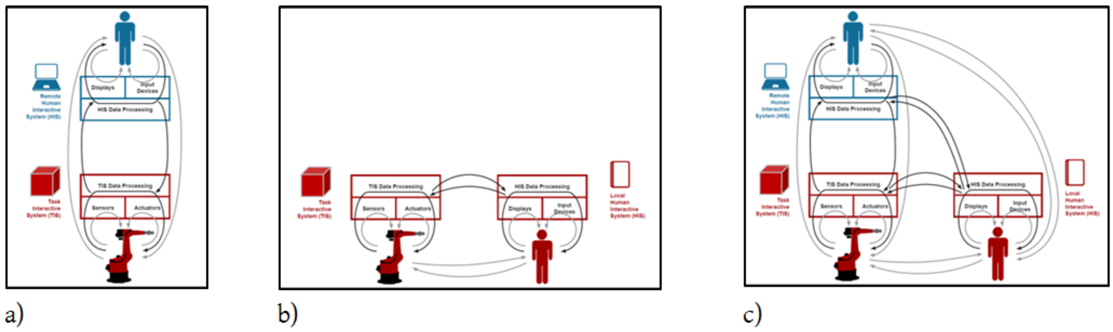


Figure 2.54: *Combination of Human Supervisory Control: a) remote HSC b) local HSC c) extended HSC*

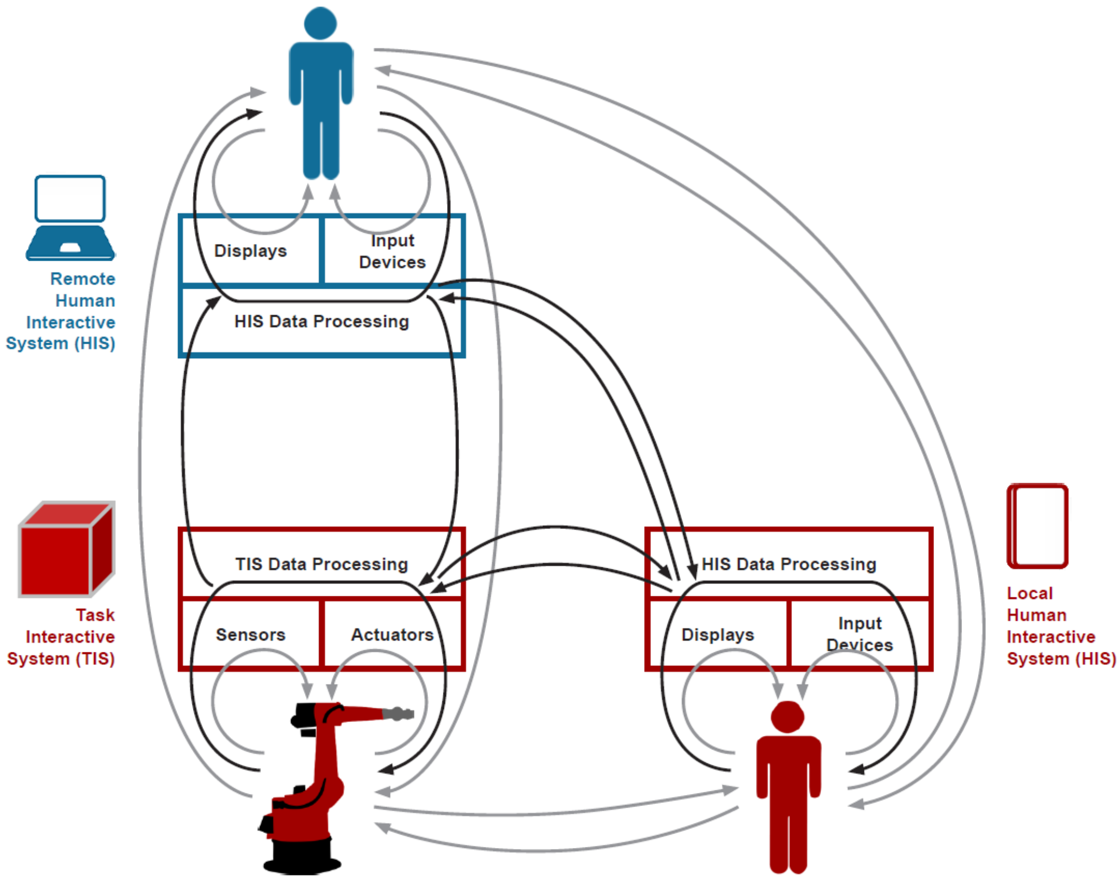


Figure 2.55: *Extended Human Supervisory Control required for Telemaintenance*

explained material is provided in Figure 2.56: An external expert on the left side has to figure out a problem in a production plant. He or she has a mental model of the current situation there (the dashed blue square), an understanding "what is going on" (situation awareness). The expert also has an idea about the solved problem or the situation (the blue square).

We walk through Figure 2.56 beginning on the left side. The expert has a mental image of the current and the target state (the blue square) and can identify measures necessary to reach this goal. He then communicates the instructions for the measures to the service technician with the message (green circle), which is the mental model encoded in communication and software tools. This message is transferred with the communication link and eventually loses some information or context (yellow circle). This contains the risk of misconception by the technician. Nevertheless the technician translates the heard and seen instructions to a mental model of his own (red triangle), and begins to perform the actions at the plant. The action and the plant are monitored by different cameras with different viewpoints and are presented together with sensor information through a user interface to the external expert. The expert can then take further actions, if he can diagnose the suitability of the current actions of the service technician with the task goal. In the picture the dotted blue square does not match the blue square exactly, so that further instructions are required. The expert is also able to control the robot directly (teleoperation).

The specific project context of this theoretical and general model will be shown in the following chapter.

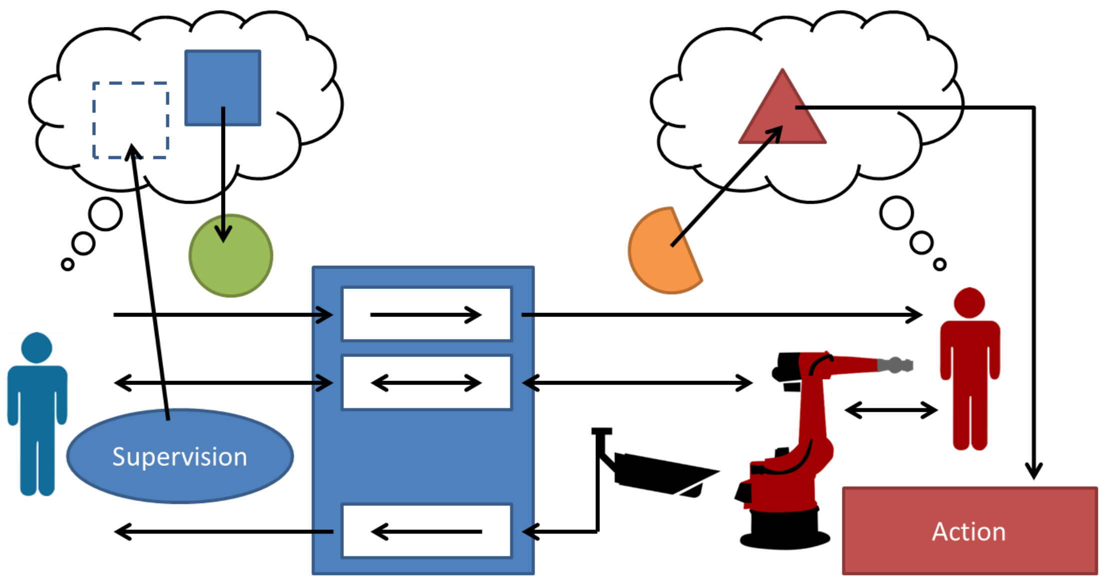


Figure 2.56: *Communication processes in extended Human Supervisory Control*

Chapter 3

Project MainTelRob: Initial situation and statement of the problem

The frame of this work is the three-year research project "MainTelRob" (Maintenance and Telematics for Robots) and its sucessor "Bayern.digital" funded by the Bavarian Ministry of Economic Affairs, Infrastructure, Transport and Technology. The project consortium of MainTelRob consists of the Zentrum für Telematik e.V., KUKA Industries and Procter&Gamble. The projects aims to develop a telemaintenance infrastructure for an industrial robot in a plant producing plastic parts for electrical toothbrushes. Next to network control theory and bandwidth management, one focus of the project lies in the design of the human robot interaction in the telemaintenance scenario (this thesis). In order to understand the challenges and the side requirements of this topic, this chapter describes the project setting, the used work flow and methods and the general results of the methods that will be used in the later chapters. Some special results will be covered in the related chapters.

3.1 Project setting

3.1.1 Motivation and project goals

Although simple telematics applications have been on the market for several years, many interesting research problems need yet to be solved to improve the reliability and efficiency of facilities through interactive services. The Zentrum für Telematik e.V. (ZfT) and its industrial partners, KUKA Industries (former Reis Robotics) from Obernburg and Procter&Gamble (former Braun) from Marktheidenfeld, cooperate on MainTelRob ("Maintenance and Telematics for Robots") to solve these problems and create a comprehensive new telemaintenance system.

MainTelRob focuses on interactive telematic methods for real-time situations. The huge progress in the field of telecommunication and information technology allows deploying increasingly ambitious services to help reducing the need for costly long-distance travels. In combination with the control and automation technology in telematics, this offers novel possibilities to gather sensor and status data from remote industrial plants and even, if appropriate, to respond to real-time critical situations.

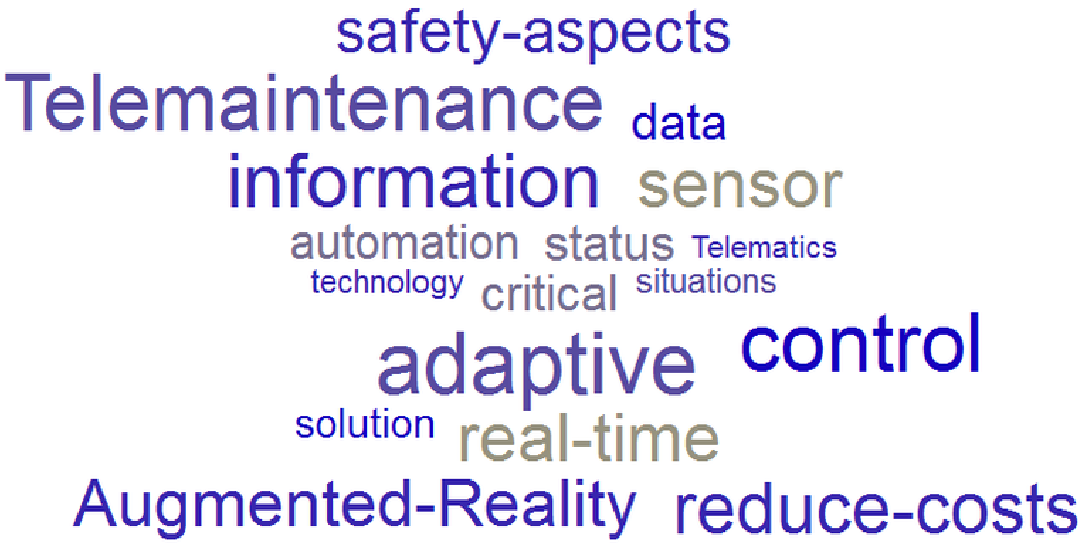


Figure 3.1: *Word cloud "Project goals" created with http://worditout.com/*

These methods are implemented with the help of state-of-the-art control and communication technologies. While offline data transfers are already an established best practice in the field of industrial remote maintenance, we go further by establishing control loops over communication lines. Safety aspects must be guaranteed through autonomous adaptive control solutions and a reliable identification of the system state. In addition, novel visualization techniques like Augmented Reality will be developed for the user interface, to enable the distant expert in a telemaintenance facility to identify potential problems early and intervene effectively in time-critical situations. This is how a powerful new generation of remote maintenance and analysis systems is to be launched.

The challenge of developing user interfaces for those prototypes lies in the special requirements of the industrial work domain. Highly sophisticated technical tasks need to be carried out under time pressure and in a noisy environment. The human machine interaction of the remote tasks is especially difficult demanding a high level of situation awareness (see Section 2.5.2). There is no experience with those remote tasks, as they are only possible with the developed technology.

The reason why two big companies take an interest in telemaintenance is the ongoing discussion on Cyber-Physical Systems and the Internet of Things (IoT). According to these the integration of computer science and communication technology in automation industry has a huge potential for productivity.

KUKA Industries has worldwide, geographically distributed customers who normally own more than one robot. They want to expand their service activities beyond phone support and struggle with the high demands of teleoperation regarding especially the security infrastructure. They also want to specify a proper combination of services for supervisory control. Based on other projects, they also want to intensify their human-machine-interfaces regarding Virtual and Augmented Reality.

The facility of Braun in Marktheidenfeld has to keep up with the high standards of Procter&Gamble. The quality of a molding process with involvement of industry robots is influenced by multi-dimensional parameters and breakdown times come with very high costs. Based on the project MainTelRob they want to establish a predictive maintenance strategy. Together with 71,6% of Germany's companies they see a huge potential for early information on their production system, but complain about the insufficient quality and the lack of currentness of the data [8].

3.1.2 Project concept

In MainTelRob, we aim to employ user-friendly Augmented Reality techniques to intuitively represent a remote situation to an operator. In addition, autonomous reactions and remote control are combined in order to absorb peak loads and enable injury prevention, especially during critical phases of operation. Both above mentioned essential elements of a user friendly telemaintenance system will be closely integrated to simplify the remote maintenance of complex automation facilities. The project set-up and concrete tasks are divided into two main areas, which we explain based on Figure 3.2:

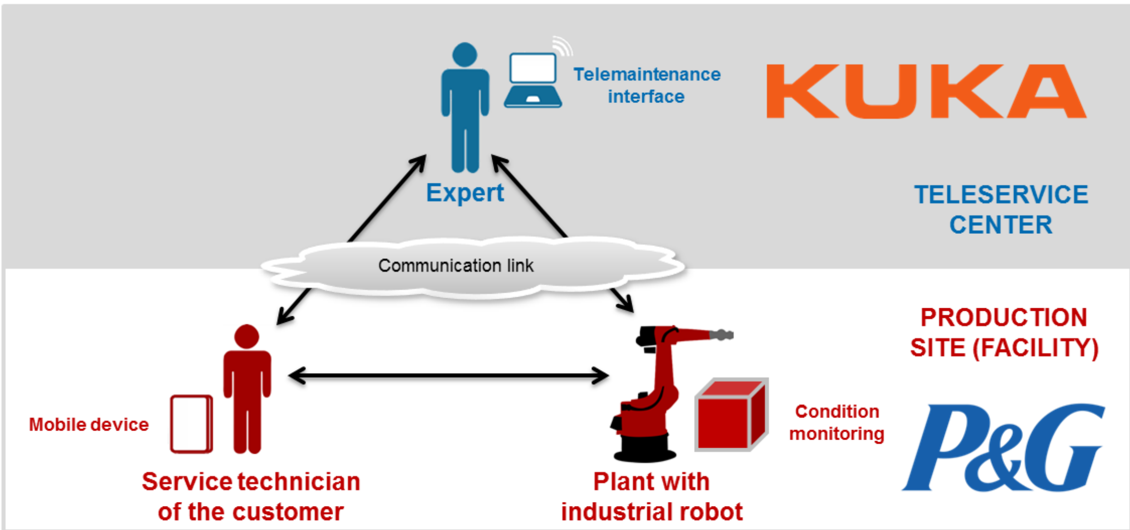


Figure 3.2: *Schema of telemaintenance*

The robot and the local service technicians are situated at the production site of the customer (provided by Procter&Gamble) as depicted in the lower half. The robot is currently supervised by a condition monitoring system, which has been further developed within this project. The service technicians, possibly with differing qualifications, will be supported by a mobile environment by providing assistance functions. Our industrial partner KUKA Industries provides the telemaintenance center, depicted in the upper half of Figure 3.2. The experts located there can use their telemaintenance environment to access the robot at the customer facility and communicate with the local service technicians. The main goal is to develop various tools for these experts, so that they can help to solve problems without physically traveling to the factory.

Figure 3.3: *Photo of the production plant used for MainTelRob*

The production site of Procter&Gamble in Marktheidenfeld is depicted in Figure 3.3. The red industrial manipulator of KUKA Industries produces the lower part of electronic toothbrushes (shown in Figure 3.4) in a two-component injection molding process.

3.2 Work flow

The challenge of developing in the MainTelRob project context lies in the special requirements of the industrial work domain. In order to address these challenges, a special project work flow had been developed and it seems to be useful for similar problems or projects in which completely new frameworks are developed for an unfamiliar work domain. It is based on the UX cycle [54] which mainly consists of the steps analysis, design, prototype and evaluation, while the repeating of one step or jumping back to the last step is encouraged (compare Section 2.3.6). We adjusted the UX cycle and the methods for our project setting.

The work flow is depicted in Figure 3.5. The steps will be explained by the

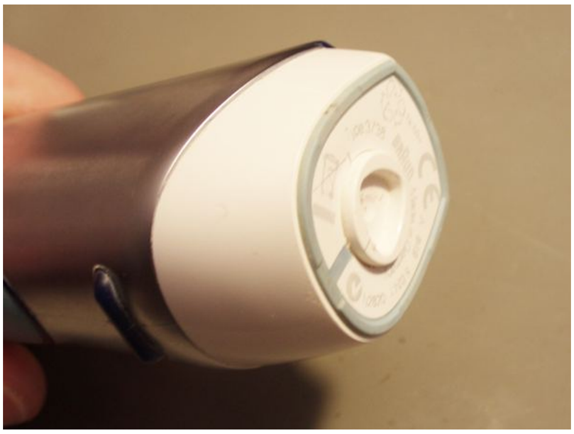


Figure 3.4: *Lower part of electronic toothbrushes*

following sections, but we give a brief survey here: In the first year of the project we carried out an elaborate contextual analysis which resulted in six scenarios with a specific use case each. This was recorded in the specification document. The first year also was used to set up the infrastructure. In the second year, different fields of knowledge resulted in the functional prototypes which implemented each scenario. This process was accompanied by a design process of the user interfaces which involved user feedback and design informing models. Each prototype undertook a formal acceptance process based on the requirements of the specifications before the end of the second project year. In the last year, the evaluation process involved a pluralistic walkthrough which guided the revision of the functional prototypes to technical prototypes. The evaluation tests with end users enabled an economical rating of the benefit of the project. In the following sections, the particular steps are described in detail.

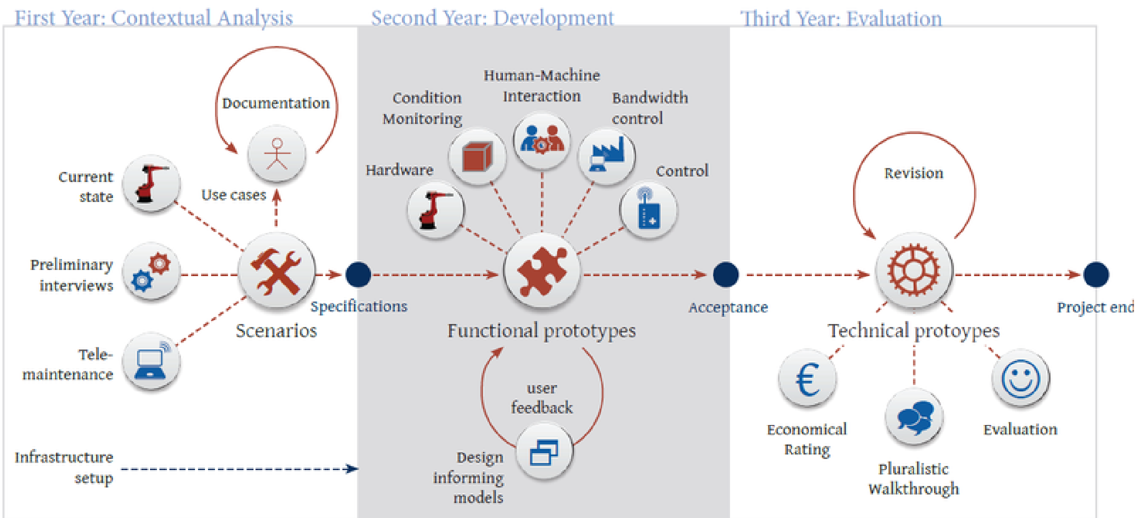


Figure 3.5: *Project survey*

3.2.1 Current state

The research team consists of roboticists but nevertheless the distinct production environment was new to the researchers in the beginning. This was one reason for the elaborate contextual analysis process. The main goal of contextual analysis is to enrich and inform the user interface design with insights related to usage scenarios and environmental factors (for theoretical details see Section 2.3.6). It involves documenting the work domain at its current stage.

3.2.2 Preliminary interviews and telemaintenance

To get a deep understanding of the way the users see their working environment, we spoke with workers on different hierarchy levels of both the factory (Procter&Gamble) and the manufacturer (KUKA Industries), which will later hold the telemaintenance center. After the plant has been identified by the project partner Procter&Gamble, the team has been given a tour of the plant. It proved very useful for the later design process and for discussions without machine access to have videos of a production cycle of different parts of the plant. The team also took many pictures of the environment and documented especially every point of human-machine-interaction like terminals.

The team even spend one week in an instruction course for industrial robots, provided for the technicians of industrial robot customers by KUKA Industries. This was necessary for a broader view of different manipulator applications, because the telemaintenance architecture should apply for all of the manufacturer's customers.

The goal of the project was to introduce new information and telecommunication methods into the production process, especially the new possibility of telemaintenance. The fundamentals for the topic telemaintenance are covered in Chapter 2 but inquiring the workers' opinion about the introduction of telemaintenance was necessary. Obviously, there was no way to document the current stage as the infrastructure and processes were set up anew. We structured our interviews by asking the questions of the current stage first and got improvement suggestions for the current stage. After that we introduced the idea of telemaintenance and tried to get as many ideas out of the interview partners without leading them in a particular way with our questions.

We wrote down all the interviews as a base for qualitative data. For getting a relaxed conversation base with people who don not know each other, we recommend taking handwritten notes instead of videotaping or audio recording, because the people talk more freely. Later in the project we also used these methods but only with people who had gotten familiar with staff and idea of the research project. The whole material served as qualitative data and especially helped us also to identify the required fields of study.

3.2.3 Specifications

Together with engineers of our project partners we held several project meetings in order to find specific problems, that could be solved in a telemaintenance setting. We always took very detailed notes and also provided a short official protocol, both serving as qualitative data. It was important to see the robot manufacturer's view together with the view of the robot applicator. The first naturally focuses on the manipulator and the second merely regards the entire setting of the plant. In the end, we decided on three local scenarios, that can be applied without any internet connection and three remote scenarios built on top of the local ones. Additionally, we decided on one specific use case per scenario. This had to be a specific task, that should be accomplished with the new prototype, but is already executed in a different way in active production. The scenarios are described in the context of their prototype development in Section 3.3.3. We wrote down the specifications and reconciled the text in several iterations with the partners, as it is usual for classical software engineering.

3.2.4 Documentation of the use cases

In the next step, we documented each use case on the base of its actual execution. One example for such a use case is the change of an axis controller in the industrial robot's switch cabinet which takes approximately half an hour. Another example is the semi-annual maintenance routine that takes up to two hours. Our project partner provided us with different service technicians who performed the task and told us what they were doing meanwhile (think-aloud technique [54]). We recorded the work on video, took notes and posed questions. Each documentation session was followed by detailed interviews on the working field. We requested the corresponding "work items" like documentation check lists and report formulas. Our documentation file was reviewed afterwards by the same technician who had performed the task for our documentation initially.

3.2.5 Infrastructure setup

The test plant is part of a running production and any change or additional equipment needs a long setup time. Especially we needed access to the industrial robot via the Internet in order to do any kind of related research. This needed to be allowed and configured with the help of the cyber security departments of both project partners. Also we wanted to provide camera streams of the plant, so that cameras had to be installed and the image section had to be discussed with the worker's council. We conducted this setup parallel to the design process, but it took one year from the project start.

3.2.6 Fields of study

For the accomplishment of the scenarios the knowledge of various fields of competence was required. See more about the theoretical fundamentals in Chapter 2. Of course, knowledge of the hardware was essential, like the theoretical base of industrial robots in general and the utilized manipulator specifically. But also the other components of the plant, like the molding machine and the montage system, which dynamically interact with each other, play an important role in the setting. The condition monitoring is the feedback loop that surveys the different sensors in the plant. Its sources are the complete range of the robot's sensors, the measurement used by the molding machine and external sensors coming into operation at the montage system. The Human-Machine-Interaction plays an important role and is the main focus of this thesis. But also the fields of bandwidth control and network technologies, which are mandatory by the use of the best-effort Internet and a deeper understanding of control theory for assistance functions for the remote control of the industrial manipulator, have been required by the scenarios. Both latter fields will be covered in detail by different publications.

3.2.7 Design informing models

As a result of the three first steps we gathered a large amount of qualitative data: the interviews, the detailed protocols of the specification discussion, the documentation of the use cases and the corresponding work items. Corresponding to the UX process described in [54] and explained in Section 2.3.6, those have been documented as "work activity notes" and clustered in a work activity affinity diagram (WAAD). As a developer team, we built some design informing models out of this data to make sure that we had the same picture in mind as the project team. We spent a long time identifying the different roles in the setting and the possible information flow between them. In particular, we assembled our knowledge of the work process in flow models, which proved very useful later on. Those models are depicted later in Section 3.3.2.

3.2.8 Design process

Based on the design informing models, we were able to envision the new work process which involved mobile devices, the internet and a telemaintenance center. We depicted our findings in an "envisioned" flow model which is described in detail in Section 3.3.2. We also modeled the tasks, which should be performed by the workers in our envisioned scenario with task models. We acquired direct user feedback with the help of those models and paper prototypes to make sure that the whole team, especially the end users, had a similar view on the production process and the problems.

3.2.9 Functional prototypes

It was very important for the project to use short iterative software development cycles with different working prototypes, so that the users could comment on the product early. We try to use this approach for all of our research and development projects, as we always aim to create completely new solutions which naturally contain a high amount of uncertainty. Technical risks can also be faced early with this strategy. But it is very important in our opinion to have fixed deadlines to the iteration process nevertheless. For each scenario, we lined out a specific date for an official acceptance of the prototype in the second project year. Based on our specifications, we used standard acceptance protocols for software projects for those deadlines and let end users perform the use cases with the new software. For some scenarios we had to repeat the acceptance meeting, because of missing features. We documented the acceptance meetings additionally per video and text. Those acceptance meetings also provided a good platform for detailed discussion and feature requests, which were also included in the documentation. The first two project years resulted in an accepted prototype for each of the six scenarios.

3.2.10 Pluralistic walkthrough

In the last year, we focused on the three scenarios, having the most promising feedback. We organized an expert workshop with different people from the two project partners, where we performed a pluralistic walkthrough following the proposed method in [129]. This method was suitable for our project setting, because it splits the UX inspection from the end user feedback. A UX inspection on the software was provided by three students of Human Computer Interaction (UX novices).

As we were able to perform the workshop with a group of several highly trained and therefore expensive specialists, we needed to safe time. We identified central tasks for each prototype, which had to be solved with paper prototypes. Each person had their own file, where free annotations were also possible. Afterwards the correct solution was provided with the working software. The discussion was documented in detailed notes. The workshop took a whole day and the UX novices were present in the workshop.

The gathered data out of the workshop and the UX inspection was summarized in a diagram. Our development team rated the different annotations with importance and effort. We decided on a redesign of some parts of the software and implemented missing features for the final evaluation.

3.2.11 Final prototypes and evaluation

The main problem for the quantitative evaluation of the prototypes was, that the subject tasks require a very high work domain specific expert knowledge and, hence, can only be performed by few people. Additionally every test in an active production environment is accompanied by huge effort and costs. We nevertheless managed to identify evaluation methods for our three key scenarios which will be described together with the final prototypes in the associated chapters.

3.3 General results

As mentioned above, some results are mandatory for the understanding of the work in general, so they are described briefly in this section. We focus on general results here, the specific data used for the generation of the prototypes is described at the corresponding chapters. After a summary of the contextual analysis the derived design informing models are described. After that the scenarios and the use cases are explained, which are better understandable with respect to the complete setting.

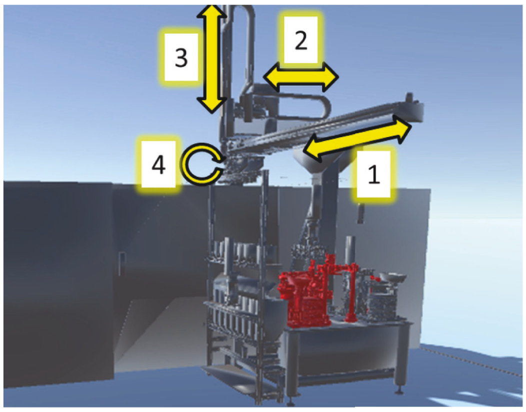
3.3.1 Contextual analysis: the plant

Contextual analysis is a user research method used to understand the task environment and the task procedures that users follow to reach their goals. As some of the knowledge will be summarized in the "design informing models" described in Section 3.3.2.

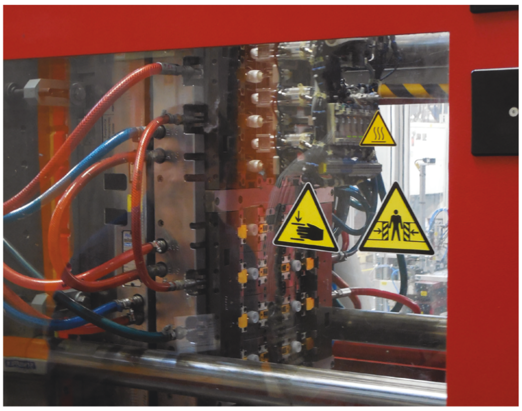
The plant produces bottom parts of electronic tooth brushes in a two-component injection molding process and it consists of the robot, a molding machine and a montage system. Some photos are displayed in Figure 3.6 showing the long shot on the plant a), the CAD model b), a detailed image of the molding machine c) and some photos from the montage system d). The plant is provided with different Human Machine Interaction points, mostly terminals on different parts of the plant. Each part of the plant has at least one user interface and some kind of switch cabinet for PLCs. There are some information piped to the "BDE terminal" which also provides central ERP and maintenance functions, but not all. It is important for contextual analysis to make a list of the available user interfaces and the kind of information presented there.

For a better understanding of the manufacturing process, a survey of the way of a part through the plant is displayed and described in Figure 3.7. It uses a viewpoint from above the plant, where the big grey part on the left is the molding machine (labeled with SGM 95), the red part is the robot and the detailed part on the right is the montage system. It is very important to understand, that always sixteen parts are produced together in an overlapping process. When the first eight parts are brought back to the molding machine for the soft plastics molding process (Step 5), the machine also produces new eight parts with hard plastics (Step 1). Because of this, the production process recorded on video might lead to the impression, that only part of the steps are performed. For our documentation we used video from a long shot viewpoint and a detailed view on the molding machine and the montage system together with timestamps and annotations.

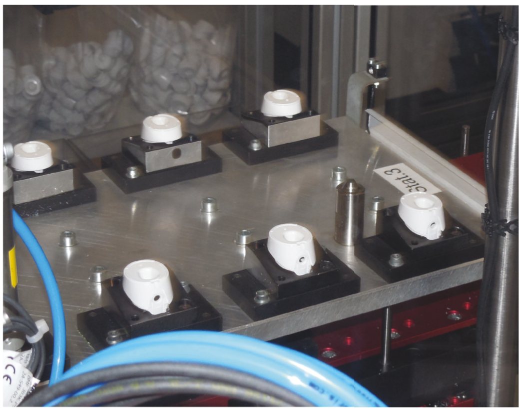
a) Plant (photo of the long shot)



b) Plant 3D model with robot axes



c) Molding machine (open) with upper and lower molding units. The first is accessed by the robot tool.



d) Plastic parts has been set into the montage system by the manipulator tool and will be processed by the montage system (insertion of the gore membrane)

Figure 3.6: *Photos of plant parts*

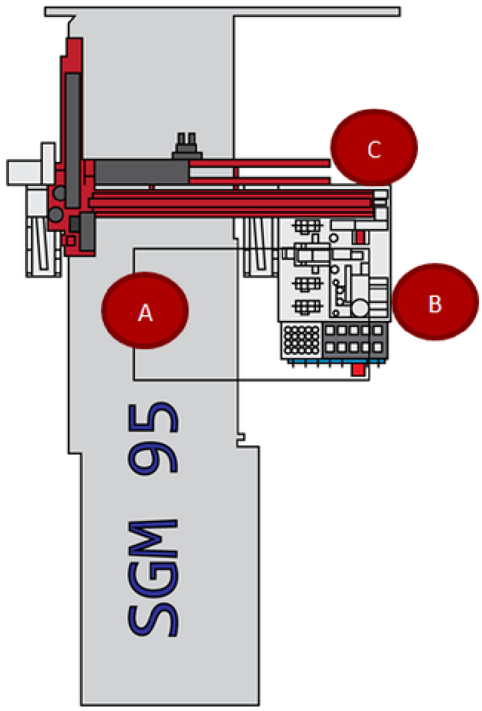
3.3.2 Design informing models

The creation of so-called design informing models helps to bridge the design gap between initial situation and a new product idea as described in [54].

3.3.2.1 Symbols, user classes, work roles, social model, personas

The team agreed upon specific symbols for the communication of the central MainTelRob elements. They are described in the following table of Figure 3.8 and will be used throughout this thesis.

In the project survey and creation of the symbols only two people have been considered – the service technician at the local facility and the expert at the telemaintenance center. But to be precise there are several different user classes [54] involved. On the local side, the machine operator and the higher qualified service technician take care of the plant. Their shift supervisor also plays an important role



Part production at plant 95

1. First molding process in A: duroplastic plastics (Pre-molded part)
2. Transport from molding machine (A) to montage system (B) by industrial robot
3. Montage process of Gore membran insertion in (B) (advancing group)
4. Transport from montage system (B) to molding machine (A) by industrial robot
5. Second molding process: soft plastics (finished part)
6. Transport from molding machine (A) to container for finished parts (C) by industrial robot

Figure 3.7: *Part production process*

in the scenario. Furthermore on the side of the telemaintenance center, the expert is accompanied by a specialized technician and his boss.

To be even more precise, also the user classes need to be subdivided into the so-called work roles [54]. This helps to see the range of all the people who might interact with the application. The work roles cover all the people we have spoken to and also the people the interview partners. They mostly differ in qualification level. We also created the personas [54] on this base.

On particular interesting model is the so-called social model [54] describing the relationships between the different roles and also the feelings of each person involved. This is also based on the qualitative data and was extremely helpful to understand the human-to-human communication in a telemaintenance scenario.

3.3.2.2 Current flow model

The entire setting can be described with flow models. As described above, the plant consists of an industrial robot, a two-component injection molding system and a handling unit as depicted in Figure 3.9.

The machine operator takes care of maintenance while the machine is running,

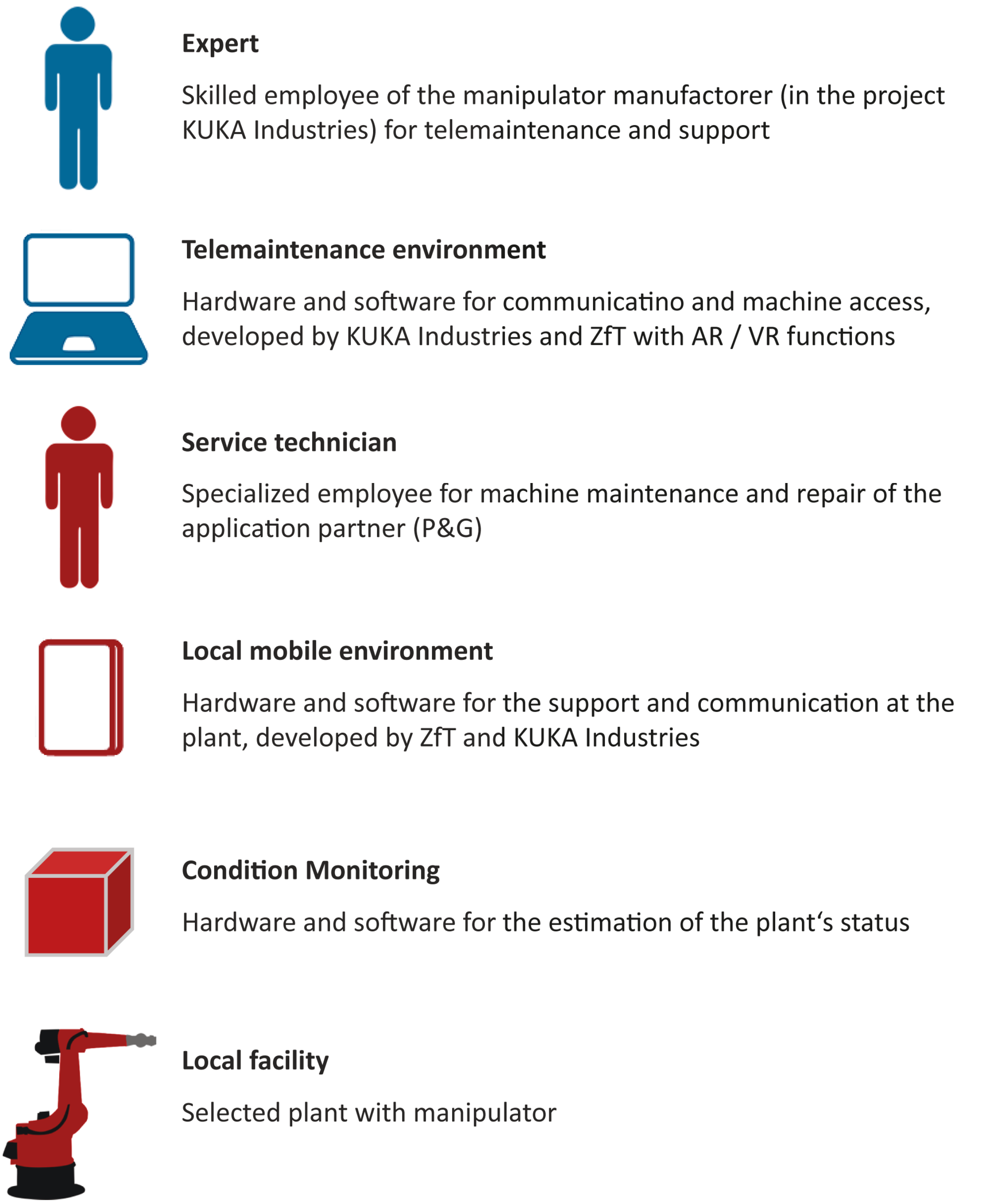


Figure 3.8: *Symbols for different major parts of the project*

which involves cleaning, inspection and lubrification (CIL). He gets his instructions from the BDE (a special terminal) and has to visit different checkpoints around the plant in order to execute the instructions. These include mostly visual checks. If he or she finds a failure during those checks, it needs to be reported to the shift supervisor.

The shift supervisor monitors the failure reports and instructs the service tech-

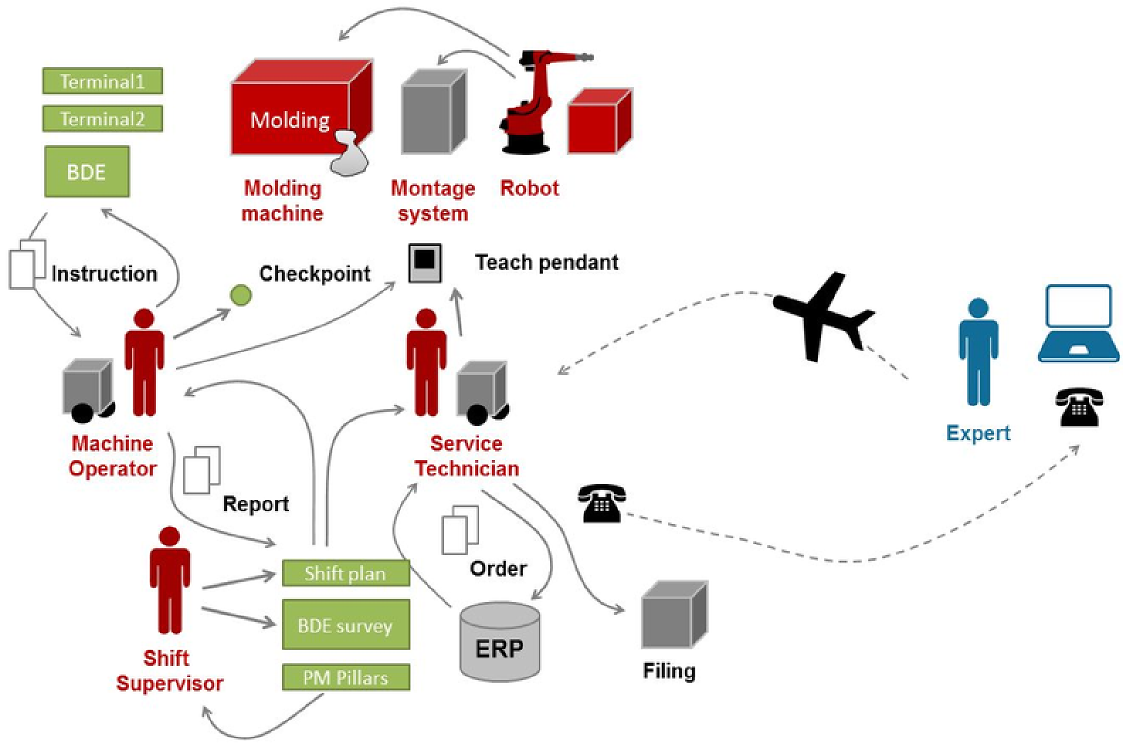


Figure 3.9: *Current flow model*

nicians to solve the problems. He or she also plans the shifts for both machine operators and service technicians. The shift supervisor is under the pressure of the company's predictive maintenance requirements, for example stating the machines to not stop unplanned. This requires a high maintenance quality. He or she can predict repair actions and file them in the Enterprise-Resource-Planning (ERP) software.

When the service technician is called, he or she gets his order directly from the shift manager or out of the ERP software. He or she uses the teach pendant (the programming and steering device of industrial robots) to operate the robot. At the P&G facility where the project had been developed, the service technicians are highly trained and are able to fix every failure on their own. But generally, there is the possibility that they cannot find the failure. Every shutdown period of a plant is very expensive. The shift supervisor can, in extreme cases, call service personnel of the robot manufacturer, who tries at first to provide phone support and in a worst case needs to fly to the facility and fix the problem. Also for optimization tasks, the input from external personnel can be very useful but is not provided very often due to high cost and scarce availability.

3.3.2.3 Envisioned flow model

As described above, the project aims to provide a telemaintenance infrastructure which enables the expert of the robot manufacturer to solve the problem remotely or find optimization possibilities. Furthermore, the teach pendant of the industrial robot should be replaced by a mobile device like a tablet computer. Figure 3.10 provides the envisioned flow model.

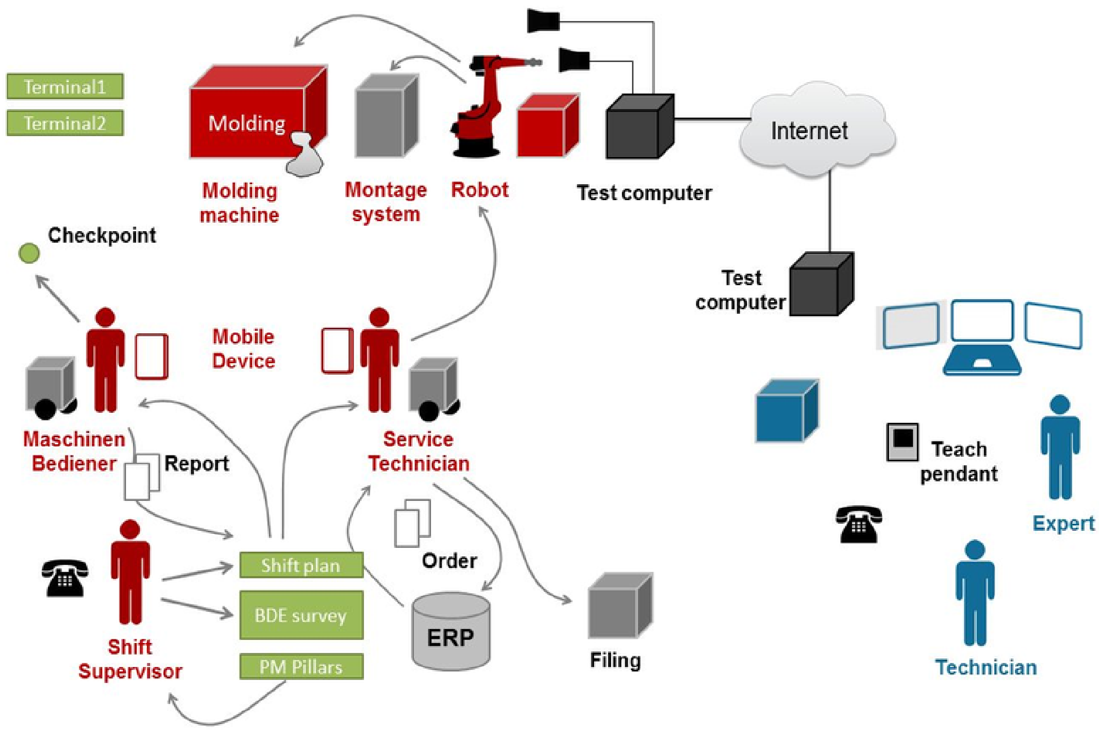


Figure 3.10: *Envisioned flow model*

Although many parts of the flow model and also their corresponding processes stay the same, the telemaintenance infrastructure is added. Different cameras are cable-connected to a test computer and provide different static viewpoints on the plant. The service computer at the facility is connected via the Internet to the service computer at the robot manufacturer. An expert has access to telemaintenance equipment like several monitors and a teach pendant. He or she can also ask some of the special technicians who work in the development department in case of special questions like optimization problems.

3.3.3 Scenarios

During the requirements engineering process, we focused on useful scenarios that can enhance the modern production process with telematics methods. One main task was to identify functions leading to a measurable progress, which means: Cost-reduction by using control theory, intelligent network applications and Augmented Reality. For each scenario a use case has been identified, that was implemented in the project, in order to measure the quality of realization.

These scenarios will be described in detail in the following sections. For a survey the six scenarios of the project can be subdivided (depicted in Figure 3.11) in local scenarios and remote scenarios, while scenarios with higher numbers build on the others. Especially for the sixth scenario all other implementations are needed.

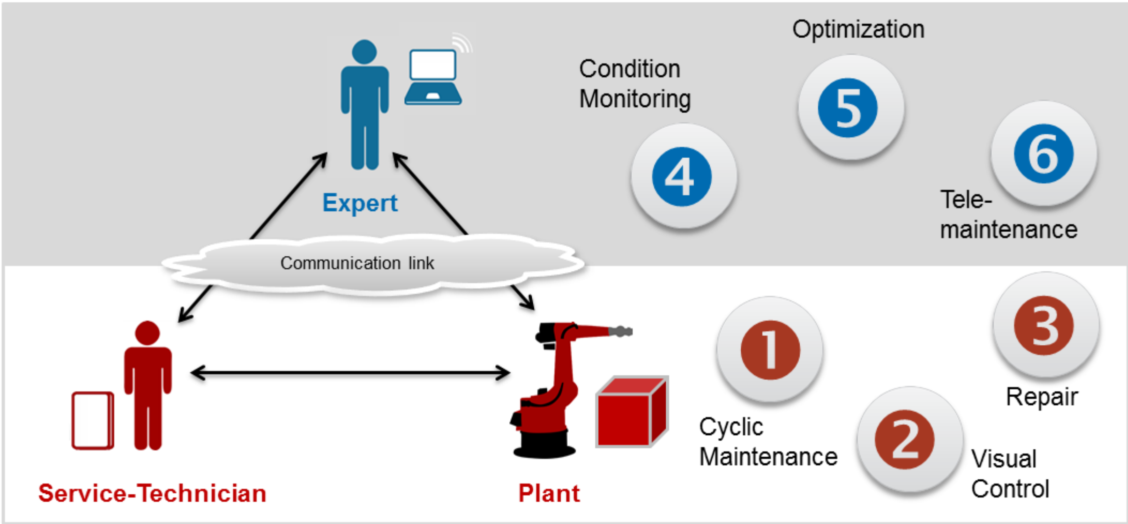


Figure 3.11: *Visualization of the local and remote scenarios*

3.3.3.1 Scenario 1: Planned cyclic maintenance with facility downtime

Frequent maintenance measurements are planned in regular intervals following the manufacturers' instructions. While different checklists are currently generated in a database, printed and processed by a service technician, this will be possible in the future with the help of the mobile device.

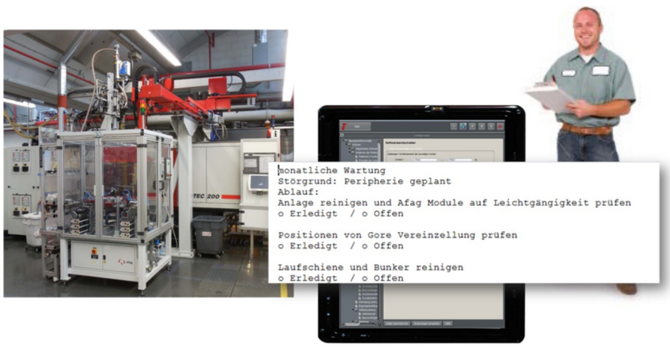


Figure 3.12: *Schematic visualization of scenario 1 of MainTelRob*

The use case in the specifications was the realization of the digital checklist for robot maintenance.

3.3.3.2 Scenario 2: Manual cyclic control on the enabled facility

A machine operator collects the produced parts from the facility following a defined walking route. On this route the operator has to survey the status of the running facility. We provide guidance with markers, digital checklists and task descriptions for these recurring tasks.

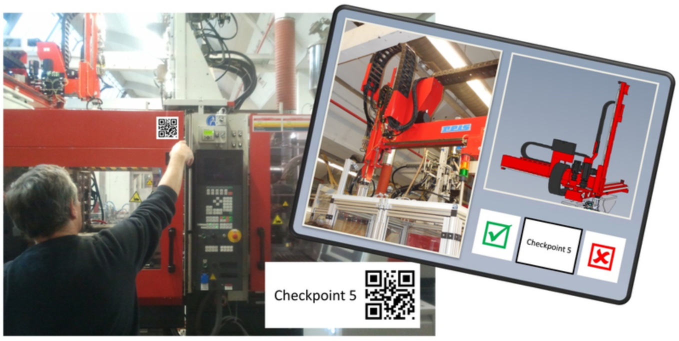


Figure 3.13: *Schematic visualization of scenario 2 of MainTelRob*

Use case: One walking route of the machine operation is realized on a mobile device with different Visual Control Points at the plant.

3.3.3.3 Scenario 3: Repair without external help

In case the facility is stopped due to a mechanical breakdown, a service technician needs to find the cause of the failure and repair it. We develop a support device for the debugging routine and the error correction, e.g. by giving pictorial instructions for detailed working steps.

Figure 3.14: *Schematic visualization of scenario 3 of MainTelRob*

Use case: A controller error is simulated. A service technician should perform the repair according to instructions on the mobile device.

3.3.3.4 Scenario 4: Predictive maintenance

The facilities run for long periods. Many failures do not occur suddenly, breakdowns mostly happen as results of subtle processes like abrasion. We aim to improve the condition monitoring, so that these processes can be detected and resolved. In the project we focus on monitoring the mechanical motor load.

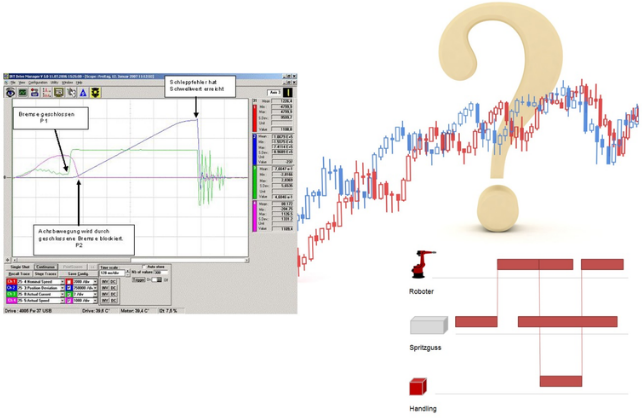


Figure 3.15: *Schematic visualization of scenario 4 of MainTelRob*

Use case: On a specific robot position the motor current is measured and used as a reference. For each production cycle, the current is measured and compared to the reference value. If the value is greater, a warning message is generated and displayed.

3.3.3.5 Scenario 5: Optimization with external help

Service technicians try to optimize the facility regarding cycle times, material or energy consumption. Here an expert appraisal is highly desirable but it is too expensive in most cases. In the project a remote interface is be developed to consult an external expert.

Figure 3.16: *Schematic visualization of scenario 5 of MainTelRob*

Use case: Optimization of the robot cycle with respect to time. The tool should visualize the workflow of the plant, so that different optimization measures can be discussed.

Scenario 6: Repair with external help

For difficult repairs, specialists from the robot manufacturing company are needed. The project provides a maximum amount of situation awareness for these external experts. With the help of remote-control, video communication and Augmented Reality they should feel like standing right in front of the facility, while giving advice to the local operators. We regard the use-case of motor exchange on a robot, a process currently not possible to support via telemaintenance.

Figure 3.17: *Schematic visualization of scenario 6 of MainTelRob*

Use case: The expert instructs the service technician remotely on a motor exchange. For this case normally an expert has to travel abroad to the plant.

3.4 Survey

As described above, all of the six scenarios fit into the rough project model. They therefore also fit in the extended Human Supervisory Control model which has been described in Chapter 2. In Figure 3.18 this model is applied on the practical context and can be enriched with the different required functionalities.

The larger circles with icons represent major operational fields:

- Circles with colored numbers are the MainTelRob scenarios
- AR Tab (see Section 6.5) and CAT (see Section 5.4) are supporting applications for scenario 5 and 6
- Scenario 4 is a local scenario on the one hand but on the other hand interacts strongly with scenario 5 and 6 (see Section 5.2)
- Synchronization and autonomy functions are displayed in grey, because they are part of the project, but not detailed in this thesis.
- Also security, network and control topics are a necessary part of the project but will be described in this thesis only in their context of human interaction.

The smaller circles with numbers represent communication paths and their content:

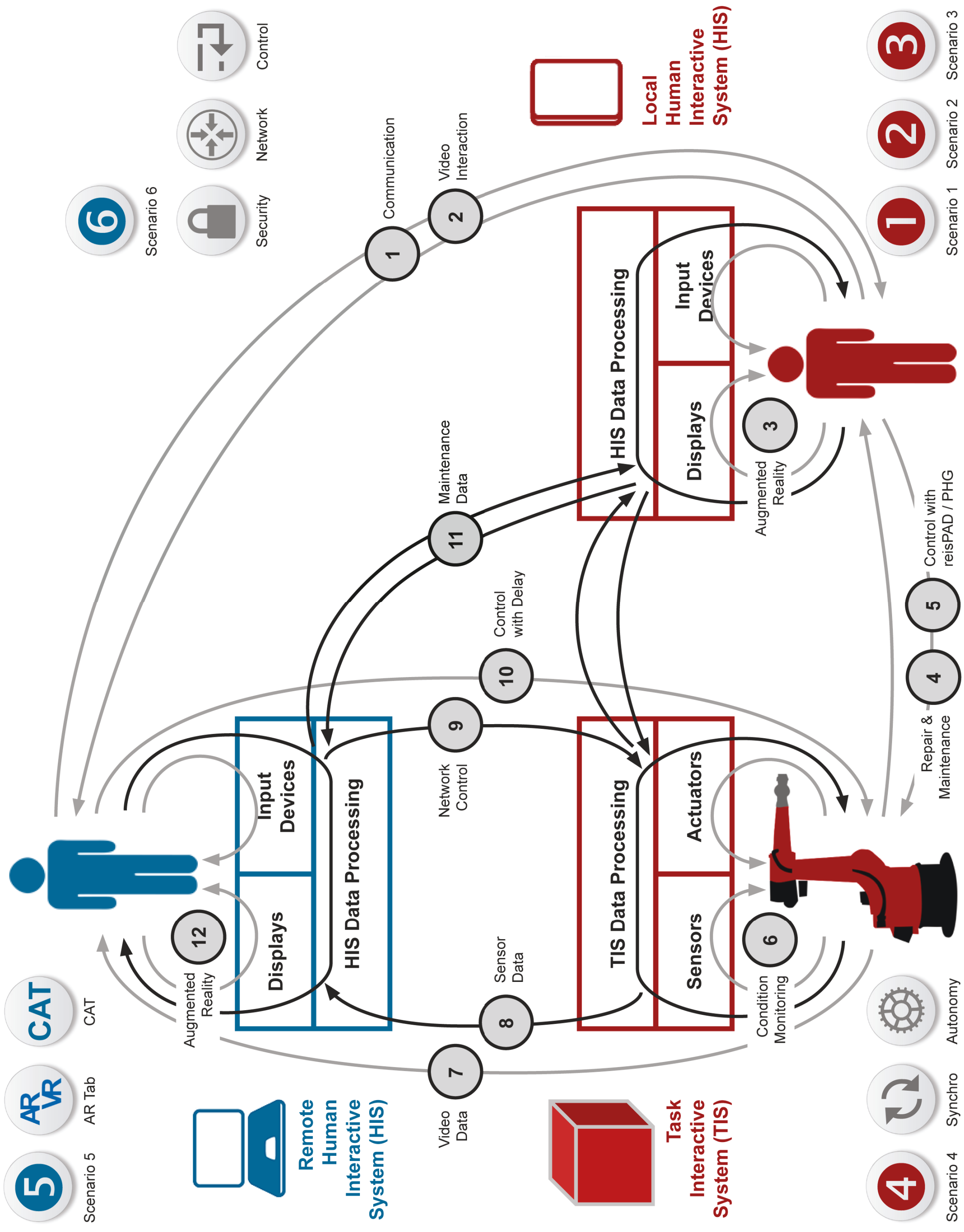


Figure 3.18: *Extended Human Supervisory Control and MainTelRob scenarios*

1. Communication: Oral communication (Voice over IP) between the two humans.
2. Video Interaction: A special functionality for cooperation on base of shared video data (see Section 6.3).
3. Augmented Reality (local): Methods for the visualization of instructions.
4. Repair and maintenance: CIL (cleaning, inspection and lubrication) but also more advanced repair operations for which production or robot data is needed, see Chapter 4).
5. Control with reisPAD or old teach pendant (PHG): Controlling and programming the industrial manipulator.
6. Condition Monitoring: Functionalities for a permanent comparison between the actual plant status and the optimal process (see Section 5.2).
7. Video data of the robot, which is transmitted via the Internet.
8. Real-time sensor data of the robot and the plant, also asynchronous data transmission.
9. Control of the network between both servers via bandwidth management.
10. Control of the robot via the internet with controlled delay.
11. Augmented Reality functions to enhance situation awareness.

On this base we can distinguish between different application fields and derive the required design goals, described in the following sections.

3.4.1 Mobile device for local maintenance

The first three maintenance and repair scenarios are mapped to the service technician with the Local Human Interactive System. As the teach pendant should be replaced by a mobile device, the hereby research questions and design goals are as follows:

- How does the change of interface affect the user?
- Which functionalities can be included in a mobile device?
- How can the documentation effort be decreased and concurrently the documentation quality be increased?
- How can current production data be used for maintenance?
- Is it possible to enhance repair instructions with Augmented Reality?

These questions will be answered in the next chapter.

3.4.2 Intuitive visualization of complex data

The fifth scenario (optimization) deeply interacts with the condition monitoring. Therefore, these two fields can be viewed together in some kind of asynchronous optimization process. The interface of the remote expert should answer the following questions:

- What is the best way to provide the information for problem analysis in a intuitive remote interface?
- How can software assist the understanding of the setting and the plant process (situation awareness)?
- How can the information of several plants be merged?
- Can expert knowledge lead to predictive maintenance models?
- How can Virtual Reality techniques be applied in teleanalysis?

These questions will be answered in Chapter 5.

3.4.3 Telemaintenance as human supervisory control

The sixth scenario covers the telemaintenance task and is therefore an integration of all the other functionalities under real-time conditions. In this complex case, the following questions will be regarded:

- Which infrastructure and architecture are required for telemaintenance?
- Which functionalities help to get an appropriate problem solution?
- How can the results of the other scenarios be used?
- Which features help the expert to gain situation awareness?
- What should be considered in order to optimize the virtual team performance?

These questions will be answered in Chapter 6.

Chapter 4

Mobile device for local maintenance

This chapter introduces own work with mobile devices in the active production used in the context of a holistic telemaintenance scenario. It covers the first three scenarios of project MainTelRob focusing on manual maintenance, inspection and repair without external help. These three activities should be supported with a tablet PC in an "all-in-one strategy". Research focuses on the usability of new teach pendant on base of a tablet PC, the design process of the applications and their acceptance in the real production environment. Another focus lies on the use of Augmented Reality for supporting repair operations.

4.1 Survey

With respect to the overall extended Human Supervisory Control model presented in Section 2.6.3, we regard in this chapter a classical Human Supervisory Control scenario with direct visual feedback. As visualized in Figure 4.1, this chapter focuses on the local service technician equipped with a mobile device representing the local Human Interactive System HIS. A technician can use the device to perform different actions on the robot. The first three scenarios of the project MainTelRob are realized with a mobile device. This chapter captures the local maintenance (Section 4.2), inspection (Section 4.3), usability of the tablet device for robot control (Section 4.4) and local repair without external help (Section 4.5).

Of course, maintenance, inspection and repair operations are already performed in the working environment. A main part of this chapter is about the process implementation in the industrial work domain, the results of the contextual analysis and the different steps of the design process (following Section 2.3.6). The research focus lies on the integration of the mobile device along several questions:

- How does the change of interface affect the user?
- Which functionalities can be included in a mobile device?
- How can the documentation effort be decreased and concurrently the documentation quality be increased?
- How can current production data be used for maintenance?
- Is it possible to enhance repair instructions with Augmented Reality?

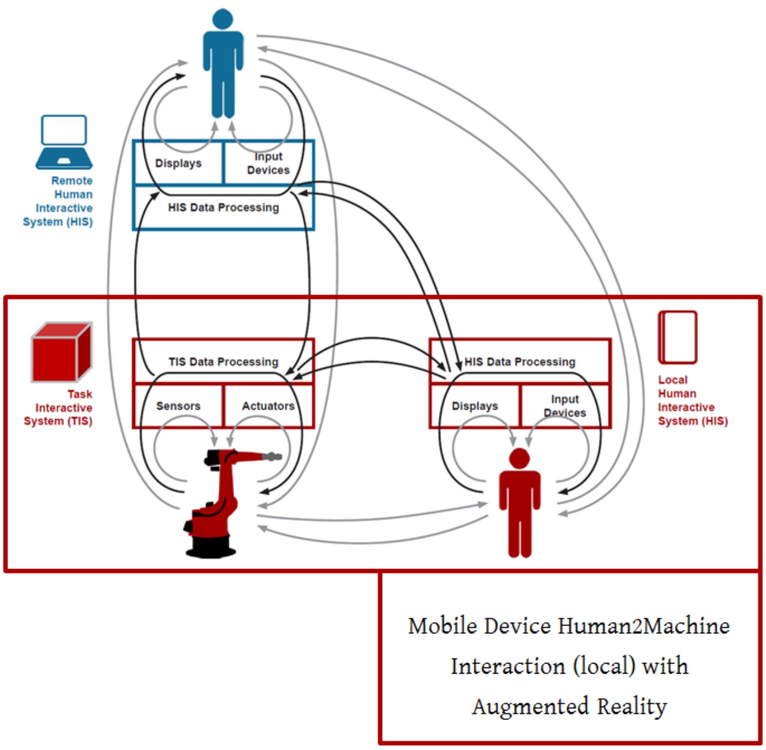


Figure 4.1: *Context of this chapter visualized in Extended Human Supervisory Control Model*

4.1.1 Motivation

Mobile devices like smartphones and tablet PCs are utilized widely in everyone's private life. They are also increasingly used at the workplace: 69% of the employees use private mobile devices in the enterprise; this is usually put under the statement "Bring your own device" (BYOD) [130]. Every "brain worker" has more than three networked devices on his or her work place on average in the year 2014 [131]. It is widely discussed (for example at the "Verband Deutscher Maschinen und Anlagenbauer" GMA) how to use those mobile devices in the industrial automation. The member survey of the GMA in 2012 stated "human-machine-interface" as the second most important impetus for ideas and inspiration for technical and socio-economic development in the automation industry for the next three years [7].

A study about production work in the future [8] asked deciders in enterprises about their impression on working conditions. There are several findings which are also important for this work. First, the human worker is still considered important or very important for production in the next five years (96.9% of the interviewed enterprises). Second, it is expected that in addition to the above mentioned "brain workers" also the production workers will use mobile communication techniques in the work context. Third, the utilization of mobile devices is expected to establish new possibilities in the use of actual production data while on the one hand reducing the effort for documentation and on the other hand enhancing its quality.

4.1.2 Goals

Starting point for the use of a mobile device for the MainTelRob project context was the innovation "reisPAD" of KUKA Industries. In their newly released "ROBOTstar VI" control system they upgraded the teach pendant (Figure 4.2 left) to a touch device (Figure 4.2 right). The "reisPAD" fulfills the same robustness requirements but is more flexible [132], as new functionalities can be added easily via software. This served as a base for the development of mobile device functionalities.

Figure 4.2: *Teach pendant (left) of a a Reis/KUKA Industries manipulator and a new reisPAD (right)*

The users of the mobile device are service technicians and machine operators who take care of the molding machine, montage-unit and of course the industrial robot. They need to execute numerous operations for which they partly need additional instruments (for example tools for maintenance). Thus the mobile device needs to be suitable for rough conditions and can be used with one hand. Every functionality required for maintenance operations shall be included in one device (all-in-one strategy [11]) which is schematically depicted in Figure 4.3.

With respect to the MainTelRob scenarios (see Chapter 3) we will cover the scenarios 1, 2 and 3 in this chapter, which are the so called "local scenarios". The required functionalities are maintenance (see Section 4.2) and inspection tasks (see section 4.3), but also robot control (see Section 4.4) and repair tasks with Augmented Reality (see Section 4.5).

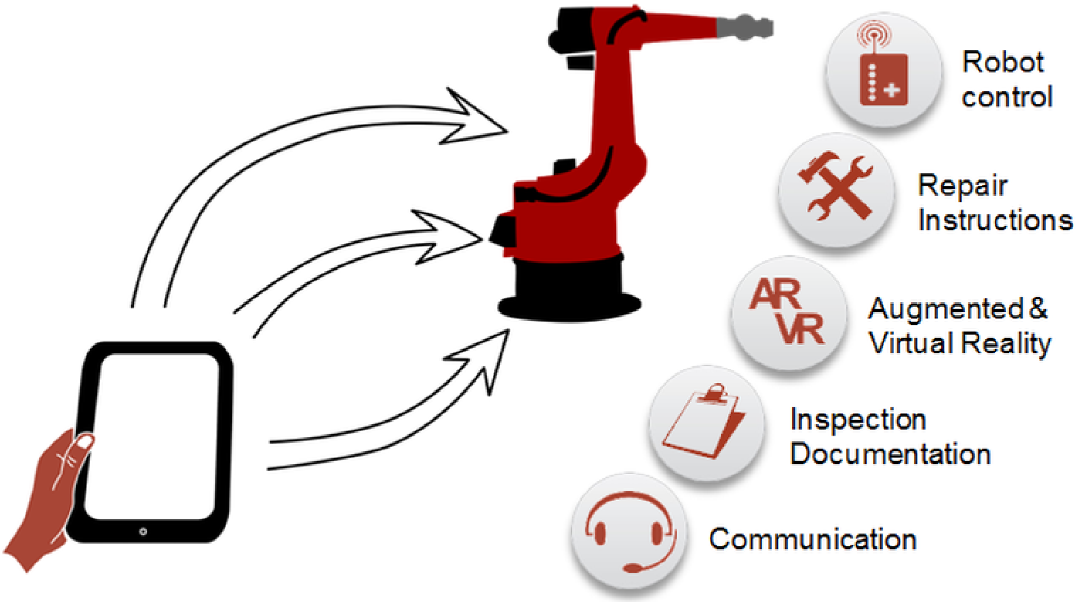


Figure 4.3: *All in one strategy for the mobile device*

4.1.3 Related Work

As described above, there is currently a lot of interest in the use of mobile devices in an industrial context [133], because they provide economic benefit especially in maintenance [134]. While in 2009 still few researchers used mobile devices in the context of condition monitoring and maintenance [135], the use of mobile devices in maintenance was already in a nascent state. Of course the use has increased between then and now. There has also been a wide use of touchscreen technology in the industrial context, so that there are already practical guidelines available (i.e. [136]).

Of course the research on Human Computer Interaction with mobile devices in general must be considered (compare for example [137] for a survey or [138] for a practical approach). One interesting feature of mobile devices is context-awareness. Systems able to adapt the operations to the current context without explicit user intervention can be of use in an industrial setting. This context can be but is not limited to location awareness [139].

Still, there are not many case studies on the specific requirements for the use of mobile devices in production environments. Although not used as widely yet, there has been a lot of research interest on Augmented Reality in industrial production environment.

A survey on possible use of Augmented Reality for production and maintenance [119] enables the discussion of advantages and disadvantages. The same authors provide an own library and implement a solution for maintenance of an airplane door.

There has been a concept and implementation of an AR application for the installation of a door lock into a car door [140] with the help of a See-Through-HMD (head mounted display, see Section 2.5.6.3) and optical tracking with 2D markers. The user interacts with the application using speech commands. The application provides the introduction of the manufacturing process visualized on the workpiece.

An AR system supporting users on a CNC machine [141] uses actual production data and visualizes them on a screen plane. The user is able to detect the working steps of the machine at the workpiece. AR can be used for the visualization of the working paths for industrial manipulators. There have been implementations for offline programming [142] or live programming with speech and mouse interaction [143]. Similar realizations have been provided by research of the university of Würzburg [144], also enabling a direct interaction with the robot with a tracking device [145].

4.2 Maintenance with a mobile device

4.2.1 Contextual inquiry and analysis

As described in Chapter 3.3.1, contextual analysis is very important for the understanding of the work environment and processes. We documented each use case in detail with video and writing. For a deeper understanding on the work domain, we include part of the documentation for scenario 1 (maintenance) here. We have asked a service technician of Procter&Gamble to undertake the normal maintenance process of the robot, which is performed twice a year. This process takes approximately 1.5 to 2 hours in addition to the mounting of the fall protection system. We have found it particularly interesting, because it provides a deeper understanding of the plant, the manipulator and maintenance. The process itself is shortened compared to our initial documentation and visualized in the photos in Figures 4.4 and 4.5.

While the plant is still running, the service technician checks the gears and the vacuum pump for quiet running. Then he puts the robot in a safe state and turns off the plant. Different preparation steps are required for the maintenance: opening of the molding machine, installation of a montage board and a fall protection system on top of the plant and the preparation of the personal tool and maintenance carriages.

At the beginning the service technician is standing on the montage board on top of the plant. The robot on this plant uses an automatic lubrication system, so that only the functionality of this central system needs to be checked and the lubrication of the cleaned parts is required. The tooth belts are tested for sufficient tension by hand. Further measurement instruments are only used if incidents connected to the belts have occurred.

1. Axis 1 CIL: cleaning, inspection, lubrication (first part):
    - Cleaning of guard rail and toothed rack.
    - Cleaning of bearing.
    - Control of guard rail, toothed rack and bearing on abrasion.

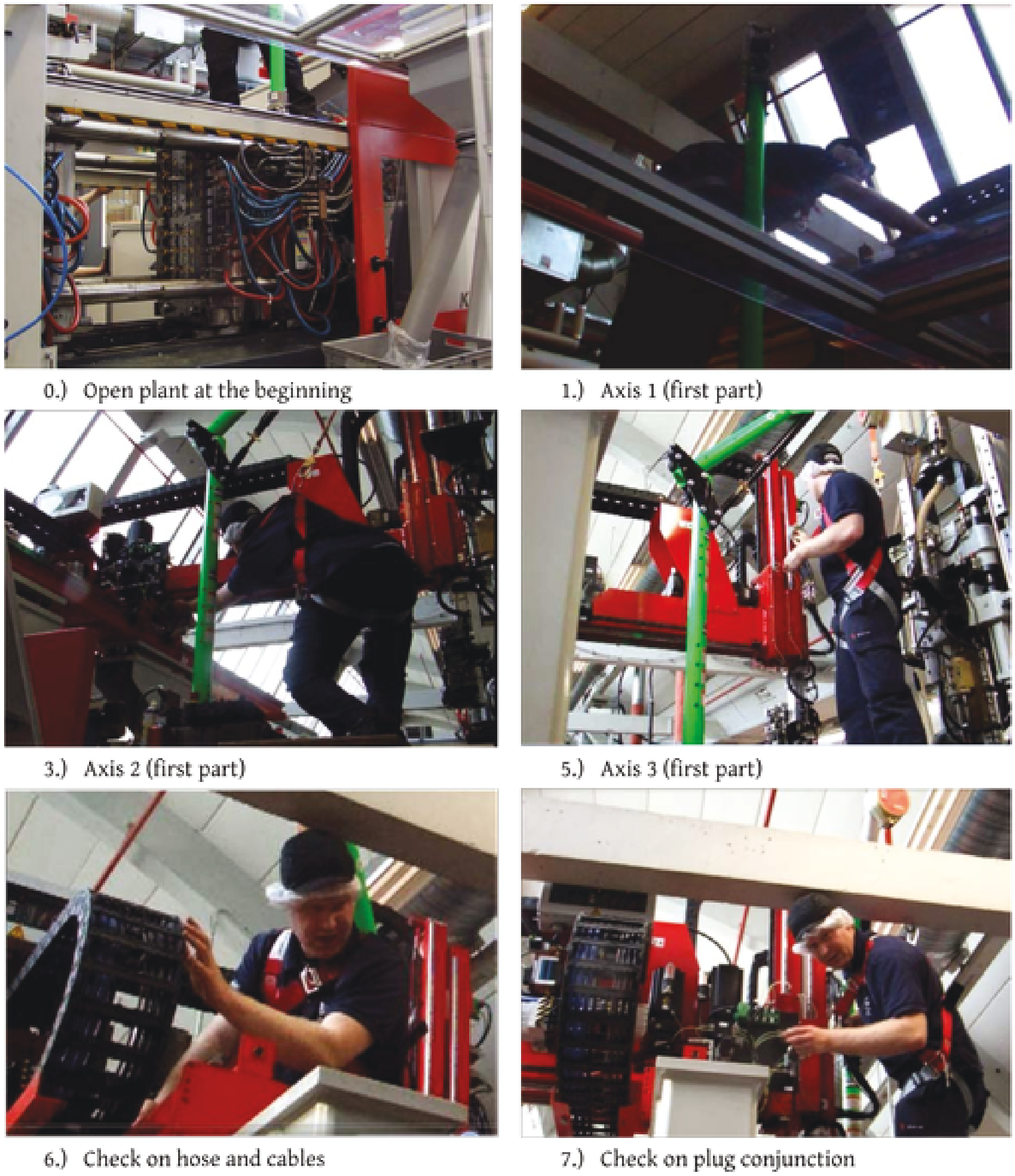


Figure 4.4: *Maintenance of industrial robot (visualization part 1)*

- Lubrication.

2. Move robot with the teach pendant so that axis 2 can be reached by the technician.

3. Axis 2 CIL (first part), see Axis 1.

4. Repositioning of the robot, so that axis 3 is on its highest position.

Figure 4.5: *Maintenance of industrial robot (visualization part 2)*

5. Axis 3 CIL (first part), see Axis 1.
6. Check on hoses and cables on abrasion and leakage (especially on parts with high stress for example at curvatures).
7. Check on plug conjunctions (for example motor plugs).
8. The service technician need to change his position, so that he is able to reach the rest of axis 1.

9. Second part of axis 1: cleaning, inspection, lubrication.
10. Check of the pneumatic system.
11. The service technician dismounts from the plant.
12. He tests the new battery and inscribes the values on the battery (they are changed every 24 month while checked every 6 months).
13. For accessing the switch cabinet, he needs to change his clothes (personal safety equipment: no hairnet but cap, safety glasses, earplugs and long-sleeves jacket).
14. Mounting of safety signals on the path (which is also used by self-driving robots for logistics).
15. Opening of the switch cabinet.
16. Changing the battery.
17. Test of the tight fit of modules and circuit elements in the switch cabinet.
18. Checking the air ventilation of the switch cabinet.
19. Performing a backup of the software.
20. Exchange of the CPU battery.
21. Importing backup, because of the exchange of the CPU battery all robot programs have been lost.
22. Dismounting the montage board.
23. Driving axis 3 in the molding machine.
24. Second part of axis 3: cleaning, inspection, lubrication.
25. Check on tight fit of axis 3.
26. Replacing the ladder, so that the second part of axis 2 can be reached.
27. Second part of axis 2: cleaning, inspection, lubrication.
28. Unscrewing the vacuum valves and cleaning of the filters.
29. Filling out the maintenance order form.
30. Dismounting of all maintenance equipment.

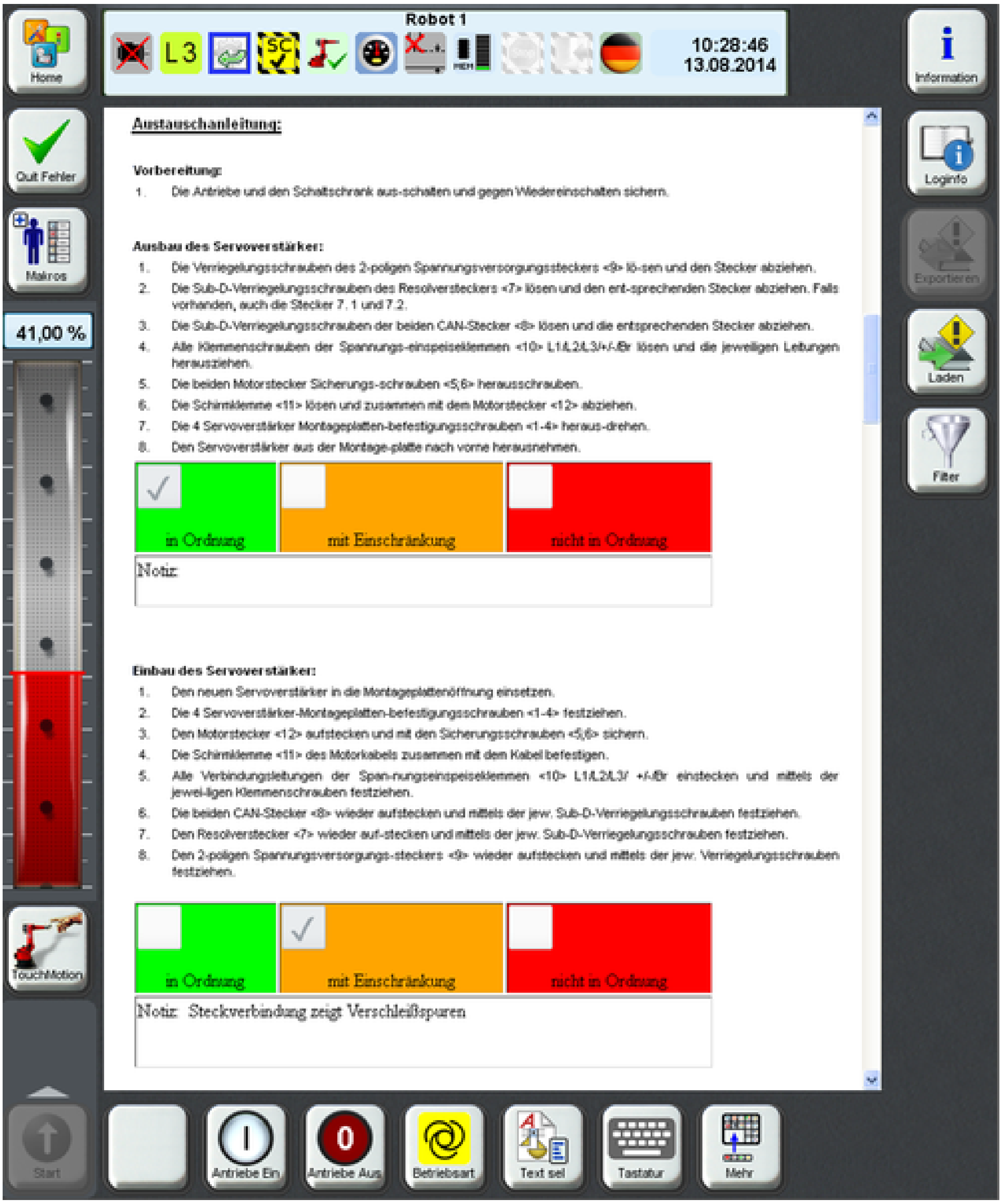


Figure 4.6: *For scenario 1 implemented checklist with checkboxes (representaion on the reisPAD of ROBOTstarVI)*

4.2.2 Maintenance checklist

In the previous section, the process of a regular maintenance operation has been described. After working for nearly two hours the service technician fills out the maintenance order form. It is easy to forget some steps during the maintenance operation, that is why scenario 1 of MainTelRob was the integration of the checklist in the reisPAD Software depicted in figure 4.6.

As described above, the use of the reisPAD presumes the robot steering "ROBOTstarVI", which has not been installed in the production environment yet. For that reason a demonstrator was built on a Microsoft Surface tablet PC on which the software of the reisPAD "Provis" was running which was connected to a virtual robot controller running "ROBOTstarVI".

In a simulated scenario the maintenance process above was performed with the electronic checklist on the reisPAD. All points of the maintenance order form have been included in an HTML document, which can be edited by the user. The advantage of having this information in the robot system is that they can also be addressed for tele-maintenance.

The functional extension essentially consists of the development of an HTML-Editor based on WebView in JavaFX. The editor enables the access to the document, its processing and its saving on a storage medium connected to the robot control (in the particular case the compact flash storage). The HTML checklist has been prepared, so that the service technician can include the annotations of the checkpoints with "Order executed", "Order partly executed" or "Order not executed" with a traffic light highlighting. He can also add free annotations in a text field. Additional electronic information like photos and videos can be included.

The data is stored on the robot control and can be addressed remotely via FTP for external use (i.e. in a production planning system or for telemaintenance).

4.3 Inspection: Visual control

Additional to scenario 1 which was described in the previous section, there are inspection operations which are performed during the active production. This is also a local scenario which can be performed without the Internet connection but is a precondition for the remote scenarios. The machine operator collects the produced parts from the facility following a defined walking route. On this route the operator has to survey the status of the running facility. He or she passes several checkpoints which are precisely defined and decides on the state of the checkpoints. The use case is the set of checkpoints for one specific plant and their defined control tasks, like checking the heat of the molding machine on one display or checking the nozzles for unwanted plastic remains. In the actual production environment, those information are presented on the BDE Terminal, a stationary interface at the plant. A software is used to check different control points.

4.3.1 Contextual analysis

The core of the development approach is a detailed contextual analysis. For the development of the vertical prototype of the mobile device, P&G provided original documents (for example service processes and forms), which have been implemented in the scenario. Using the material of this "work activity data", so-called "design

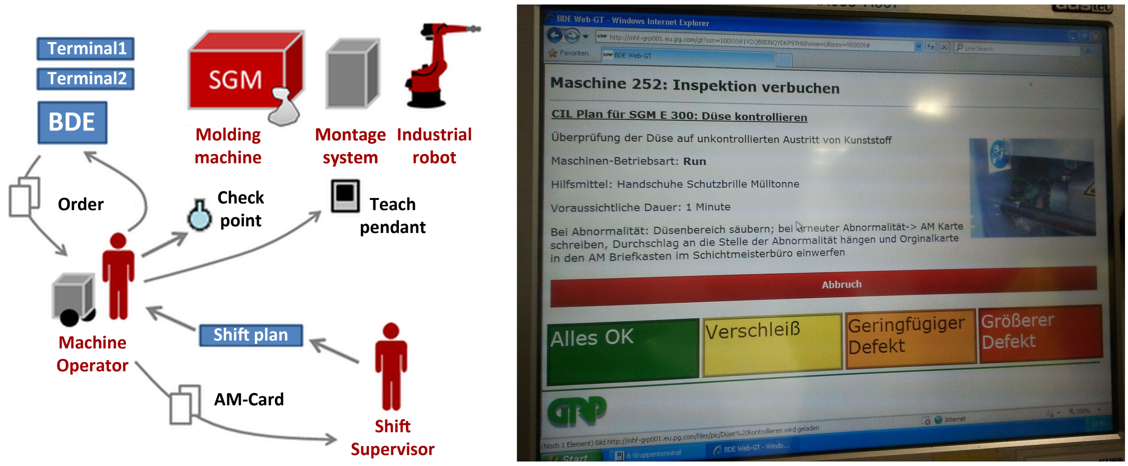


Figure 4.7: *Flow model (left) and stationary BDE Terminal (right) for Scenario 2*

informing models" [54] have been generated, which helped the developers to understand the user perspective.

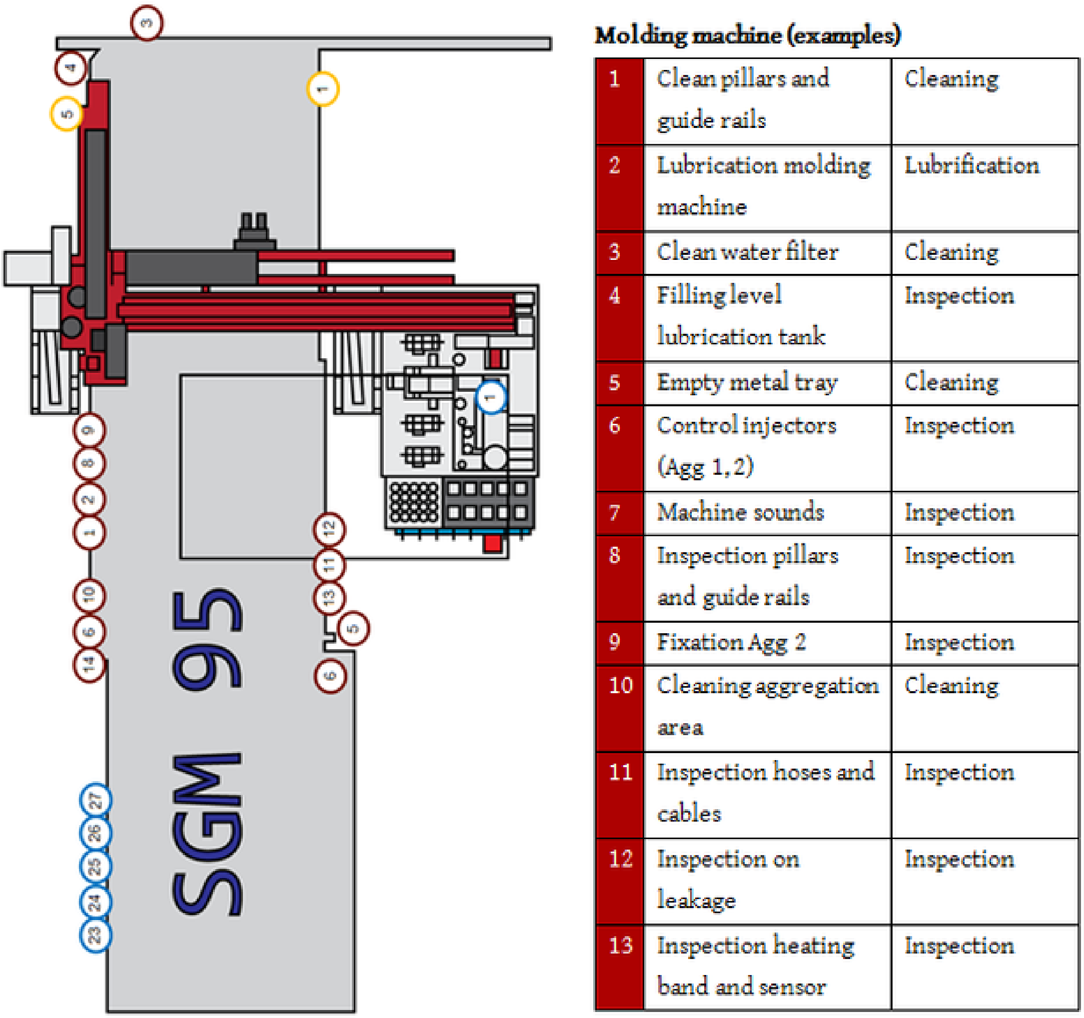
The general flow models of Section 3.3.2 already contain this scenario, which is also displayed in Figure 4.7 on the left. They only differ in one aspect: The machine operator now gets a mobile device instead of using the BDE terminal (Figure 4.7 on the right). This is a result of the interviews and the physical model of the different checkpoints around the plant which is depicted in Figure 4.7 and 4.8. The worker in the actual case should go to the terminal, get the instructions, go to the checkpoint, visually check the state, go to the terminal again, and acknowledge the status of the checkpoint there. The use of the tablet computer promises shorter paths and a higher documentation quality.

4.3.2 Prototypes

This example of a context-aware application for an industrial production environment was designed following the process of Section 2.3.6. In this section, we mainly visualize the different versions of the prototypes.

4.3.2.1 Paper prototypes

Following [146] and the practical workbook [147] we created and discussed several paper prototypes, which are shown in Figure 4.9 and 4.10. As already described in literature, working with paper is unfamiliar for software developers. This is even more true for the industrial domain. But it has become clear to the team, that it is a good method for discussion. The upright version of the application (Figure 4.9) seemed to be more convenient at first. But the team realized, that a camera picture had to be rotated and the person had to perform mental transformations. So the horizontal format in Figure 4.10 was chosen for further development.



1	Clean pillars and guide rails	Cleaning
2	Lubrication molding machine	Lubrification
3	Clean water filter	Cleaning
4	Filling level lubrication tank	Inspection
5	Empty metal tray	Cleaning
6	Control injectors (Agg 1, 2)	Inspection
7	Machine sounds	Inspection
8	Inspection pillars and guide rails	Inspection
9	Fixation Agg 2	Inspection
10	Cleaning aggregation area	Cleaning
11	Inspection hoses and cables	Inspection
12	Inspection on leakage	Inspection
13	Inspection heating band and sensor	Inspection

Figure 4.8: *Physical model for a part of the CIL (cleaning, inspection, lubrication) tasks for Scenario 2*

4.3.2.2 First software versions

In order to discuss the idea within the project team, we used a demonstrator on a Microsoft Surface tablet PC programmed with Processing [148], depicted in Figure 4.11. It used QR-Codes for the localization of the checkpoints and showed the same information as the BDE Terminal (see Figure 4.7 right) had provided. We used the zxing library [149] for detecting QR codes, which have been printed on regular paper and temporarily glued to the plant.

4.3.2.3 Prototype for formal acceptance and pluralistic walkthrough

A further developed prototype depicted in Figure 4.12 was created. The simultaneous display of camera video and instructions was one result of the paper prototyping, while the four different options for the check came from the original design of the

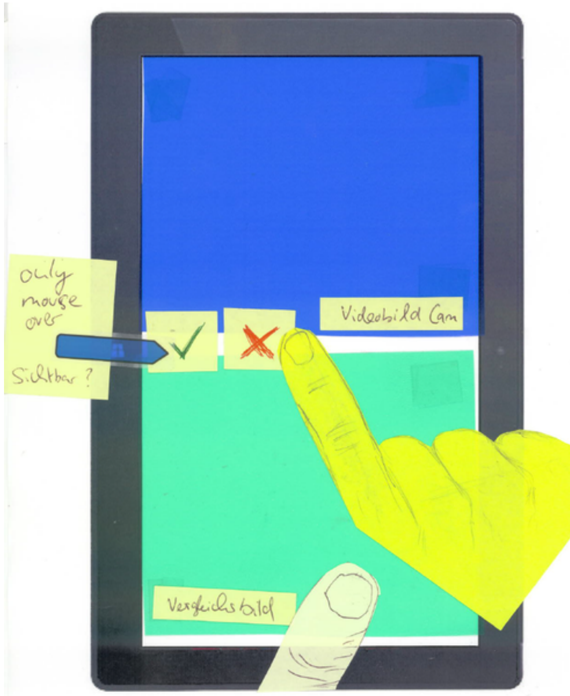


Figure 4.9: *Scenario 2 paper prototype first version*

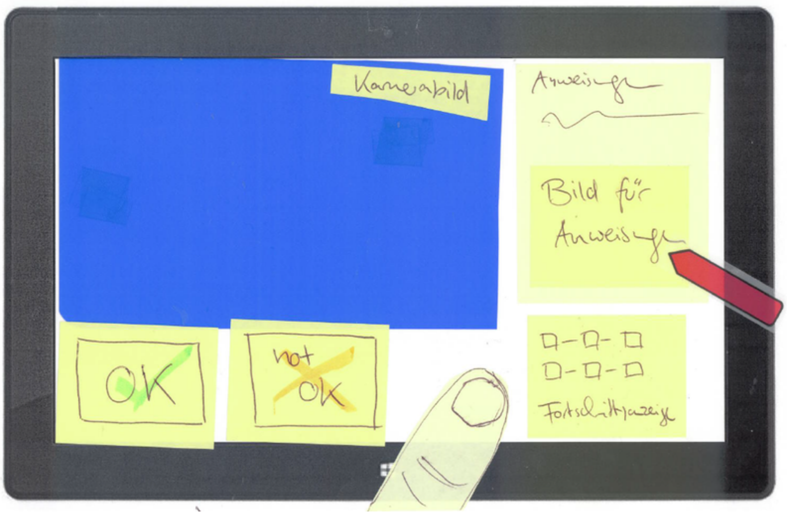


Figure 4.10: *Scenario 2 paper prototype second version*

BDE terminal (see Figure 4.7 on the right).

This prototype was tested with end users as part of the prototype acceptance. After the project members and the shift supervisor had tried the prototype, a machine operator was called who had never used a tablet computer before. The pictures in Figure 4.13 show him as he was using the prototype, which was also recorded on video.

Figure 4.11: *Scenario 2 first raw version first version*

Figure 4.12: *First prototype for manual cyclic control*

4.3.2.4 Final version

As described above, the same version of the software used in the formal acceptance was also used for the pluralistic walkthrough. It was performed with different printed versions of the software while the participants had to answer questions. The method has been described in more detail in Section 3.2.10. Some results on this scenario are presented in Figure 4.16. The participants could agree on the four different checking methods and in the end we used only two options.

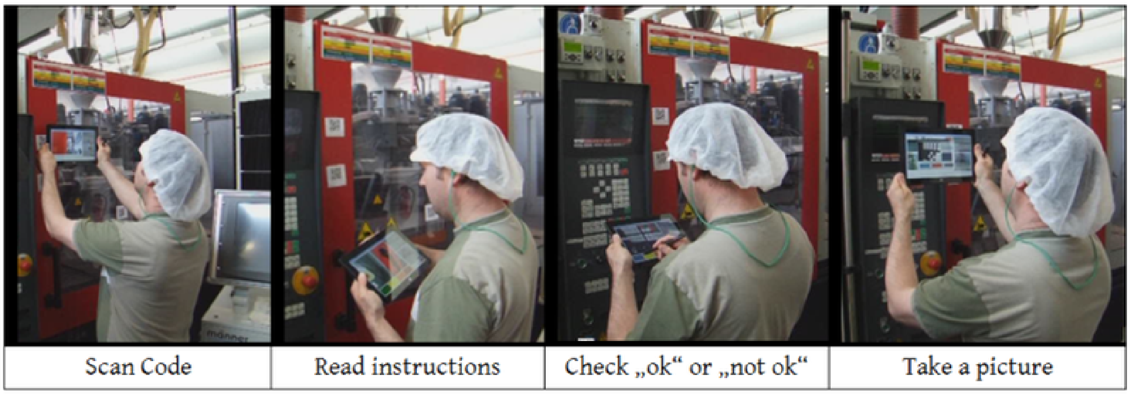


Figure 4.13: *Test person using tablet PC for CIL for the first time*

Some additional errors had been found in the pluralistic walkthrough, so that the interface was redesigned providing a "smartphone camera like" perspective. It is depicted in Figure 4.14. We also decided on showing either the camera video or the instructions. The application was implemented for an industrial tablet computer, the Pokini Tab. Gathered data was recorded within the overall web-based software framework, so that it could be used via remote access.

Figure 4.14: *Final prototype for longtime evaluation*

4.3.3 Evaluation

The last chapter already described the development process of the prototypes. The progress shown in the different development steps was already a result of formative evaluation during the development. Thus, we cover the qualitative studies merely in a brief way and focus on the techniques and the quantitative study.

4.3.3.1 Qualitative studies

a) Discussion with users on behalf of the reisPAD demonstrator

b) Official approval of the scenario with the reisPAD demonstrator

Figure 4.15: *Human Centered Design: Gathering user feedback*

All prototypes have been developed iteratively together with end-users, as described in the last section. After some design walkthroughs in the project team, the prototypes have been approved in a standardized process with the help of questionnaires (Quasi-empirical UX Evaluation [54]). All project participants were asked to perform a "Think Aloud" behavior when they used the prototypes for the first time. In a second iteration this was repeated with end-users which have not participated on the project yet.

Another big part of the qualitative evaluation was the pluralistic walkthrough as a combination of a guided design walkthrough in the team and a UX inspection by Human Computer Interaction specialists. The results have been clustered in a diagram, partially displayed in Figure 4.16.

4.3.3.2 Quantitative study

For the quantitative study a new tablet PC which fulfilled the robustness criteria for the industrial production was purchased (Pokini Tab [150]), the final prototype (Figure 4.14) was tested on this device, also with end users in a small prestudy.

The study has been carried out during the regular machine operator shift for several weeks. We recorded the working duration for each checkpoint and asked the probands to fill out the QUESI questionnaire [72], the NASA Task Load Index questionnaire [60] and a short interview after each shift.

After a short adjustment period the device can be used intuitively by people who were not accustomed to the use of portable touch-devices. This shows the values of

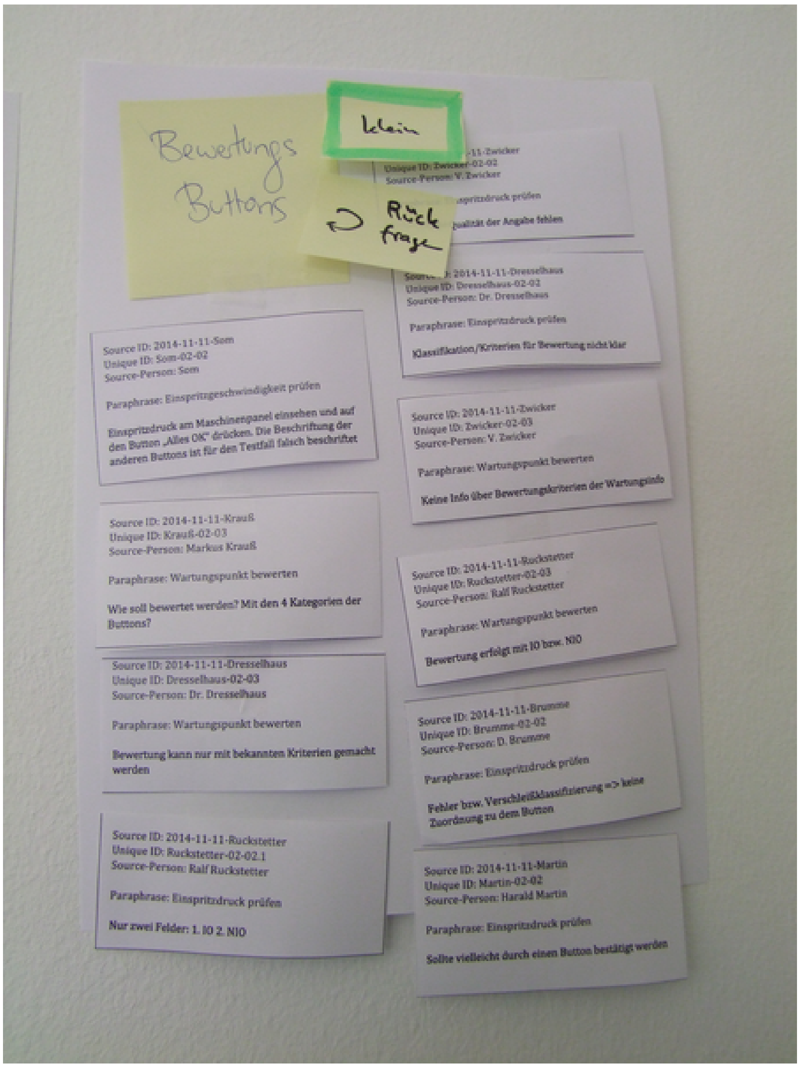


Figure 4.16: *Excerpts of the pluralistic walkthrough evaluation*

the QUESI questionnaire in Figure 4.17. The highest value is 5 out of 7, which still shows room for improvement. As there is no direct comparison to another software used for the same tasks, these results only show, that it is a promising approach.

The cognitive task load is relatively low, which can be seen in Figure 4.18. Only the time demand exceeds the others. That is mainly because the machine operators are used to perform all inspection points in a block. But also the qualitative feedback of the users pointed out the high amount of time used to scan the QR-Codes under different light conditions. So this clearly indicates further room for improvement.

The further qualitative findings on the question "Which things did you recognize positively in this shift?" have been promising: Thus, the photo function and the additional mobile information to the maintenance point and to the required tools have been seen very positively.

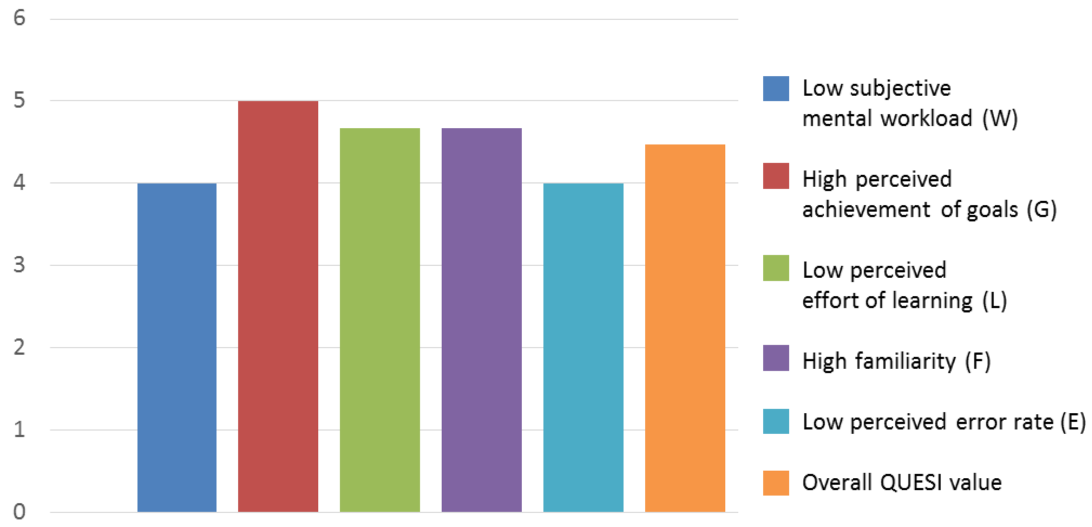


Figure 4.17: *QUESI values for scenario 2*

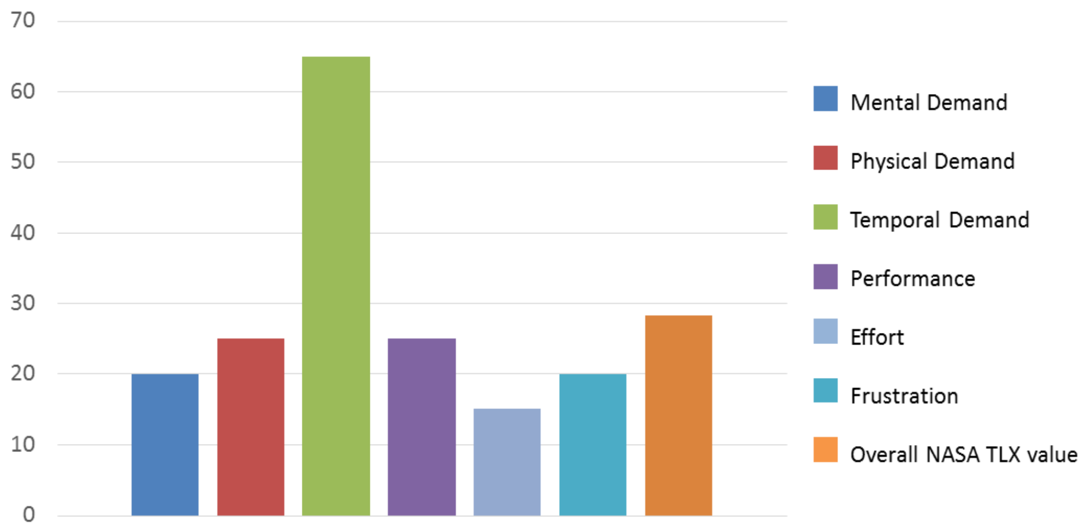


Figure 4.18: *NASA TLX values for scenario 2*

4.3.3.3 Questions in respect to the study

We tried to put the results of the estimations in proportion to the "Produktionsarbeit der Zukunft-Industrie 4.0" study [8] by asking the deciders of P&G the same questions.

The demand for mobile communication is seen equally, as can be seen in Figure 4.19 where the rating of P&G is highlighted.

The vast majority of the deciders interrogated by the study see a great potential for quality improvement through the use of a mobile device (Figure 4.20). The interviewee of P&G at Marktheidenfeld expressed only partial agreement. We asked

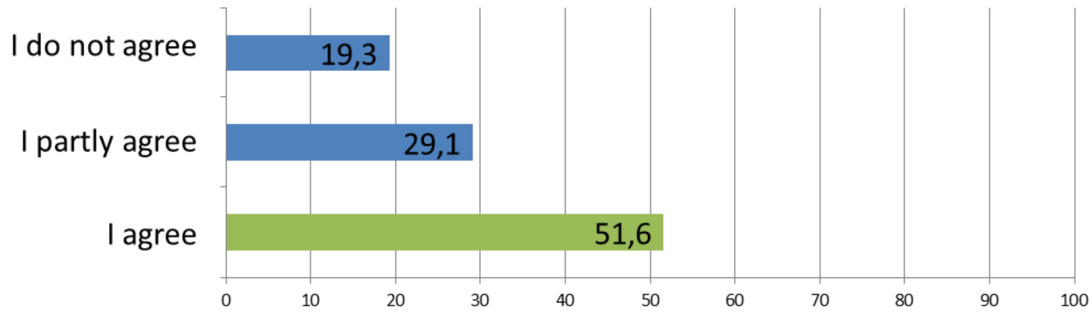


Figure 4.19: *Production worker will use mobile communication further in working context*

them if they think that the MainTelRob results will help to increase the quality of documentation, which they answered with "completely agree

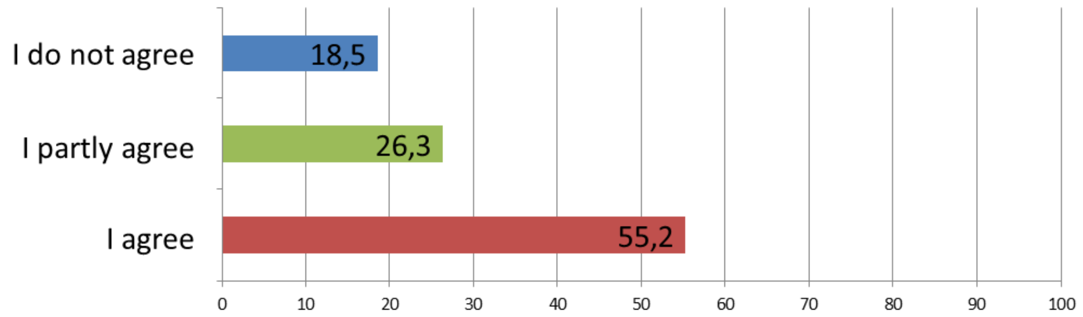


Figure 4.20: *The use of mobile devices increases the quality of documentation*

The study estimates a high documentation effort. The interviewees even expressed the opinion, that the effort is too high (69.8% agreed or agreed completely to the sentence "Today the effort for documentation of guidelines and norms considerably exceeds their benefit"). The interviewees in the MainTelRob context only agreed partly on that sentence. But they think along with the study's findings and see a great potential for the use of mobile devices to decrease the effort of documentation. They also expect the results of MainTelRob to help them to reach this goal (Figure 4.21). Procter&Gamble altogether plans an increased use of mobile devices in production and the project results will be used in an internal working group in Marktheidenfeld.

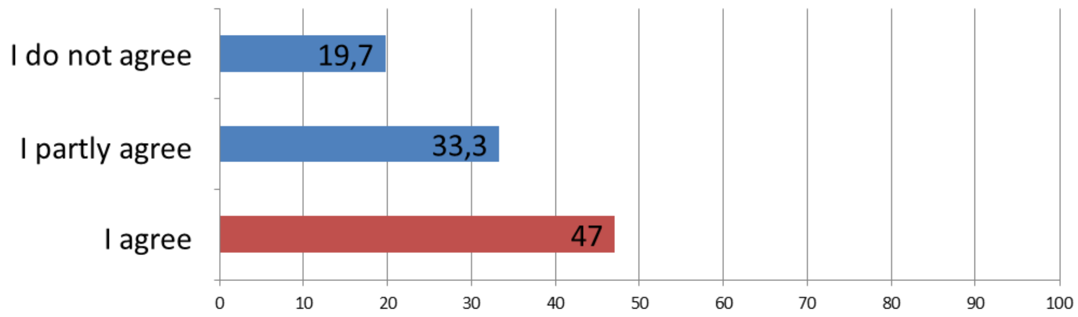


Figure 4.21: *The use of mobile devices reduces the effort for documentation*

4.4 A comparison of two hand-held devices for industrial robotic control

The present study addresses the comparison of two controlling devices for industrial robots, concerning their efficiency, effectiveness, and usability (user-friendliness). The so-called "teach pendant" ("Programmierhandgerät PHG) is fully operated by hardware keys; meanwhile, the "reisPAD", as its further advanced complement, is controlled via a touch panel.

According to our first impression, the PHG seemed more favorable for conducting simple tasks, whereas advanced functions are better accessable on the reisPAD. The latter is true despite disregarding current design rules as well as concepts of touch-operation during its development. These hypotheses were analytically tested using Nielsens Heuristics [67] and the Keystroke-Level Model [69], furthermore they were examined empirically in a study employing 16 probands. There were no significant differences regarding the efficiency of both devices, but concerning effectiveness and usability, the reisPAD showed a better performance, as expected.

For a long time the robotics industry had a tendency to disregard the comprehension of usability aspects in the developmental process of such control units. The reisPAD is therefore not only meant to be an advancement regarding usability, but an innovation in this industrial sector. This statement shall be tested here. The aim of the study is to analyze efficiency, effectiveness, and usability of the controlling devices named above. The study further asks if these factors permit conclusions to be drawn about the advertised user friendliness. In addition, the execution of selected tasks serves as comparison for user friendliness in the respective device's menu. The realized use cases for empirical and analytical operation were suggested by KUKA.

4.4.1 Methods

4.4.1.1 Analytical methods

To generate hypotheses, we decided to carry out a heuristic evaluation according to Nielsen [67] for both devices, as well as either an analysis following the Keystroke-Level Model below 'KLM', [68] for the PHG, or an adapted KLM-analysis for touch-controlled devices for the reisPAD, respectively.

Heuristic evaluation according to Nielsen. Heuristic evaluation helps to identify factors that could interfere with efficiency, effectiveness, and intuitivity in usage. Nielsen's heuristics are ten principles for assessing a system, for example "error prevention" or "visibility of system status". Detected errors are therefore grouped in five categories according to severity, these rank from 0 (no problem for usability, rather a problem in terms of layout) to 4, which is equal to a total lack of usability. Both heuristics evaluations were performed by five of the seven experimenters. For the PHG, there was total of 48 errors found (none in category 0, 15 in category 1, 21 in category 2, 10 in category 3, and 2 in category 4). Individual evaluations showed a compliance of 65.27%.

There were only 31 errors found for the reisPAD (3 in category 0, 8 in category 1, 10 in category 2, 7 in category 3, and 3 in category 4) with a compliance of 51,67%. Some of the problems found on the reisPAD were fixed by a software update before beginning the study with the probands, for instance incomprehensible error prompts ("2>=2") and a maloperation of the keyboard.

The PHG showed errors for all of the ten principles. In contrast, for the reisPAD interestingly no errors were found in principle 2 ("congruence between system and real life") and principle 6 ("recognition and memory"). This means that some conventions known from smartphones or other touch-controlled devices (e.g. menu navigation) have successfully been applied on the reisPAD, resulting in an easier handling.

Keystroke-Level Model. The Keystroke-Level Model was applied to examine efficiency of both devices in advance. The KLM contains predefined time specifications needed to conduct certain interactions with a system. For instance, the time necessary for a key stroke is given as 0.3 seconds by this method. Because there is a fixed path for every interaction with a system, this method allows to determine a time span for every action or task using a standardized model. Two use cases were generated, operated and compared for both devices.

For the first use case, the language of the control unit had to be changed. According to KLM, on the PHG this takes 12.5 seconds, whereas only 8 seconds are needed on the reisPAD. In practice, time spans of 10 seconds (PHG) and 6.5 seconds (reisPAD) were measured. All measurements include the respective waiting time while loading the language version.

The second use case was to write a small program to return the robot arm to its initial position. This took 29.6 seconds on the PHG, and 16.7 seconds on the reis-

PAD. Despite enclosed instructions, there were high deviations from the theoretical values while performing the use cases. (PHG: 58 seconds, reisPAD: the experiment was aborted after 40 minutes, because of the simulation permanently showing error messages due to the former version. Thus, saving and execution of the program was not possible.)

4.4.1.2 Empirical methods

Participants. The participants of the study were 16 students (11 male, 5 female, average age: 21.06 years) of the Julius-Maximilian-University of Würzburg from the study courses Human-Computer-Systems (14), medial communication (1) and teaching post/special needs education (1), during their first to 7th semesters. They were recruited personally by the evaluators.

Material. The study was carried out using simulations of both control units as well as a thereby controlled simulation of a 6-axis articulated robot on a touch-sensitive computer. Experiments were recorded with a video camera and mobile eye-tracking glasses. For further data collection an ISONORM 9241-110 questionnaire [71] was used for each condition. Achieving a high score in one of these indicates an overall positive perception of usability. Moreover there was a SEA-scale [61] to be completed for every subtask to measure the test person's subjective effort. A low score on this scale from 0 to 220 shows a low level of effort. Additionally, relevant personal data was collected using a demographic questionnaire, such as previous experience with devices like the tested ones. The questionnaires were available on a notebook. For the main investigator a guideline was provided.

Experimental design. A Within-Subjects-Design was used for data acquisition. The manipulated independent variable here were the control units. To exclude learning effects, the probands alternately began the study with the PHG or the reisPAD.

Measured criteria. Data acquisition can be grouped into the criteria effictivity, efficiency, and satisfaction. To measure effectiveness, the number of solved subtasks was counted, and the accuracy of reached spots was compared to the positions marked by the investigators beforehand. Efficiency was determined firstly by the time used by the robot to conduct certain tasks with different methods, secondly by the relative time watching the robot during the task (blind handling), and thirdly by the subjective effort of the proband during the tasks. For rating satisfaction, ISONORM-questionnaires, interviews, and protocols with statements of the probands were used, respectively.

Hypotheses. The following six hypotheses are based on the results of the analytical method and inspired by suggestions from our visit at KUKA:

1. If the reisPAD allows a more effective handling, there has to be a significant

difference between the number of subtasks solved under different conditions. Intuitivity of menu navigation and menu structure of the devices was rated according to accessibility and correct and distinct naming of menus and functions, as well as the logic designation of functions to menu items.

2. Industrial robots have to administrate their work steps with extreme precision. Therefore the ability to navigate the robot arm to a desired location is crucial. Control units should enable this process using only little time input. This should be tested empirically with the following hypothesis: If the efficiency of the operating method was improved, deviations in the position of reached spots should differ between the conditions.

3. If the robot's operation is more efficient on the reisPAD, then there is a significant difference in the robot's operation time in axial and Cartesian systems between the conditions. To minimize financial loss, malfunctions of a robot have to be eliminated as fast as possible. This hypothesis examines to which extent the controlling units allow the rapid operation of a robot.

4. If the reisPAD and the PHG are equally usable via blind handling, there is no significant difference between the relative times watching the robots during operation from two different points. Blind handling accelerates the robot's operation and saves money by removing problems and programming new work flows. An intuitive handling of the controls is mandatory in this process. This hypothesis evaluates how blind handling is influenced by touchscreen use, placement of buttons and regulators, and interaction with these.

5. If the reisPAD is more efficiently usable and operation is perceived less exhaustive, then there should be a significant difference in the average SEA-scales between the conditions. This hypothesis analyzes the mental strain of users during handling the devices.

6. If the reisPAD allows a more comfortable handling of the robot, results of the ISONORM-questionnaires should differ significantly between the conditions. They measure the test person's satisfaction concerning the control units. The examined sectors of the ISONORM-questionnaires model the quality of basic system requirements and enable conclusions about the overall usability of the systems.

Tasks. There were four tasks assigned to each device. First, the robot's motor had to be started and the operation speed had to be adjusted (1). Following that, the system language had to be changed to English and then back to German (2). The third task (3) included steering the robot from the reference position to a predefined red spot in the integrated box, using axial operation mode. Then the robot had to be directed to a second, green spot. Afterwards, operation mode was set to the Cartesian system (movement control in the coordinate plane). The robot had to be steered back to the red and then to the green spot.

Experimental procedure. Empirical data acquisition was performed between January 21st and 27th in 2016 in building 82 (Hubland Nord). Every test person had 60 minutes time. After a mutual introduction, data privacy statements and declaration of agreement were reviewed. A brief explanation of the experimental procedure was given and test persons were asked to complete a demographic questionnaire. The eye-tracking glasses were calibrated. Subsequently, the menu tasks (1) and (2) under the first condition were addressed, along with the respective SEA-scales, and test persons were given a minute to get used to the robot control units. After the main investigator reset the robot to its reference position, task (3) had to be dealt with. Task (4) was addressed following adjustment to the Cartesian operation system by the investigator. There were also SEA-scales to be filled in directly after finishing tasks (3) and (4). Following completion of the tasks, an ISONORM-questionnaire had to be revised for each device that had just been handled. Meanwhile, the simulation for the next condition was started. The same sequence was applied therein. When a test person was not able to solve a task, he/she could cancel; or when a time limit of one minute was exceeded (or two minutes for task (3) and (4)), the experimenter offered cancellation. The experiment was concluded with a half-structured interview about handling, the test person's preference, and the devices themselves; further the probands were presented the devices' hardware version. All experiments were journalized handwritten.

4.4.2 Results

4.4.2.1 Statistical analysis

Seven dependent t-tests were calculated, all descriptive values and the exact results can be seen in Figure 4.1. Because several tests were done using the same data set, the alpha level of $\alpha = .05$ had to be adjusted according to Bonferoni [151]. This is why p values are compared with $\alpha = .007$. Below, the preceded hypotheses are tested on the basis of the results.

Table 4.1: *Results from t-testing*

	PHG		**reisPAD**					
	M	SD	M	SD	df	t	p	d
Subtasks solved	0.72	0.45	1.69	0.57	15	5.783	<.001	1.879
Accuracy of points	29.54	23.13	61.33	105.67	11	1.120	0.287	0.440
Op. time axial	168.44	64.22	176.69	63.75	15	0.553	0.588	0.129
Op. time cartesian	93.50	51.89	116.38	77.71	15	1.791	0.093	0.311
Eye contact device	0.25	0.12	0.21	0.18	8	1.252	0.246	0.272
SEA-scale average	97.13	24.50	62.73	25.73	15	4.917	<.001	1.369
ISONORM score	3.02	0.82	4.73	1.14	15	5.165	<.001	1.718

Notes:

- M equals the arithmetic center.

- *SD* equals the standard deviation of measurands and a dimension of variation.
- *df* equals degree of freedom, here equal to the number of probands minus 1.
- *t* equals t-value.
- *p* equals p-value, it shows the probability to gain such a test result or an even more extreme, if the null hypothesis is true.
- *d* equals effect size and indicates the size of the statistical effect. Up from 0.2 there is a weak effect, from 0.5 a medium, and from 0.8 a large effect [152]. Sea-scales can show a score between 0 and 220; the higher the score, the higher is the subjective mental exhaustion. The ISONORM total score can have values between 1 and 7; the higher the values, the better the demands had been met.

The first hypothesis assumed that the number of subtasks solved is different between the conditions PHG and reisPAD. Statistical analysis confirmed this hypothesis and showed that more tasks had been solved on the reisPAD. The second hypothesis argued that the reached points are more accurate on the reisPAD, which could not be verified by t-testing. Neither could a significant difference be seen regarding the required operation time in axial or Cartesian systems. To examine blind handling, the relation between time watching the control unit (measured by eye-tracker) and the total operation time was determined. Results confirmed that in line with the hypothesis, blind handling still is possible even without haptic feedback on the touchscreen and that there is no significant difference to the PHG.

Regarding the SEA-scales, the reisPAD can be used with substantially lower human effort, which confirmed the initial hypothesis. Likewise, the reisPAD achieved higher scores in all seven categories of the ISONORM-questionnaire. As expected, total scores showed significant differences between the two devices. Qualitative analysis of the interviews and accordingly the recorded comments of the probands, backed up the results from the ISONORM-questionnaires. 12 of 16 probands stated to prefer the reisPAD over the PHG; further there were as much positive statements registered about the more intuitive control of the reisPAD. However, two probands preferred the PHG, because the robot's operation axes were labeled more clearly. This perception was shared by overall seven participants. The remaining two participants were not able to specify a preference.

4.4.3 Suggestions for improvement

Although the reisPAD performed better, during this study several ideas for improvement appeared with regards on usability. The full capacity of the bigger display size of the reisPAD is used and it offers many new functions, but that can sometimes lead to a confusing view. Many important functions such as the alteration of operation modes therefore were not noticed without instructions. In standard operation mode, there are ten regulators, four of which are not assigned for controlling a

6-axis articulated robot. To facilitate usability, they can be underlaid with grey colour. Moreover, the axes in an axial coordinate plane are assigned with numbers between 1 and 10, a better solution would be to depict the axis on the robot to relieve the working memory. Another operation mode is the graphic presentation of control dials, which can be used by circulating movements on the display. In our experiment we noticed usability problems even after explaining the functions to the participants. To support this function visually and facilitate navigation, a circle could be shown when using a dial. Since many errors were not noticed by the users, error messages could be displayed bigger and linked to solutions or references to the online guidelines.

4.4.4 Self-criticism and methodical review

Retrospectively, many aspects of the execution of the study show potential for improvement. This can be differentiated into reviewing the method as well as criticism towards the evaluators themselves.

4.4.4.1 Methodical review

As mentioned above, the study was conducted using computer simulations of the respective control units. Since only the reisPAD is used via a multitouch display in reality, whereas the PHG has physical keys, criteria like blind handling were not representative during navigation in the simulation. Furthermore, selection of participants contributes strongly to infringement of external validity. In the absence of experts, students of different courses of study and semesters were thus recruited, thereby missing evaluation within relevant specialist groups. By changing the starting device during the study, a learning effect should overall be prevented, but such a permutation is lacking for the reisPAD's control methods (two axial operation modes). Thereby the internal validity was infringed and an assessment of the also surveyed second axial operation mode would not be reasonable. Another problem occured from the application of the ISONORM-questionnaire, which normally rates satisfaction. Besides topics relevant for the study (e.g. conformity with expectations) it also covers aspects of control that were of no importance for the participants (e.g. error messages).

4.4.4.2 Self-criticism

Due to communication issues on the part of our industrial partner KUKA our actual preliminary time was shortened, leading to a more superficial preparation than required. The guidelines for the main investigator showed some shortcomings. These missing details in instruction resulted in partly different implementations of the tasks given by the investigator. For instance, it was noticed in the debriefing, that evaluators did not cancel the task after the same time (scheduled as two minutes), when a test person did not solve the task without help. To avoid such problems and thereby the complication of data assessment, a detailed investigator's guideline and possibly an augmented preparation for the study's execution would have been advisable. Another factor compromising a smooth and stress-free experimental procedure was a compendiously time interval between testing each test person, despite pretesting (about 45 minutes study, 15 minutes interval). Since technical problems frequently occurred already before or during evaluation, the scheduled break often was insufficient. Extended pretesting could have shown up front that this break was measured too short.

At last, the majority of the critical issues mentioned above resulted from an inadequate preparation due to a lack of preliminary time. The main cause lies within the factors determining study evaluation stayed ambiguous over several weeks (test person: students or experts, location: laboratory or at KUKA's facility). The deadline for definitely determining these factors was shifted repeatedly. The lack of time in study evaluation could have been avoided by reacting and proceeding more consequent at this point. In summary, the study still showed satisfactory results despite obstacles and complications during the cooperation with KUKA.

4.4.5 Discussion

Aim of the study was to evaluate the difference between efficiency, effectiveness, and user satisfaction on the PHG and the reisPAD, respectively. Amongst others, we assumed that distinct points can be reached more accurately with the reisPAD, a fact that could not be verified experimentally. The times required for operating tasks further did not differ significantly between the two devices, in contrast to the hypothesis. However, already during experimentation the probands revealed distinctive opinions concerning the different control units of reisPAD and PHG. Thus, a part of the probands preferred the labeled axes of the PHG: "It makes more sense with X and Y written on it" (quotation of a test person). Then again, others perceived the more intuitive control of the reisPAD to be easier handled. It was though noted several times that presumably the robot control is simplified when used daily: "Labeling of the axes would be pretty cool. But if you work with it, I guess you figure it out." (quotation of a test person).

In conclusion, most of our hypotheses were confirmed. Especially regarding satisfaction and efficiency the reisPAD performed better than the PHG. It remains to consider that in our study only a small part of functions of both PHG and reisPAD were examined. On the one hand, this was due to the large extent of functions on both devices, on the other hand the participants, being no domain experts, first had to be introduced to the systems. Advanced functions such as application of macros, and basic functions such as programming could not be tested without exceeding the reasonable time frame.

To enhance representativeness and validity of this study, it is feasible to examine a broader spectrum of functions. Interviews may help to identify possible localizations of testing coverage by determining priorities, incorporating importance and frequency of usage. An essential requirement for a conceptional replication of this study is testing participants from a relevant target audience.

4.5 Repair with Augmented Reality

Repair is more difficult than the above covered inspection tasks. To reduce the load of people working in production, a lot of research is performed. Particularly for working on complex tasks the use of mobile devices may facilitate the work. If there are repair or maintenance tasks where the technician needs further support, a system would be able to provide additional information. As Augmented Reality is a good way for providing this additional support (see Section 2.5.6), this section mainly is about the system "Eyestruction" which is compared on mobile devices with different interaction concepts.

4.5.1 Contextual analysis

As described in Chapter 3, the use case for scenario 3 was a "controller exchange". The exchange of a malfunction servo amplifier (also called "controller") with a new one is performed as a standard failure handling. There are some error messages of the robot, which indicate "controller error". The technician will then try to change the controller. If the error is gone after the repair operation, the plant is ready to restart. If the error is still present, it has occurred due to a different reason, mostly broken cables.

Each servo amplifier controls one robot axis and they are located in the switch cabinet (blue boxes in Figure 4.22). They are interconnected with different cables and plugs and can only be exchanged after a complicated extracting procedure.

The correct connection of the plugs in the right order is mandatory for installing or dismounting the controller. If a cable is connected the wrong way, short-circuits may occur inside the plant or the robot may drive erroneously. This may cause a complete breakdown of the plant, the destruction of expensive robot parts or in the worst case injury to people. In order to protect property and personnel, it is indispensable to omit failures and increase the diligence of the maintenance.

In the precise case of the controller exchange in the switch cabinet, apprentices or new employees need a longer training in order to enable them to perform this maintenance action on their own. As already a small mistake may lead to severe consequences, their work needs to be surveyed by a trained employee, at least the result needs to be reviewed. Supporting an untrained worker by equipment would help to minimize failures and redundantize the review in the best case. It can also be used for training purposes.

4.5.2 Failure diagnosis

Failures which reduce the plant availability generally occur unexpected and require a fast reaction by the machine operator or service technician to work against an imminent production breakdown. The cause can be multifaceted. If the failure occurs in a critical path, the whole production may need to stop. Due to their experience, the service technician can often find the cause and clear the faults.

Figure 4.22: *The switch cabinet with integrated servo amplifies (blue)*

As the interviews with both the robot manufacturer and the plant operator indicated, hardware malfunctions inside the robot are extremely hard to find, because the cause cannot be seen directly. Especially if the malfunction is located in the switch cabinet, an effective support for failure localization and removal is needed.

In scenario 3 of MainTelRob (see Section 3.3.3) the controller failure of axis three is used as a specific use case. If there is any internal robot failure, an error is displayed in the teach pendant with error number and a short message. Due to the limited space of the old teach pendant, the service technician either knows how to solve the failure indicated by the error or he has to consult the manipulator documentation (usually available as file folder in the office). In the old teach pendant there was no space for an error history either, if an error message has been acknowledged by the user, it was difficult to find out which errors had occurred.

The reisPAD wanted to solve this problem. By touching the status line (see Figure 4.23) a failure history window can be opened.

With the new functionalities, a documented failure list can be linked with the "logbook app". The logbook needs to be opened and the message needs to be highlighted (by touching it). With the I-Button (Info-Button), a failure description with possible causes and short instructions for solving the failure can be displayed. This is realized by searching the complete documentation list for the error code (in this case error code 615, see Figure 4.24).

This documented error list in HTML can contain further HTML links for detailed

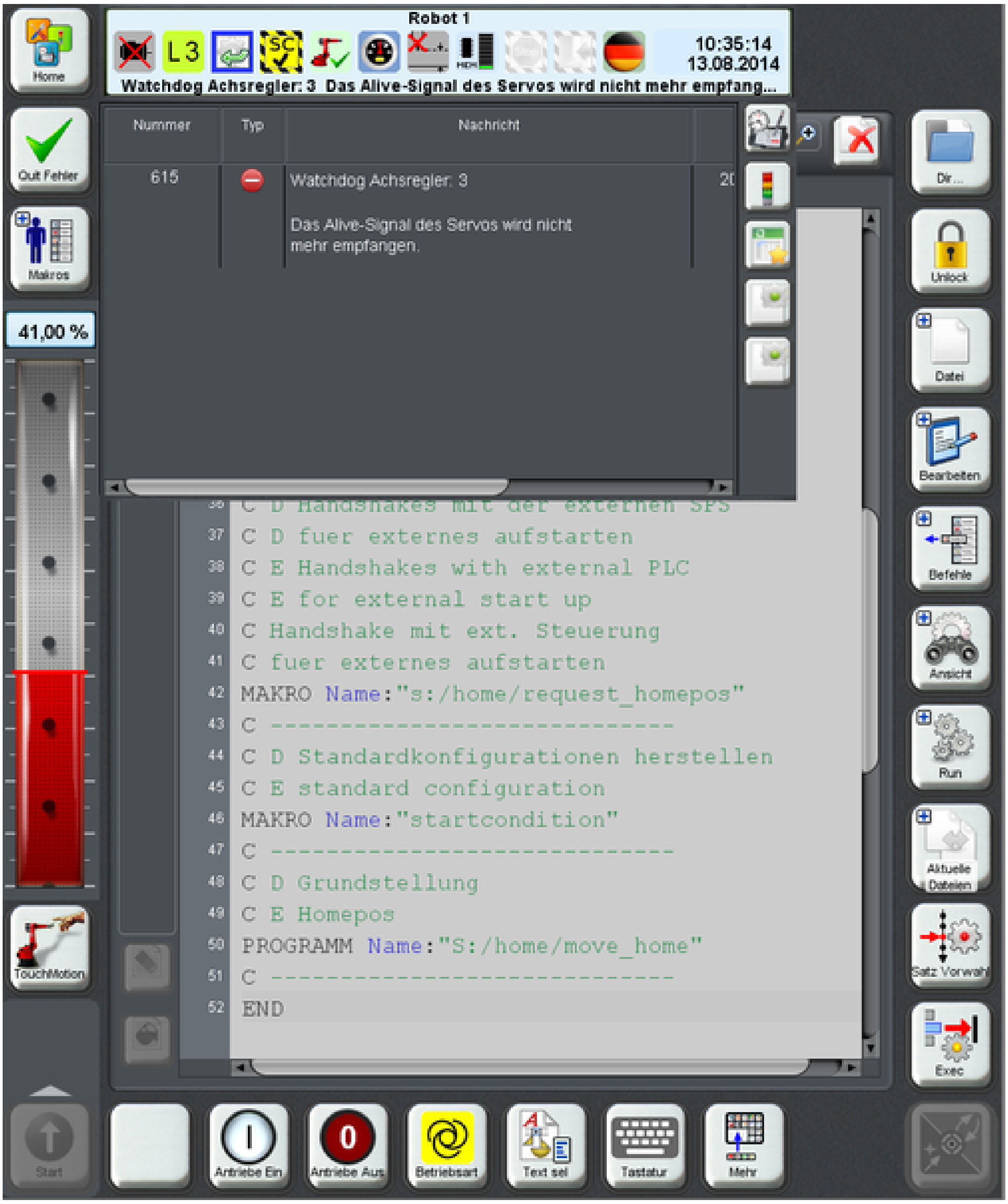


Figure 4.23: *Showing a failure in axis 3 controller on reisPAD*

descriptions on the failure cause or recommended measures. In the example the service technician can find a link to an instruction for the controller exchange. This document can be customized, so that documents on the compact flash storage or a connected network can be added.

In the example the routine "8W-Questions" of Procter&Gamble has been added, which is a list of questions on the current status of the plant which is used for docu-

ERROR CODE	DESCRIPTION	CAUSE	MEASURES
S615	Watchdog Axis Controller The Alive Signal of the Servo Controller cannot be received.	The communication between robot controller and axis controller is disturbed. It is probably a hardware malfunction.	Please check the hardware components: MModul, cable connections and servo controller. Try to change the controller: Instruction for controller exchange 8W-Questions

Figure 4.24: *Documentation for error code 615*

Figure 4.25: *Discussion on the functionalities of scenario 3*

mentation directly after a breakdown. In the discussions with the users (depicted in Figure 4.25), they agreed that this offers sufficient support for the failure diagnosis.

4.5.3 Instruction prototypes with Augmented Reality support

The instruction on the controller exchange mentioned in the previous section is a static document, which is used in a printed form for training. In the project, we chose this instruction for further improvement. We wanted to compare the static document with dynamic visualization improved by Augmented Reality methods, which we called "Eyestruction". We wanted to test it on a tablet PC and on Augmented Reality glasses.

4.5.3.1 Instructions

The instructions for the controller exchange have been provided by KUKA Industries and describe the different working steps. In the context analysis, the procedure has been documented by video and text. There are some necessary preparations like driving the robot to a reference position or switching off the voltage, which are omitted in the used instructions, because we used a switch cabinet without any voltage or connected manipulator.

4.5.3.2 Used Augmented Reality techniques

From the related work we deduce that an application for the training of employees requires several things:

- The user must not be guided through the task in a too high detail. The application should only provide clues for the correct processing of each working step. On the user request, further explanations can be provided.
- The user should not use a photo-realistic simulation of the active work piece on the real image. A colorful indication on the spot which requires the attention suffices.
- For enhancing the generation of mental models during the maintenance process, the user needs a progress bar. This enables the user to put the actual working step in the overall context.
- The application should omit unnecessary visual feedback, so that the user has a chance on spotting unexpected failures and solve them.

The mask for the application respects those deductions and is depicted in Figure 4.26.

The user can see the actual working step and the number of all working steps in the headline. Additionally, a progress bar is provided (green), which also gives an overview of the tasks. The instruction of the working step is deliberately kept brief, so that the user can deduce a large part of the task out of context. The indications on the working piece (pink) help him or her on that. If additional information is needed, the user can use the optional explanation, which is located on the bottom part of the screen. The user is able to navigate freely through the application with the buttons (left: "back", right: "next") and can process the tasks on his own speed.

4.5.3.3 Used devices

We wanted to design a fundamental concept for the realization of maintenance applications. In order to test different AR metaphors, two off-the-shelf devices were chosen, which have completely different interaction- and visualization capabilities.

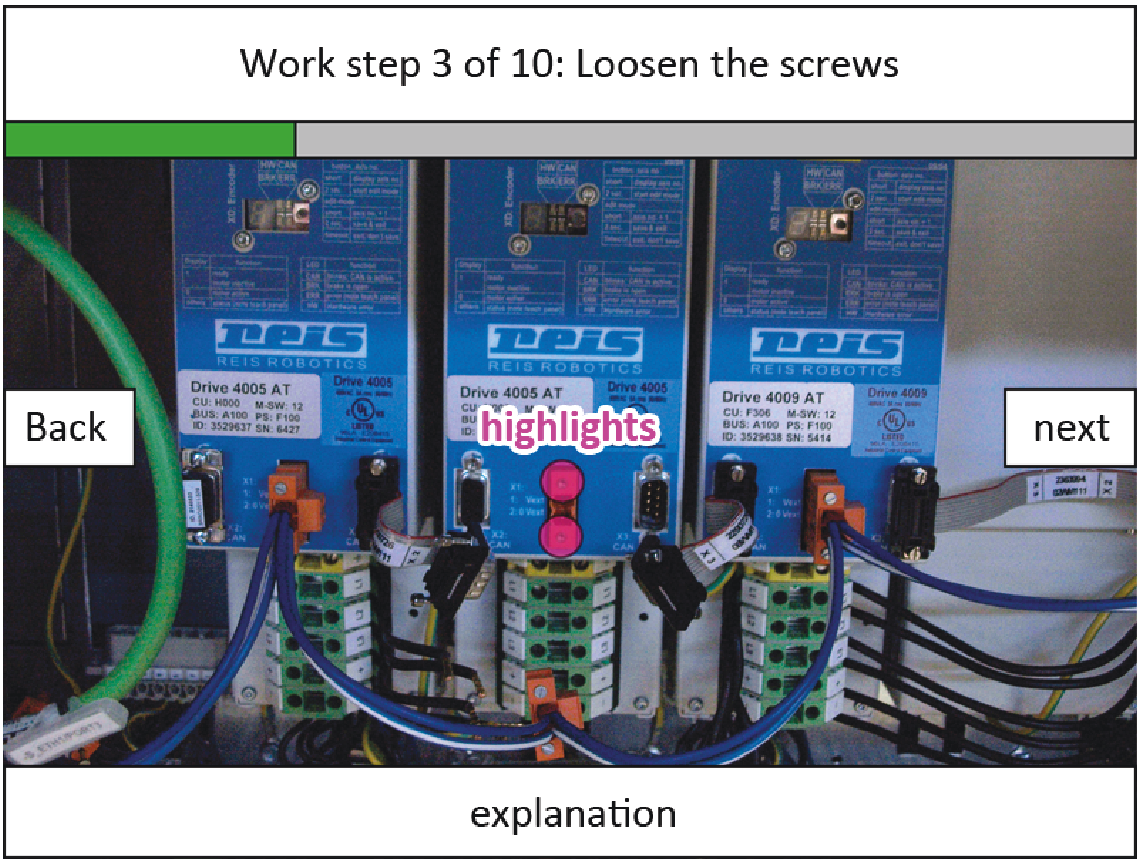


Figure 4.26: *Layout concept for the Eyestruction application*

Samsung Galaxy Tab 2

In order to test a conventional interaction, an tablet PC was chosen. As these devices and also smartphones have been well-established in the consumer market for several years, a lot of persons are accustomed to this interaction concept.

The used Samsung Galaxy Tab 2 (see Figure 4.27) was provided by the chair for psychologic ergonomics of the University of Würzburg. Technical data is provided by Figure 4.28.

Epson Moverio BT-200

The Optical See-Through Head-Mounted Displays (HMD) follow a completely different interaction principle. They offer projection planes in the glass, so that the normal perspective can be overlain by a digital image. In the opposite to VR HMDs, the user can sense the environment directly, because the glass is still mainly transparent. This is mandatory for this maintenance application: The safety regulations require, the worker to have direct eye contact to his or her working environment.

The Zentrum für Telematik e.V. has an Optical-See-Through HMD, the Epson Moverio BT-200, which is used in this study. It is also an off-the-shelf device, produced for the consumer market. The display can be controlled by a touchpad, which is not as intuitive as the touch interaction on the tablet. The user can use

Figure 4.27: *The used Samsung Galaxy Tab 2*

Operating system	Android 4.0.3
Screen size	10.1 inch
Resolution	1.280 x 800 Pixels
Camera	3.2 MP, no Autofocus, no flash

Figure 4.28: *Device specific data of Samsung Galaxy Tab 2*

both hands while he or she is wearing the glasses, which is an advantage, because the application can support the user the whole time of the process, not only when he or she directly consults the instruction (by taking the paper instruction or the table PC in his or her hands).

The advantage of having free hands while using the device is shortened when the user has to use the touchpad. So we thought of a camera-based interaction method. The glasses have projection planes on both eyeglass lenses, so that a stereo rendering is possible for enabling 3D-content for both eyes. By using this function, the available resolution is cut in half concerning the width. Device-specific data about the Epson Moverio BT-200 can be found in Figure 4.30.

4.5.3.4 Used components and backends

Android and Metaio SDK

Both devices work with Android OS Version 4.0.3., so that the developed application can be used on both devices with appropriate abstraction. For developing the

Figure 4.29: *The Optical See-Through Head-Mounted Display Epson Moverio BT-200*

Operating system	Android 4.0.3
Screen Size	0,42 inch
Resolution	960 x 540 Pixels without stereo-rendering 480 x 540 Pixels with stereo-rendering
Camera	640 x 480 Pixels, no autofocus, no flash

Figure 4.30: *Device specific data for Epson Moverio BT-200*

Android SDK from Google in the programming environment Eclipse was used. Applications for Android devices can be developed and tested with the programming language Java.

The testing of AR principles for maintenance applications was in the focus of this work, so that no AR system was implemented anew. We used the Metaio SDK ([153]). For research purposes it was allowed to be used freely until December 2015. It was an extensive framework for AR applications, which offers libraries for every desktop and mobile platform. It offers implementations of different tracking methods and different visualizations for AR-content. The attention of this research lays in the concept of the AR interaction. In the following sections, the used components are described in more details, while a survey of those components is shown in Figure 4.31.

The Metaio SDK expands the Android SDK with AR specific functionalities. In the background, all data is managed by the persistence framework OrmLite with an

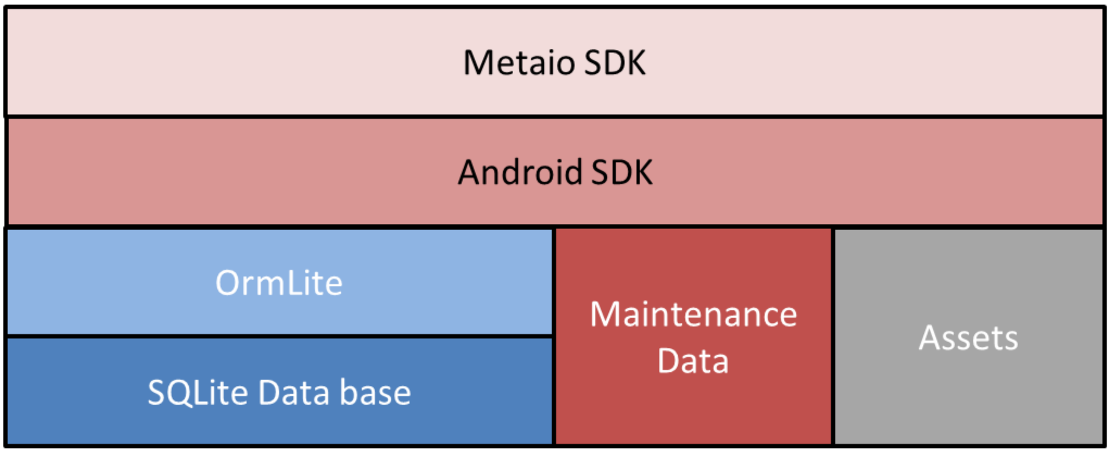


Figure 4.31: *Survey of the used components for the Eyestruction implementation*

SQLite database. The data is provided by a maintenance file. All other data used by the application is read from an "Assets" folder.

Maintenance data

For the visualization of content in the application, the maintenance data is necessary. It provides instructions, graphic indications, and audio data for the working step. The data is read from an XML-file, whose structure is specified by a DTD-file. An example for this maintenance data is provided in Figure 4.32.

Maintenance data specifies marker `<markierungen>`, working steps `<arbeitsschritte>` and maintenance operations `<wartungen>`. Every working step can have multiple markers and every maintenance operation has multiple working steps. This maintenance data is stored in the device and parsed by the class `WartungParser`.

Assets

All data used by the application is summarized as assets. This includes 3D-models (see Figure 4.33) and audio-data. 3D-models have been created with the 3D graphic software Blender [154] and are stored as OBJ-files. The rotating marker is used to indicate to the user, in which direction he or she has to tighten the screw. An algorithm must be seen to be believed. [155]. Audio data are stored in MP3-format and have been recorded by using the Software Audacity [156].

Class diagram for data structure

The simplified class diagram is depicted in Figure 4.34. All data structures in the database have the abstract class `Entity`, which defines a unique ID used as index in the database. The methods `equals` and `hashCode` guarantee that two instances of one class who have the same ID are the same object.

The class `Wartung` describes the entire maintenance operation and has a specific name. The class `WartungHatArbeitsschritt` specifies which two objects of `Wartung`

```
<daten>
    <markierungen>
        <statischeMarkierung name="Markierung1" ... />
        <statischeMarkierung name="Markierung2" ... />
        < ... >
        <rotierendeMarkierung name="Markierung3" ... />
        < ... >
    </markierungen>

    <arbeitsschritte>
        <arbeitsschritt name="Arbeitsschritt1" ... >
            <hatStatischeMarkierung>Markierung1</hatStatischeMarkierung>
            <hatRotierendeMarkierung>Markierung3</hatRotierendeMarkierung>
            < ... >
        </arbeitsschritt>
        <arbeitsschritt name="Arbeitsschritt2" ... >
            <hatStatischeMarkierung>Markierung1</hatStatischeMarkierung>
            <hatStatischeMarkierung>Markierung2</hatStatischeMarkierung>
            < ... >
        </arbeitsschritt>
        < ... >
    </arbeitsschritte>
    <wartungen>
        <wartung name="Wartung1" >
            <hatArbeitsschritt>Arbeitsschritt1</hatArbeitsschritt>
            <hatArbeitsschritt>Arbeitsschritt2</hatArbeitsschritt>
            < ... >
        </wartung>
        <wartung name="Wartung2" >
            <hatArbeitsschritt>Arbeitsschritt2</hatArbeitsschritt>
            < ... >
        </wartung>
        < ... >
    </wartungen>
</daten>
```

Figure 4.32: *Example for maintenance data*

and `Arbeitsschritt` belong together. The class `Arbeitsschritt` symbolizes a working step, which has a specific name and contains text instructions, explanations and links to external data of the audio files. The `ArbeitsschrittHatMarkierung` is also used for mapping between `Arbeitsschritt` and `Markierung` for assigning the correct marker type (static or rotating).

Initialization (MainActivity)

The class `MainActivity` initializes the application. A survey of the start process is depicted in Figure 4.35. The class initializes the database, loads the assets with the help of the Metaio SDK and starts the GUI (see Figure 4.36). The user can choose with two buttons, if he or she wants to use the text or audio version of the application. Further activities depend on the device type, which is automatically loaded from Android.

Figure 4.33: *Used 3D-models: rotating marker, static marker and arrow*

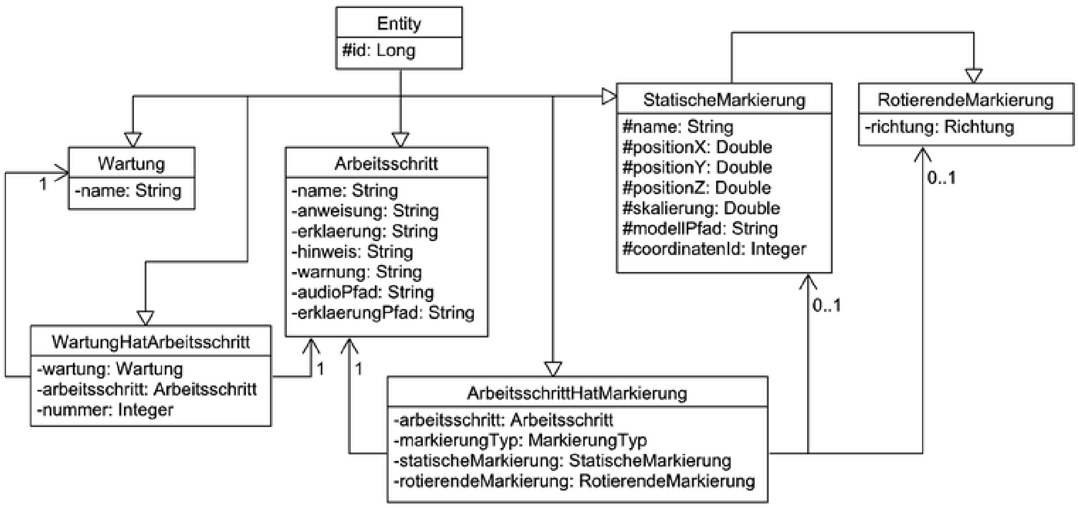


Figure 4.34: *Simplified class diagram of used classes*

4.5.3.5 Tracking configuration

To achieve precise tracking with different devices without a high installation effort, the marker-based tracking method with 2D-markers was chosen. Metaio SDK offers 512 defined markers which are applicable for the camera-detection, if the appropriate XML-file for the marker configuration has been generated.

Controller cover

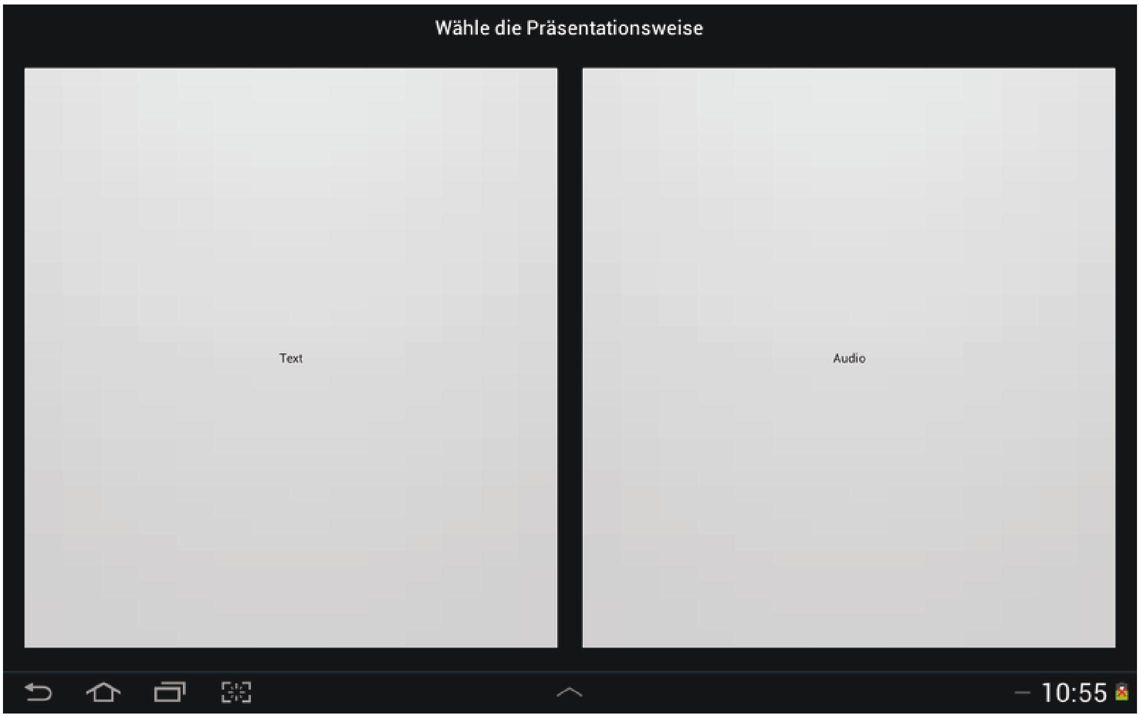


Figure 4.35: *Layout of MainActivity GUI*

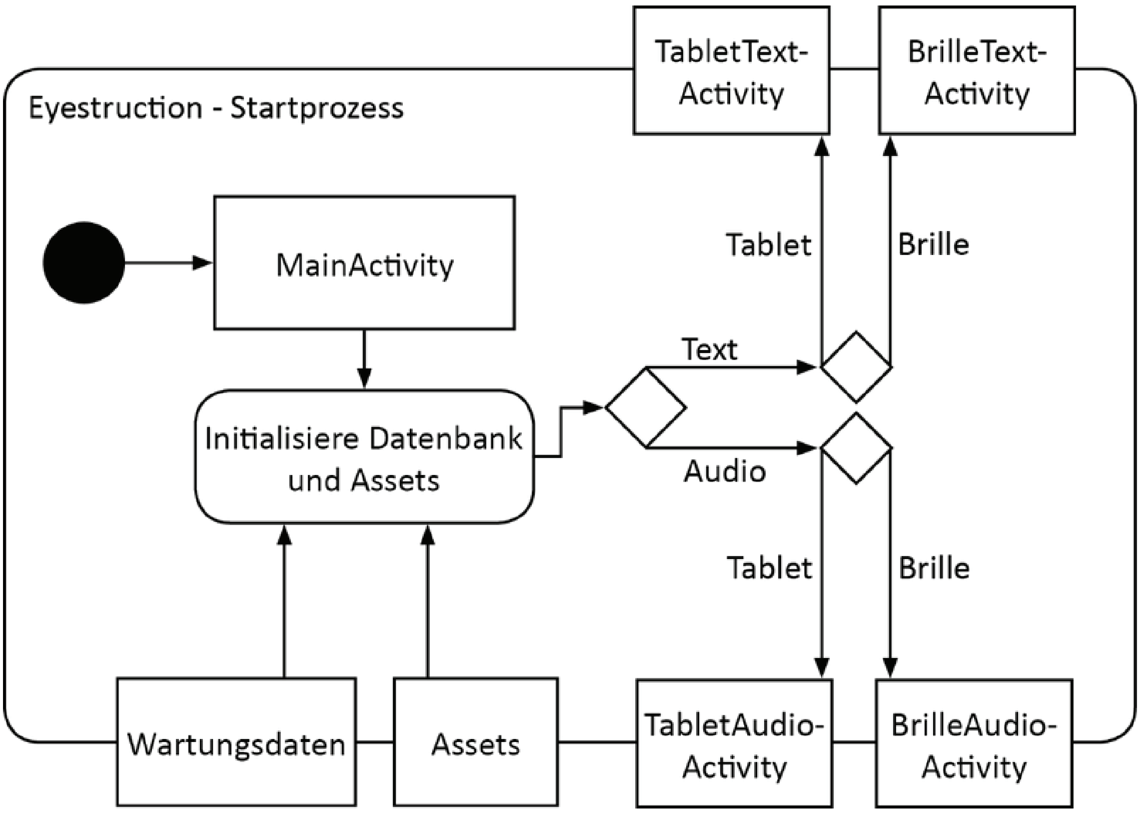


Figure 4.36: *Activity diagram for initialization process of Eyestruction*

To enable the installation of the developed Augmented Reality System in an arbitrary but similar switch cabinet, a cover for the servo controller have been developed (see Figure 4.37). The controller box was measured and a paper cover was designed, so that all markers keep a constant distance to the origin of the tracking coordinate system. So the tracking configuration can be reused without changes.

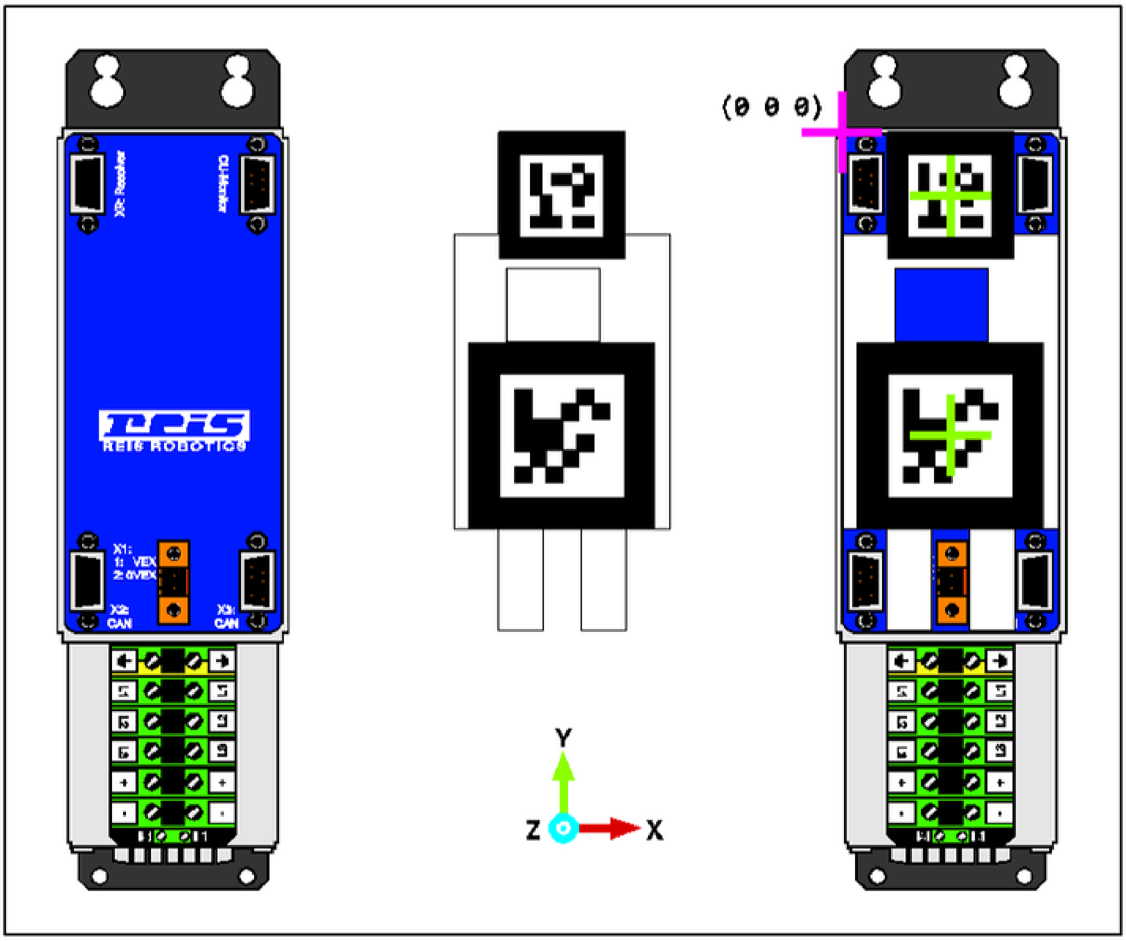


Figure 4.37: *Left: Controller box, Middle: Controller cover, Right: Controller with adjusted cover*

In Figure 4.37 the controller, the controller cover and the adjusted cover on the controller are depicted. Each marker defines its own coordinate system (green), which has a defined distance to the coordination system of the tracking configuration (origin marked pink). The system uses those distances (measured in millimeter with respect to the marker's center). The Metaio SDK offers a method for integrating several markers in one coordinate system, but there have been a lot of errors during the development with this method, so it was not used further.

Another additional marker was used, which was placed beneath the controller in the switch cabinet. It was required to show the indications on the lower end of the controller. Its position has been measured and included in the tracking configuration.

The use of the markers in the real switch cabinet is depicted in Figure 4.38.

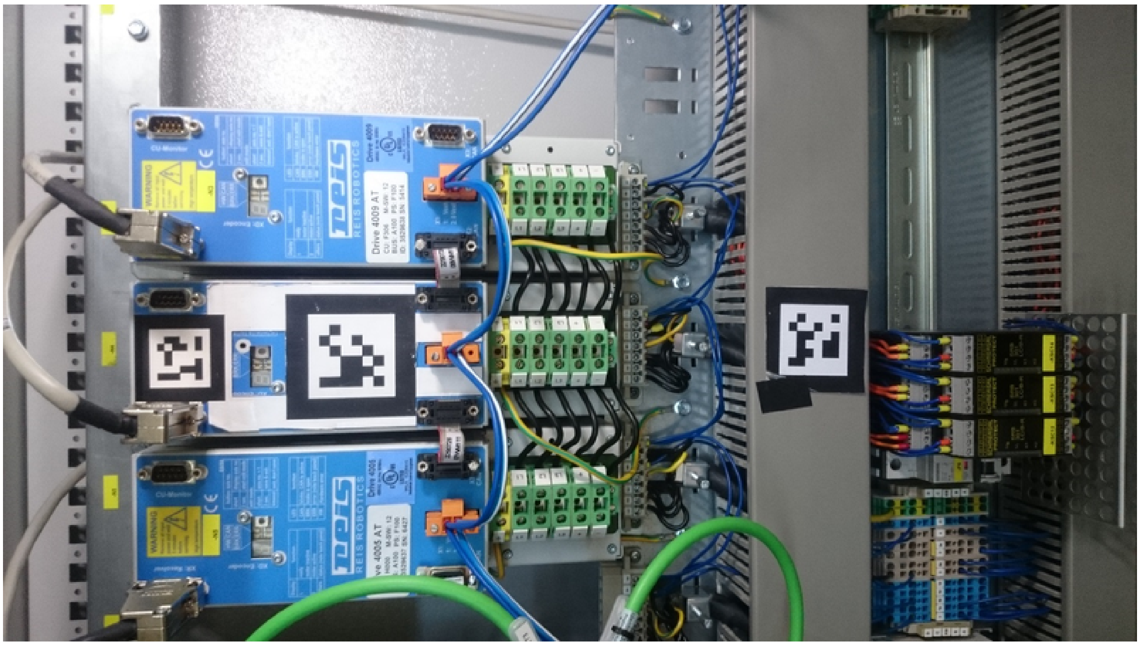


Figure 4.38: *Survey of the markers in the switch cabinet*

The tracking configuration used for the tablet PC included all the above mentioned markers. The SDK offers the possibility to combine the position data from the camera picture with other measurements like accelerometer or gyroscope of the device with a so-called Fuser . We used the SmoothingFuser which is preconfigured and needs no further adaption for a good result.

The tracking configuration of the glasses additionally contained several function markers which aren't used for tracking.

4.5.3.6 Device specific application parts

Basics

The abstract class `MarkierungARActivity` extends the class `ARViewActivity` provided by the Metaio SDK. With the help of the interface `IMetaioSDKAndroid` all functions like the including of 3D-data or the tracking configuration is provided. For our framework, the class `MarkierungARActivity` is used to communicate with the database. On every frame update it redraws all markers needed for the current working step. It is expanded by the class `MarkierungAudioARActivity`, which is able to play audio files by using the MediaPlayer of Android.

Tablet PC

The GUI for the tablet PC application is depicted in Figure 4.39, the class `TabletTextActivity` extends `MarkierungARActivity`. It realizes the layout mask defined at the beginning and uses touch interaction on the "Back", "Next" and "Explanation" field. It needs an additional GUI element for providing the full text of the explanation. It is shown in Figure 4.40 and realized with the library `AndroidSlidingUpPanel`.

Figure 4.39: *GUI TabletTextActivity*

For the audio version of the table PC GUI, the class `TabletAudioActivity` extends `MarkierungAudioARActivity`. It has static text buttons that play the instructions or the explanation.

HMD device

The correct visualization of the layout on both screens (right and left eyeglass lens) requires some precautionary measures. We use stereo rendering, which provides a different picture for each eye to enable a 3D-effect. The Metaio SDK requires the camera parameters of the glasses' camera and a hand-eye calibration, which estimates the offset of the eyes. Both configurations can be obtained by a Metaio "Toolbox" and stored on the device.

As mentioned above, we do not want another touch interaction with the touch device of the Epson Moverio. We decided to use so-called function markers, on which the user should look upon for triggering a specific function. The function markers are defined in the tracking configuration. The system can measure, if a function marker is detected by the camera for a preset period of time. A counter is incremented for every frame, in which the specific function marker has been detected. For every frame, the marker is not detected, the marker is decremented. After the activation of one marker the system waits a small time span until it accepts a different one to prevent the user from calling an unwanted function. A representation of this loading

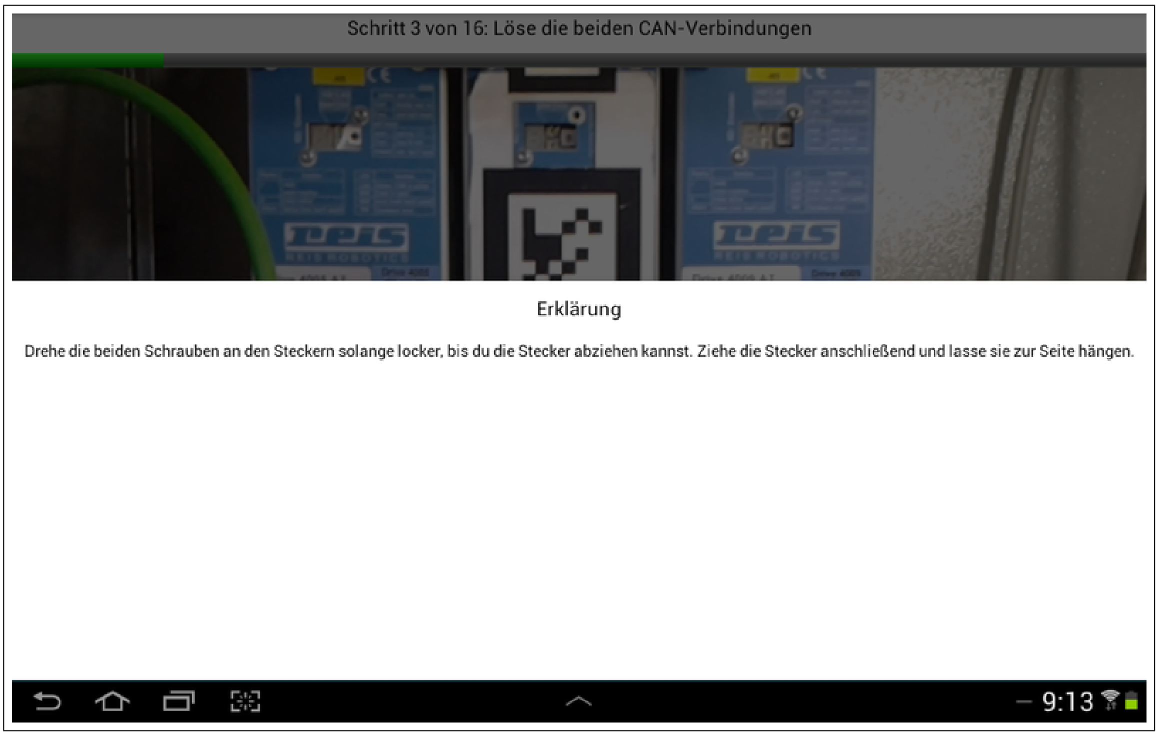


Figure 4.40: *GUI TabletTextActivity with explanation*

process is shown to the user on the lower part of the screen as a yellow bar. The function markers are located on the bottom of the switch cabinet (shown in Figure 4.41) and have paper flaps, which can be opened by the user as needed, to trigger the required function.

If the user wants to go to the next working step, he or she opens the flap of the equivalent function marker. The user directs his or her view (and also the HMD's camera) on it and waits until the bar on the bottom is full. Then the system jumps to the next step.

The stereo rendering divides the accessible screen width in half: The left half is presented on the left screen and the right half on the right screen. The GUI needs to be depicted twice with a slightly different view on the 3D-indications. The layout is depicted in Figure 4.42 and also realizes the layout mask discussed above.

For the audio version of the GUI, the text boxes are omitted. The audio file is played automatically on each new working step. But there are also function markers for "Repeating the instructions" and "Play the explanation" which can be used accordingly to the other function markers.

4.5.4 Evaluation

In the beginning, we have made the following hypotheses:

- In order to test the developed AR applications, a study in the University of Würzburg was performed. If an AR application helps a worker on his

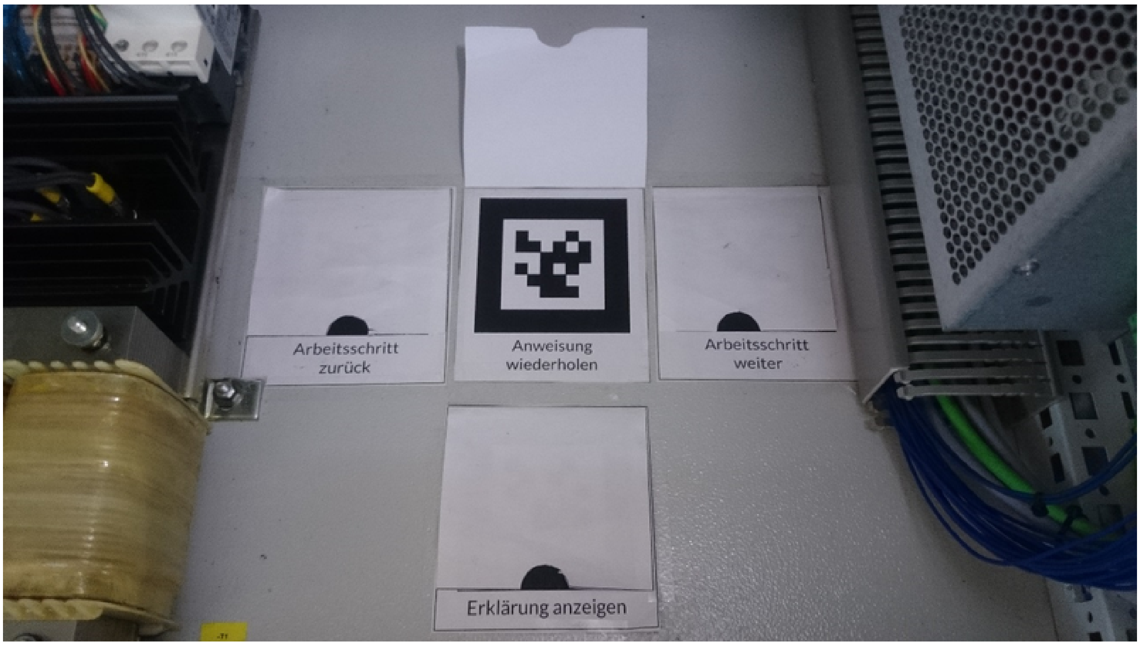


Figure 4.41: *Function marker for the interaction with the HMD application*

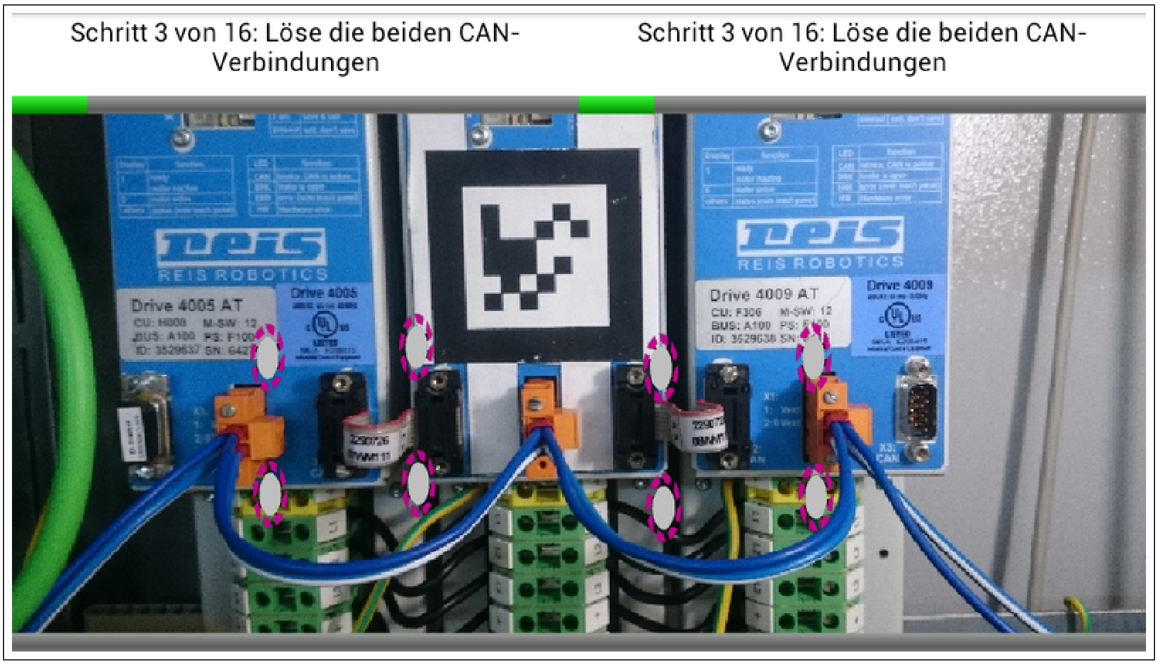


Figure 4.42: *HMD GUI with text*

Figure 4.43: *HMD GUI with audio*

maintenance and repair tasks, then we expect the users, who perform the same work with a paper instruction to require more time, make more mistakes, have a higher task load, perceive a lower ease of use and utter more negatively than users with AR applications. (AR > paper)

- The tablet text application offers a big screen of the tablet PC and accustomed information presentation with text instructions. Thus, we expect the tablet text application to be more effective, more efficient and provide a higher contentment than the tablet audio implementation (tablet-text > tablet-audio).
- The screen of the HMD is smaller, so we expect the reduced GUI of the HMD audio application to be more effective, more efficient and to provide a higher contentment than the HMD text application (HMD audio > HMD text)

4.5.4.1 Method

Participants

The study was performed with the help of the proband system of the institute for human-computer-media Würzburg. 50 students of the field of study human computer systems (18) and media communication (32) have been recruited, consisting of 27 females and 23 males with an average age of 21 years. No participant has ever performed a similar repair task before or had been accustomed to the experiment workflow.

Experimental design

The five different possible instructions are enumerated as follows:

- Paper instructions
- AR-application on the tablet PC with text instructions (tablet text)
- AR-application on the tablet PC with audio instructions (tablet audio)
- AR-HMD-application with text instructions (HMD text)
- AR-HMD-application with audio instructions (HMD audio)

The type of the instruction is an independent variable with five discrete characteristics. Several dependent variables have been measured in the study, which will be explained in the following section.

- **Success**
  At first, we measured if a participant has successfully ended the repair or if he or she needed to interrupt the experiment. Thus, the variable was dichotomous.

- **Errors**
  At the end of the maintenance, the amount of errors was estimated and used as a degree for the effectivity. The definition of an error had been provided by one of the service technicians of P&G. As a maintenance expert, he is accustomed to this switch cabinet and responsible for the service technician training at Marktheidenfeld. An error is counted if a cable is plugged not tight enough, unplugged or wrongly plugged. Also all screws are checked on tightness. Working steps, which have been left out, are also counted as errors. The variable therefore is scaled absolutely.

- **Time**
  The procedure time in minutes required by the participants served as a measure for the efficiency. The timing was initialized manually at the beginning of the repair and stopped on the participant's notification that he or she has finished. The instructions in all applications indicated that the participant should utter this notification. The measurement was noted down in seconds, but used mathematically rounded on minutes for the comparison. The variable is scaled absolutely.

- **TLX**
  For having another indication for efficiency, the NASA Task load Index has been estimated ([60], compare Section 2.3.2). The questionnaire measures the subjectively perceived load and other data interesting for efficiency. The scalement of the variable is absolute.

- **QUESI**
  The user's contentment has been measured with the QUESI ([72] compare Section 2.3.5). It measures the subjectively perceived intuitive usability of a system. The variable is scaled absolutely.

- **Comments**
  At the end of the experiment, the user is motivated to comment the application he used positively or negatively. Those comments have been translated with the values "-1" and "$+1$" so that a comment-value of for the user can be estimated. Thus, the variable is scaled absolutely.

- **Learning effects**
  Every participant performs the repair task with a weighted randomized allotted instruction. As no participant has ever performed a similar task or has had previous knowledge of the instructions or the experiment procedure, learning effects can be excluded.

Material
In order to minimize experiment guidance effects, all instructions for the participants have been handed out in written form. At the beginning of the experiment, they got a test person information sheet with a description of the procedure and possible risks. Every participant had to fill out an agreement for recording and processing the data. The participant then chose blindly a piece of paper out of a bag. On the paper he or she read the type of instruction and got the corresponding information sheet, then he or she processed the experiment:

- The paper instruction had been printed out and the participant could use it for the whole procedure. Also the other instructions could be used all the time.

- The applications Tablet-Text and Tablet-Audio were presented on a Samsung Galaxy Tab 2.

- The applications HMD-text and HMD-audio had been realized on the Epson Moverio BT-200 AR glasses.

- The participants needed to wear in-ear earphones for the audio versions. In order to achieve a comparable noise background for all experiments, the not-audio participants were asked to wear earplugs.

- The switch cabinet had been provided by KUKA Industries and is exactly in the same state as it were in the production environment, except that it carries no voltage. The participants performed the repair with real components.

- The questionnaires NASA TLX and QUESI have been printed out and the participants filled them out.

4.5.4.2 Results

The measured values were evaluated with a variance analysis with contrasts. The contrasts were built upon the hypotheses and are described in Figure 4.44. The first assumption was that the Augmented Reality applications perform better than the paper instructions (AR > paper). The second one stated that on the tablet PC the text instructions are better than the audio instructions (Tablet-Text > Tablet-Audio). The last one supposed the opposite effect for the HMD (HMD-Audio > HMD-Text). As all those contrasts are orthogonal, no further adjustment of the alpha-niveau was needed.

Hypothese	*Paper*	*Tablet-Text*	*Tablet-Audio*	*HMD-Text*	*HMD-Audio*
AR > paper	-4	1	1	1	1
Tablet-Text > Tablet-Audio	0	1	-1	0	0
HMD-Audio > HMD-Text	0	0	0	1	-1

Figure 4.44: *Contrasts used for evaluation*

Effectivity

All participants were able to complete the repair, nobody canceled the experiment. For this reason, we did not evaluate the dependent variable success. A survey of the average number of errors accounted for each application is visualized in Figure 4.45. The error bars on the values visualize the standard deviation of the measured values (as it will in the following figures).

The analysis of the contrasts did not show any significant difference:

- AR > Paper: $F(4,45) = 0.10, p = .05$
- Tablet-Text > Tablet-Audio: $F(4,45) = 2.41, p = .05$
- HMD-Audio > HMD-Text: $F(4,45) = 3.29, p = .05$.

Efficiency

As the manual time measurement in seconds is imprecise, the measured times were mathematically rounded on minutes. The measurements with the corresponding standard deviations are depicted in Figure 4.46.

The analysis of the contrasts did not show any significant difference:

- AR > Paper: $F(4,45) = 0.97, p = .05$
- Tablet-Text > Tablet-Audio: $F(4,45) = 0.00, p = .05$

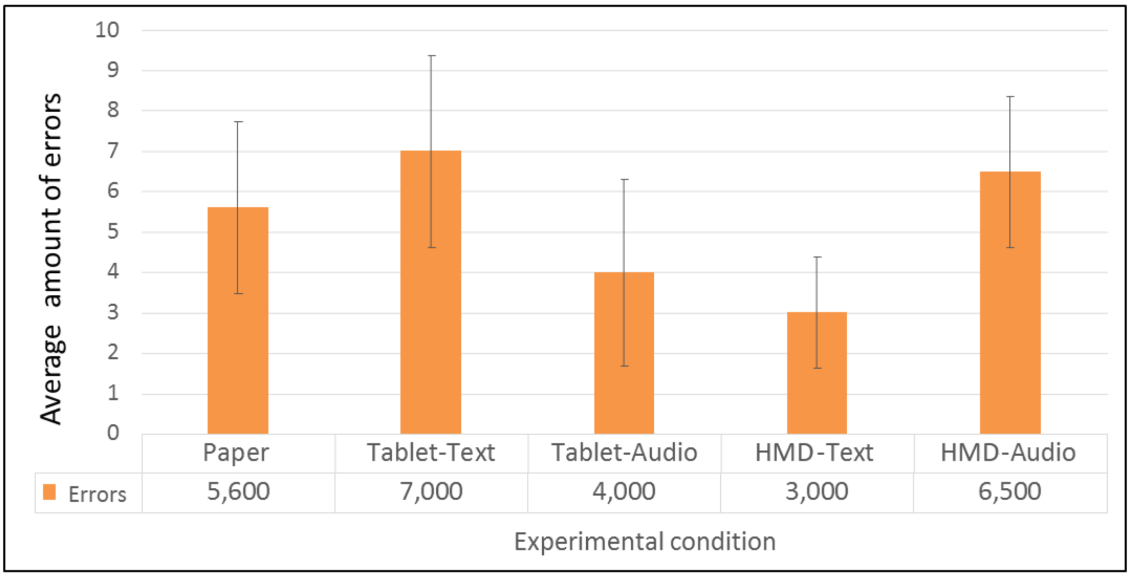


Figure 4.45: *Average amount of errors for each experimental condition*

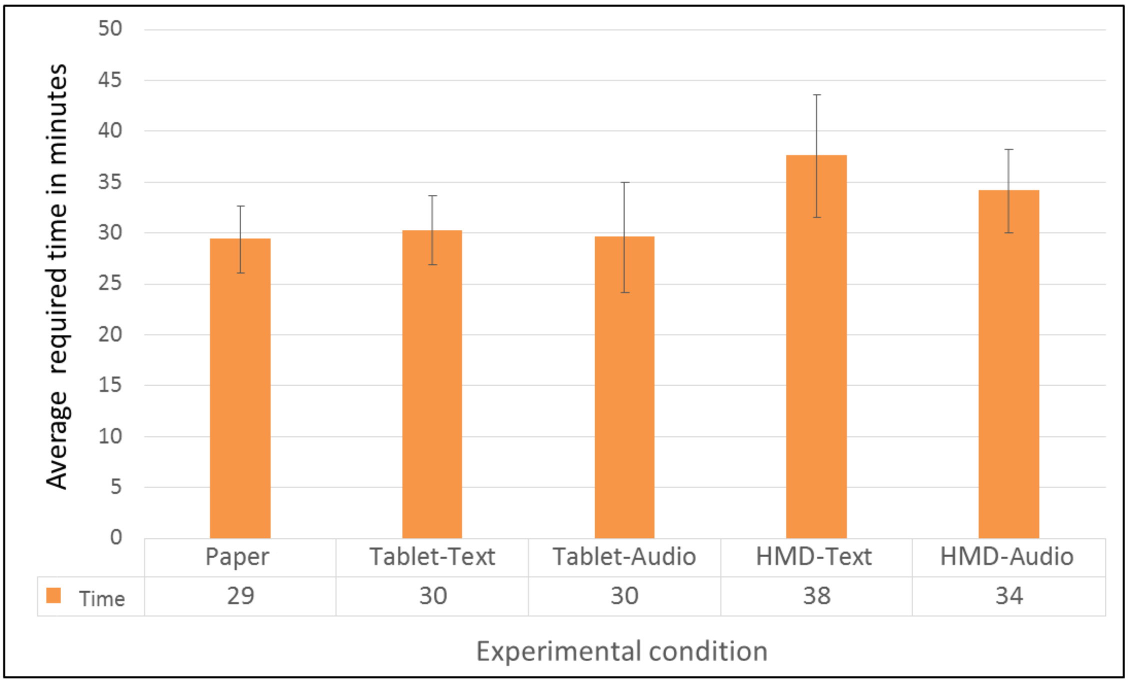


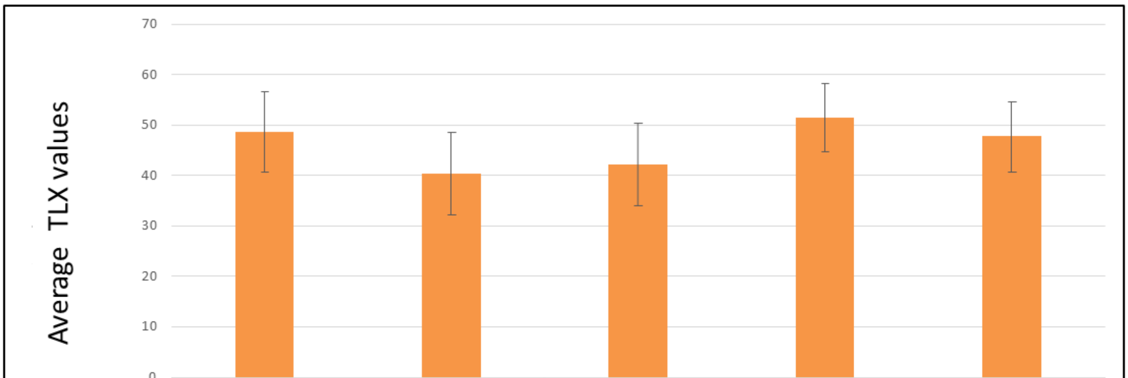
Figure 4.46: *Average required time in minutes for each experimental condition*

- HMD-Audio > HMD-Text: $F(4,45) = 0.62, p = .05$.

In Figure 4.47 the average values of the NASA TLX for the different experimental conditions is depicted. An analysis of the contrasts did not show any significant difference.

- AR > Paper: $F(4,45) = 0.31, p = .05$
- Tablet-Text > Tablet-Audio: $F(4,45) = 0.06, p = .05$
- HMD-Audio > HMD-Text: $F(4,45) = 0.28, p = .05$

Also the evaluation of the subcategories of the NASA TLX showed no significant difference.



	Paper	Tablet-Text	Tablet-Audio	HMD-Text	HMD-Audio
TLX value	48,633	40,433	42,242	51,522	47,708

Figure 4.47: *Average NASA TLX value for each experimental condition*

Contentment

The average QUESI-values for the experimental conditions are depicted in Figure 4.48. No significant differences between the contrasts have been found:

- AR > Paper: $F(4,45) = 0.62, p = .05$
- Tablet-Text > Tablet-Audio: $F(4,45) = 0.87, p = .05$
- HMD-Audio > HMD-Text: $F(4,45) = 0.42, p = .05$

Also the analysis of the subcategories of the questionnaire showed no significant differences.

For the analysis of the comments, each positive comment was rated with the value "+1" and each negative with the value "-1". For each participant a comment-value was calculated. The average comment-value is depicted in Figure 4.49 for each experimental condition. The analysis of the contrasts did not show any significant difference:

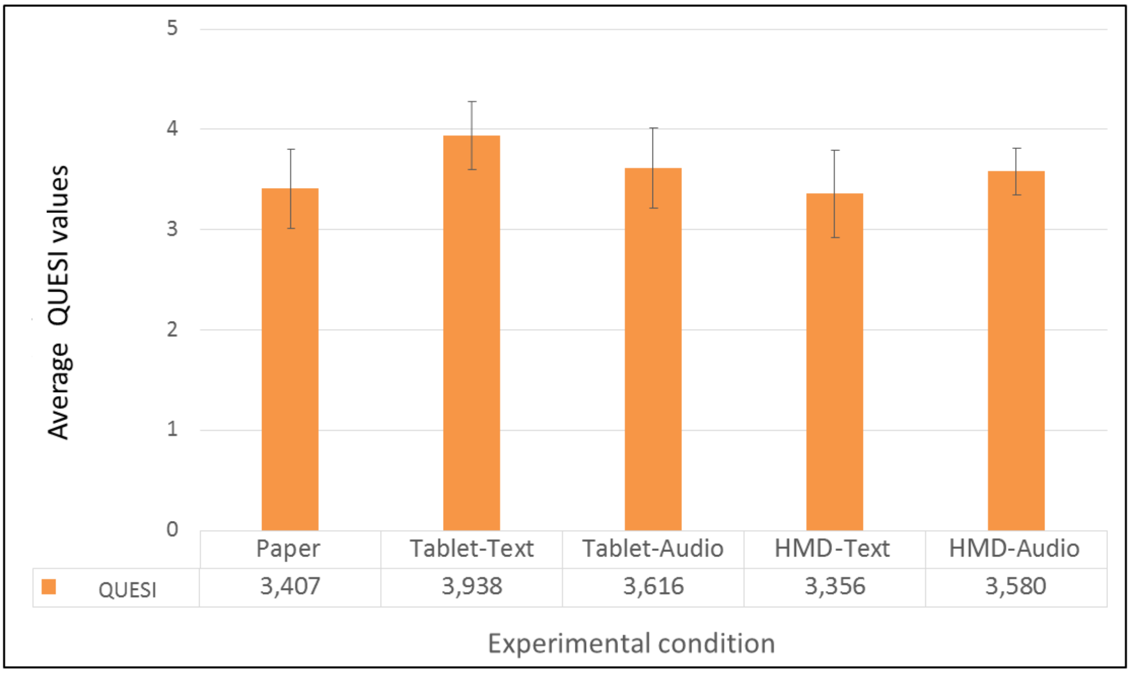


Figure 4.48: *Average QUESI-values for each experiment condition*

- AR > Paper: $F(4, 45) = 0.10, p = .05$
- Tablet-Text > Tablet-Audio, $F(4, 45) = 0.73, p = .05$
- HMD-Audio > HMD-Text, $F(4, 45) = 0.18, p = .05$

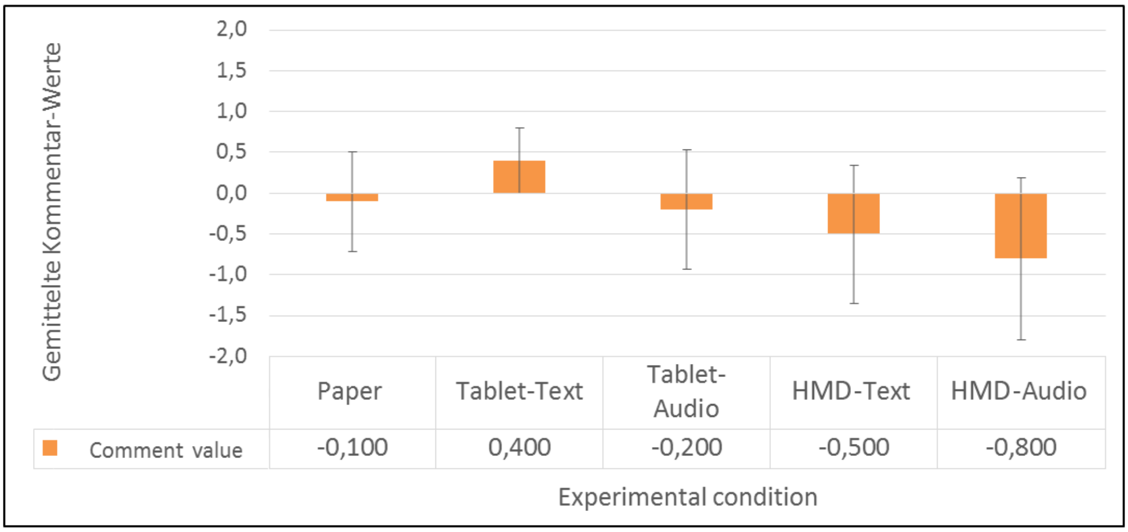


Figure 4.49: *Average comment-value for each experiment condition*

4.5.5 Discussion

4.5.5.1 Summary of the results

All measured values mentioned above are summarized in Figure 4.50. They have been evaluated with a variance analysis with contrasts. The "p-value" (significance level) for each test was 0.05. Although there have been large effect sizes, no condition shows a significant result.

Hypothesis	Effectivity (Number of errors)	Efficiency (Required time)	Efficiency (NASA TLX)	Contentment (QUESI)	Contentment (Comments)
AR > Paper	F(4, 45) = 0.10, p < .05	F(4, 45) = 0.97, p < .05	F(4, 45) = 0.31, p < .05	F(4, 45) = 0.62, p < .05	F(4, 45) = 0.10, p < .05
Tablet-Text > Tablet-Audio	F(4, 45) = 2.41, p < .05	F(4, 45) = 0.00, p < .05	F(4, 45) = 0.06, p < .05	F(4, 45) = 0.87, p < .05	F(4, 45) = 0.73, p < .05
HMD-Audio > HMD-Text	F(4, 45) = 3.29, p < .05	F(4, 45) = 0.62, p < .05	F(4, 45) = 0.28, p < .05	F(4, 45) = 0.42, p < .05	F(4, 45) = 0.18, p < .05

Figure 4.50: *Table with all results from AR repair study*

4.5.5.2 Relevance of the study

The evaluation of the measurements could not support the assumption that the user performs better using AR applications compared to the paper instructions. There are several studies found in the literature, which indicated this, but this study was not able to support the thesis for this case. But several aspects diminishing the meaningfulness of this study have to be addressed.

The used participants were not representative for the later users. The students of HCI and media-communications are not familiar from their university background with electrotechnics or physical mechanical tasks. That is why several participants noted that they had no craftsmanship capabilities and no experience in using the provided tools. In the pilot study, this problem had not occurred, so that no data on this pre-experience has been collected. Without sufficient capability or experience, participants may have felt overwhelmed, so that they could not perform the repair adequately. But without data, this thesis is a mere assumption and cannot be proven. Further, without previous knowledge on the use of the tools, the participant may have needed more time, because he needed to get accustomed to the tools first. Those participants could not profit from the use of the AR applications, because their biggest problem was using the tools and not the correct procedure of the repair.

Another point is the missing domain knowledge of the participants. It is not possible to generate a domain neutral instruction to that specific case of the change of a robot controller. Other studies, like [118], also used students for the comparison of different AR applications, but they had a neutral task. Without the necessary knowledge the participants might have perceived additional cognitive barriers, which influenced the results of this study.

A possibility for omitting the mentioned effects, is to use probands with domain knowledge, like apprentices of the facilities working with industrial robots. This requires a high logistic effort and high costs and was not possible in this context.

4.5.5.3 Effect of the used device

A qualitative evaluation of the participants' comments showed, that there have been several complaints about the quality of the visualization and the stability of the markers. The reason for this might be the resolution of the cameras. Both devices had a relatively low resolution of the camera compared to the best case. This might have led to tracking errors. The use of more advanced versions of the devices could omit such technical effects.

By using the AR HMD, the stereo rendering decreases the screen resolution dramatically. Combined with the small screen size, which covers only a low percentage of the visual field, the layout components overlay a large part of the screen. The focus point of the HMD lies in a large distance to the eye, so that some probands perceived problems with the focusing of the context (especially text), because the distance between screen and switch cabinet was lower than the virtual distance to the text. Because of those technical problems, we suspect that the study measures the technical limitations more than the differences of the interaction principles. A possibility would be to use HMDs with a higher resolution or a completely artificial setting in VR.

4.6 Summary

This chapter has described several innovations in the field of mobile devices for the use in production, which are summarized in Figure 4.51. The team followed an all-in-one strategy in order to enable the required functionalities for maintenance, inspection and repair tasks with mobile device while profiting from its special capabilities.

4.6.1 Maintenance

The research started at Section 4.2 with the change of the static hardware teach pendant of the industrial manipulator to a tablet PC like device, which has been performed in the new "ROBOTstarVI" of KUKA Industries. It is discussed, how this approach enables new possibilities for the maintenance plan. The use for cyclic maintenance actions has been exemplary implemented and showed the huge potential achievable with low effort. This application also showed the use of contextual inquiry especially for an industrial work domain.

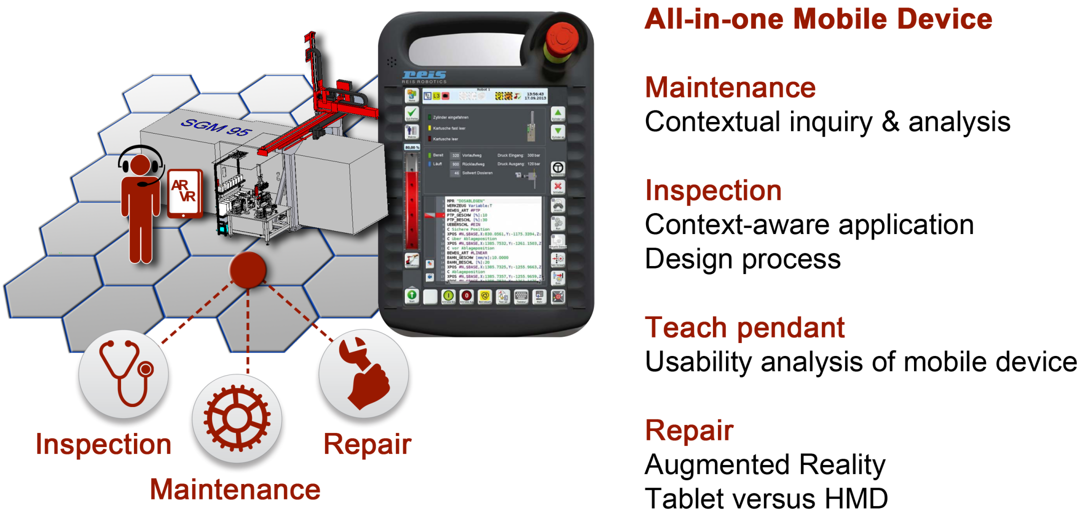


Figure 4.51: *Summary of the innovations: Mobile device for local maintenance*

4.6.2 Visual Control

Another option for using mobile devices in the maintenance context is the improvement of the so-called "VisualControl" process. This term is used to describe location-dependent inspection tasks. In Section 4.3, the application of the classic HCI developments procedures are shown. Several prototypes for the "VisualControl" application have been implemented, iteratively reworked and at the end evaluated in the real production.

The questioned workers definitely see the advantages of the mobile device. They expect the reduction of the documentation effort for maintenance tasks and plant breakdowns, as well as the quality improvement of the inspection. Especially telemaintenance and expert supported plant optimization indicate a great potential for cost reduction. The prototypes are browser based and thus platform independent and can also be used with the private communication device (BYOD).

4.6.3 User study PHG/reisPAD

In order to measure the advances of the reisPAD in comparison to the old teach pendant, a user study has been carried out with student participants. It has been described in Section 4.4. The study used heuristic evaluation, the keystroke-level modeling and usability testing. All three methods could provide valuable insights of possible improvements of both the old and the new teach pendant.

In the usability test with student users the tablet PC device reisPAD clearly exceeded the older hardware teach pendant. The participants were able to solve more exercises, with a lower subjectively felt task load, and the usability of the reisPAD was clearly better. Most probands preferred the reisPAD if they had the choice between both variants. There were significant differences in all three usability dimensions effectivity, efficiency and contentment but not for all estimated variables.

We can conclude, that problems of both interfaces were identified with the heuristic evaluation and the KLM and could be quantified in the usability test. The tablet PC implementation clearly proves a step forward for more usability in the industrial domain.

4.6.4 Repair study Eyestruction

Especially the use of Augmented Reality techniques will play an important role in the future for production and documentation. Section 4.5 describes an Augmented Reality case study uses an on-site AR-approach where a technician is guided during a repair task by a) paper instructions, by a tablet-based AR application with b) text or c) audio, and by an HMD-based AR application with d) text or e) audio. The tablet-based applications have been realized on a Samsung Galaxy Tab 2. The HMD applications have been implemented on the Epson Moverio BT-200 AR Glasses. Tracking was performed via optical markers. The switch cabinet used for the repair operation contains real components of an industrial manipulator. The study was carried out without any voltage on the switch cabinet. The repair task consists of a controller exchange which takes approximately half an hour. 50 student participants took part in this study.

Overall the use of the tablet-based version with text instructions was rated best in the qualitative results, although we could not find any significant confirmation. Notably, the participants had very different mechanics foreknowledge and cannot be compared to specialists in that area.

The study could not support the hypothesis, that AR provides better results. It nevertheless gave several important clues for future developments. The See-Through HMD used in this study provided a very low resolution and narrow field of view. Hence, chances are that this preliminary study compared the devices, not the interaction method. We assume those technical differences to adjust in the following years. For practical uses we recommend to perform own studies with the target hardware. If doubt, the tablet-based version will be a valid choice for an immediate productive use in industry.

4.6.5 Economical benefits

In the project "MainTelRob", an economical analysis of the three scenarios has been performed on base of the project results in the plant of P&G. By using a tablet PC instead of a stationary BDE terminal, the machine operator can reach each inspection point comfortably. He or she is guided from point to point, which provides a high security for the work process. The photos taken with the application and the description of detected errors help with the removal of anomalies. The instructions on each check point decrease the training period of new machine operators. On the investing side, a tablet PC has to be purchased. Currently a deeper investigation of the possible substitution of the BDE terminals have been started due to these results.

For repair tasks, the tablet instructions will prove useful, because there is a common procedure for all service technicians. The repair will be conducted more securely and also the hand-over process between shifts will be simplified. All setup work will follow the same conditions with the help of external guidance. The individual training and experience of the service technicians can be compensated by the electronic device and therefore play a minor part. Also the specialization of service technicians for specific machine groups can be obsolete. They estimate decreasing downtimes of the plants of 15 percent on the average and hope to reach a shorter training period for service technicians by 30 percent.

4.6.6 Conclusion

This feedback showed, that the methods from the usability / user experience domain can offer a great knowledge acquisition in the industrial automation. Especially important was the extensive context analysis and the iterative development with inclusion of the end users.

The described prototypes demonstrate new possibilities in the use of production data for maintenance in the context of a robot-based molding process. They are cause for thought for new implementations for human machine interaction in the era of the fourth industrial revolution.

Chapter 5

Asynchronous remote teleanalysis

This chapter introduces the own work performed for remote telemaintenance with asynchronous access to the plant and its data. The condition monitoring records the manipulator data and transfers it via the Internet to an application. We describe three possible applications for this data, which can run remotely in order to analyze and to optimize the plant and its processes: a complex sensor view with video integration of one plant cycle (application "Machinery Optimizer"), the possibility to analyze several similar plants and their data (application "Complex Analysis Tool") and the visualization of the production process with a dynamic 3D model (application "ProcVis"). Research focus lies on the user side: The goal of this work is to provide the external expert with a better understanding of the plant and its processes. This is realized for the first time in this work environment.

5.1 Survey

In the context of the Extended Human Supervisory Control scenario described in Section 2.6.3, the scope of this chapter is visualized in Figure 5.1. The remote expert gets data from the industrial robot and the plant, but is not able to control it directly. The data is used to create a software representation of the robot processes on the remote computer providing analysis and optimization functions. The remote expert can propose changes of the robot program to the service technicians, which are based on his asynchronous teleanalysis.

The fifth scenario of the project "MainTelRob" with the headline "Optimization with external help" deeply interacts with the fourth scenario "Condition Monitoring". This is why those two fields can be viewed together in some kind of asynchronous optimization process. The interface of the remote expert shall answer the following questions:

- What is the best way to provide the information for problem analysis in a intuitive remote interface?
- How can software assist the understanding of the setting and the plant process (situation awareness)?
- How can the information of several plants be merged?

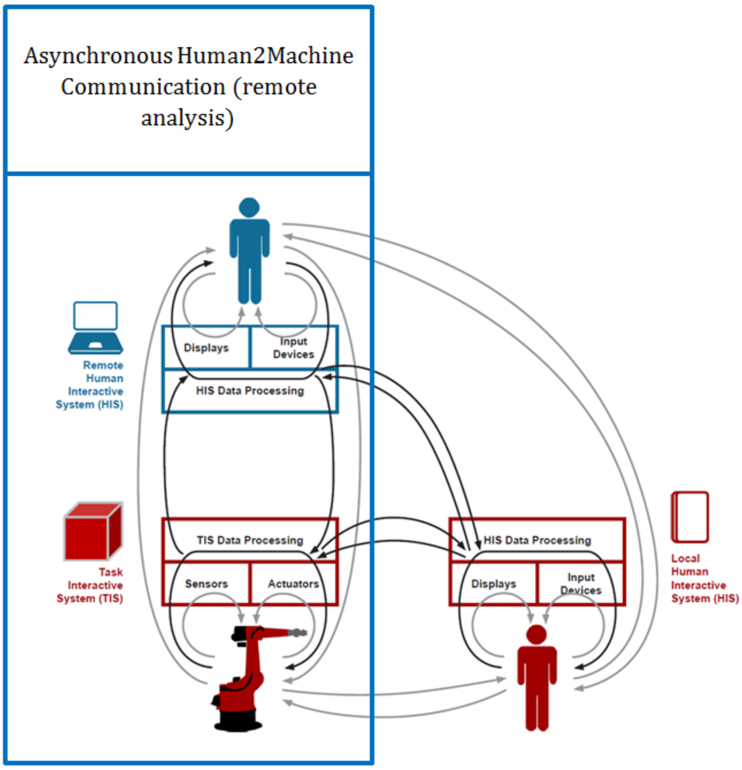


Figure 5.1: *Context of this chapter visualized in the extended Human Supervisory Control model*

- Can expert knowledge lead to predictive maintenance models?
- How can Virtual Reality techniques be applied in teleanalysis?

5.1.1 Motivation

Service technicians aim to optimize the plant regarding cycle duration, material abrasion or energy consumption. An additional expert appraisal would help a lot, but it is too expensive in most cases, because the expert would have to travel to the production site. The aim of this chapter is to develop remote interfaces, which allow a remote analysis of the plant.

Optimization is a difficult process. At first it has to be defined what means "optimal" on which kind of data. Hence, an analyst has to understand the setting and the process of the plant in order to perform any optimization. Furthermore some standard optimization procedures have to be defined like time optimization, the minimization of abrasion or energy consumption. All these measures aim to reduce costs. It is important to understand, that even small optimization changes lead to large financial savings, as the plant is producing 24 hours a day, 7 days a week.

Second, communication in optimization processes is indispensable as depicted in

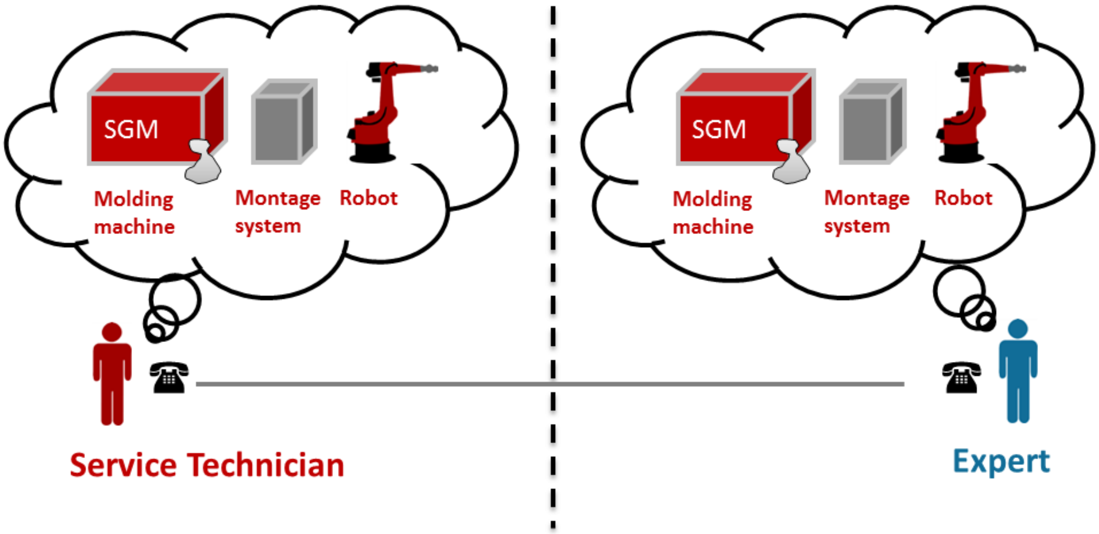


Figure 5.2: *Understanding problems in teleanalysis*

Figure 5.2. The service technician at the production site, who is familiar with the plant, has done his or her best to adjust the plant parameters. He or she now tries to help the remote expert to understand setting, process and dependencies. At first, the expert needs to get all the relevant information. He or she will then evaluate optimization possibilities. Later on, the expert needs to inform the service technician about the necessary changes on the plant and the process. In order to perform those communication tasks efficiently, they both need a mutual understanding ("common ground", cf. Section 2.4.3) on the setting, the process and the dependencies.

5.1.2 Goals

There are different optimization goals: First, a reduction of the plant cycle duration directly leads to a larger amount of parts, being produced in this plant and hence to lower production costs of the part. Second, if the abrasion can be decreased, fewer maintenance and repair efforts are necessary. In this context, it is also very helpful to determine the optimal time for maintenance by abrasion detection mechanisms (cf. Section 2.1.2.1). This task can be performed by the Condition Monitoring system. Finally, a reduction of energy consumption directly saves money. Of course, all those optimization goals are interdependent, as for example a faster axis velocity might on the one hand result in a shorter process duration, but on the other hand might increase abrasion and energy consumption.

A larger picture of the complicated optimization process provides Figure 5.3. On the lower right, we see a plant with several components (molding machine, industrial robot and montage system) like the one which has been the core of the project MainTelRob. The optimization goals described above apply for the plant. But it is very difficult and requires a high amount of expertise from different fields. We

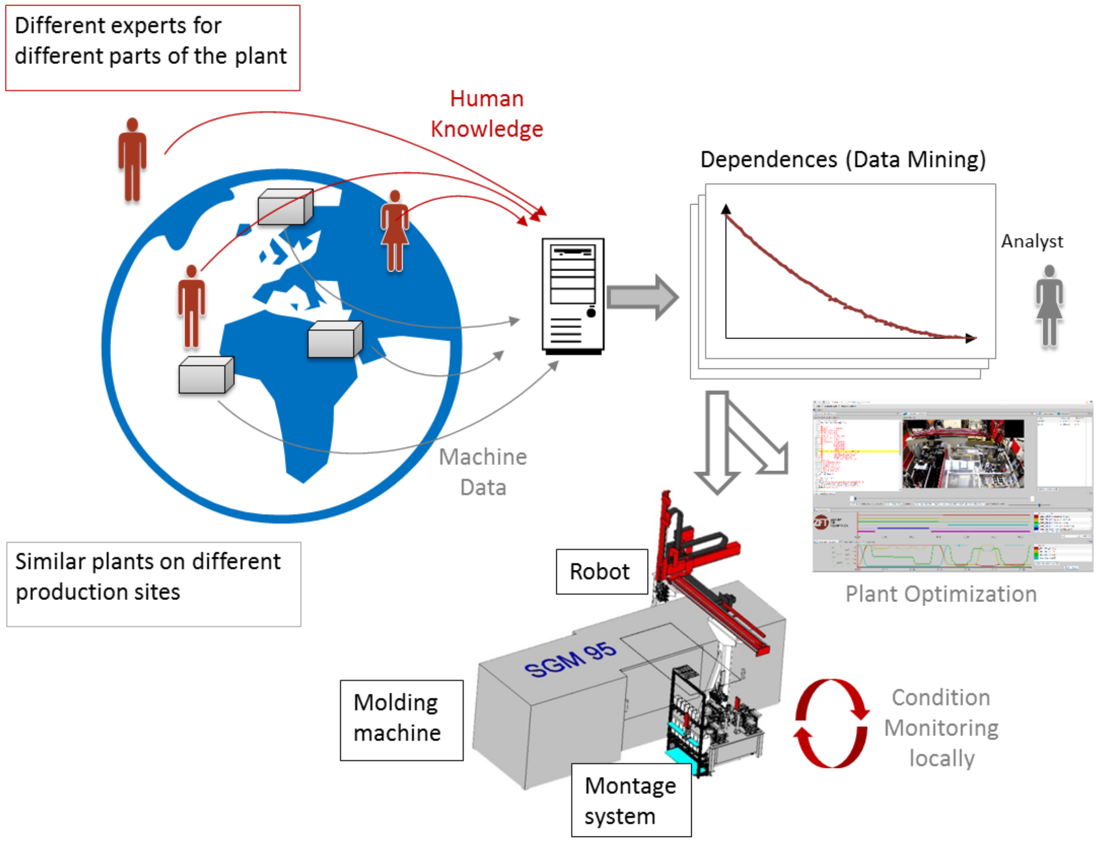


Figure 5.3: *Relation between Scenario 4 and 5*

can assume, that neither the plant nor the condition monitoring is running in the optimal configuration.

For each part of the plant, one or more experts are scattered over the whole world (depicted on the left). With the help of a computer supported teleanalysis tool, which enables them to understand the plant and the process, they would be able to suggest changes in their particular field of expertise. Only with the experts working together with the service technician at the plant, a better plant process according to the mentioned optimization goals can be achieved.

We also assume, that there are similar plants, which are established in different production sites all over the world (depicted in Figure 5.3 on the left). At those sites, special trained people reside, who may have records on repair issues and plant performance. So we have human knowledge on the one hand and machine data on the other. If this knowledge would be connected in a common server, it should be possible to find common dependencies, which can be fed to the local condition monitoring or general optimization rules. The chance of finding "known bugs" or "known dependencies", on a huge amount of plant data like for example the estimation of optimal lifetime for gears, has a high probability.

5.1.3 Related Work

There is increasing interest on data mining approaches in the production industry [157]. But there are several reasons, why the development of data mining for the automation industry is approaching rather slowly: On the one hand, the majority of the scientists with knowledge in the industrial domain are inexperienced in data mining algorithms and data mining software. On the other hand, data mining scientists lack the detailed knowledge of complex plants and production processes. The few experts with knowledge of both domains often are not allowed to access the sensible and protected factory data [157].

But especially the use of sensor information for process control has a huge potential for optimization [158]. The semantic analysis of the data of several plants is expected to provide more information than the analysis of only one plant, but the interpretation of the data mining results is not trivial [157]. So a huge room for maneuver for optimization is expected on a large scale [159].

A survey of applications is provided by [160] or [161]. The use of knowledge databases and rule-based diagnostic expert system is covered by [162]. Their main goal is to provide coworkers with fault detection and diagnosis decisions. Those decisions can also be suggested by an inference engine, while the knowledge databases are generated from experiences from human experts [163].

A software framework called "PDP-Miner" has been deployed since 2013 for plasma display panel manufactoring [164]. It enables data exploration for expert users, data analysis with an algorithm library and result management, i.e. visualization in graphs.

Another framework called "Learning-based Interactive Visualization for Engineering design (LIVE)" provides software assistance for engineering design tasks with an integration of user centered visualization and data mining [165].

The visualization of fault diagnosis results is vital: The large amount of data cannot be understood by humans without visualization. A method for facilitating the understanding of analysis results in the context of objects is the use of 3D data and Virtual Reality. This approach has been implemented for fault diagnosis for aircraft brake systems [166]. Another system which uses 3D data for failure diagnosis is proposed by [167] for rocket engines. The sensor data is processed with statistical methods, and anomalies are detected autonomously. At the end, the state of the valves is displayed in a 3D model.

5.2 Condition Monitoring

5.2.1 Concept

The fourth scenario of MainTelRob covers condition monitoring (see Chapter 3). Goal is the autonomous surveillance and logging of the situation in the work cell and the work processes to detect automatically disturbances and deviations from the normal situation and react adequately.

The robot control system stores system state, I/O data, current gradient etc. in system variables. They can also contain sensor data from connected external sensors, imported via the user program or the SPS program from an I/O port. The system variables represent the manipulator's state and the state of the process. All system variables can be accessed globally from different control functions.

The condition monitoring for KUKA Industries "ROBOTstar VI" software developed in the project MainTelRob is a new software module in the robot controller. It consists of surveillance functions that are running in the background. They compare different system variables with a preconfigured threshold and triggers reactions, if necessary. It is also possible to record the data in a file, so it can be analyzed or visualized offline.

The module uses the system variables provided by the robot control system to monitor the state of the manipulator or the work environment. At the initialization of the monitoring, different combinations of sensor values can be predefined, which characterize a relevant situation in the plant process. This allows event triggered and time triggered reactions.

System variables can be linked to each other and new variables can be calculated. This function is available, if a customer needs additional information (e.g. energy consumption of the plant). With the variable processor, SPS program or user program new information can be calculated dynamically out of basic information. The new data will also be stored in system variables, which can be monitored or recorded.

5.2.1.1 Recording information

All system data can be recorded with the record function in real time and can be stored in the robot control system as a file. The user is able to configure the variables and the recording cycle freely. Different variables can be recorded with different sample frequency. The logging of the data is done with the highest frequency (IPO clock of the manipulator is 11 ms). For minimizing the storage space, an automatic data reduction can be carried out.

It is possible to end the recording file after the occurrence of an event or after a predefined time. The closed file can be uploaded via FTP or can be fetched by an FTP client from the robot control system, which provides an FTP server.

5.2.1.2 Reactions on events

The Condition Monitoring has the following possible reactions on events:

- Triggering a message on the teach pendant (error, warning)
- Writing on system variables to document the event. These system variables can be accessed by the user program and lead to process changes.
- Setting a SPS-marker to influence the SPS process

The monitoring function can be initialized with a String parameter which uses RSVI commands, for example: `IF _RTEMPERATUR >50 Then Error:`

`Temperature too high`. It is possible to compare the current sensor data to a reference curve with has been recorded. Deviations can be used to trigger a reaction.

5.2.1.3 Architecture

The concept is kept as open as possible. By using basic data types and the possibility to combine them, the user can access an infinite number of possibilities. The architecture is presented in Figure 5.4.

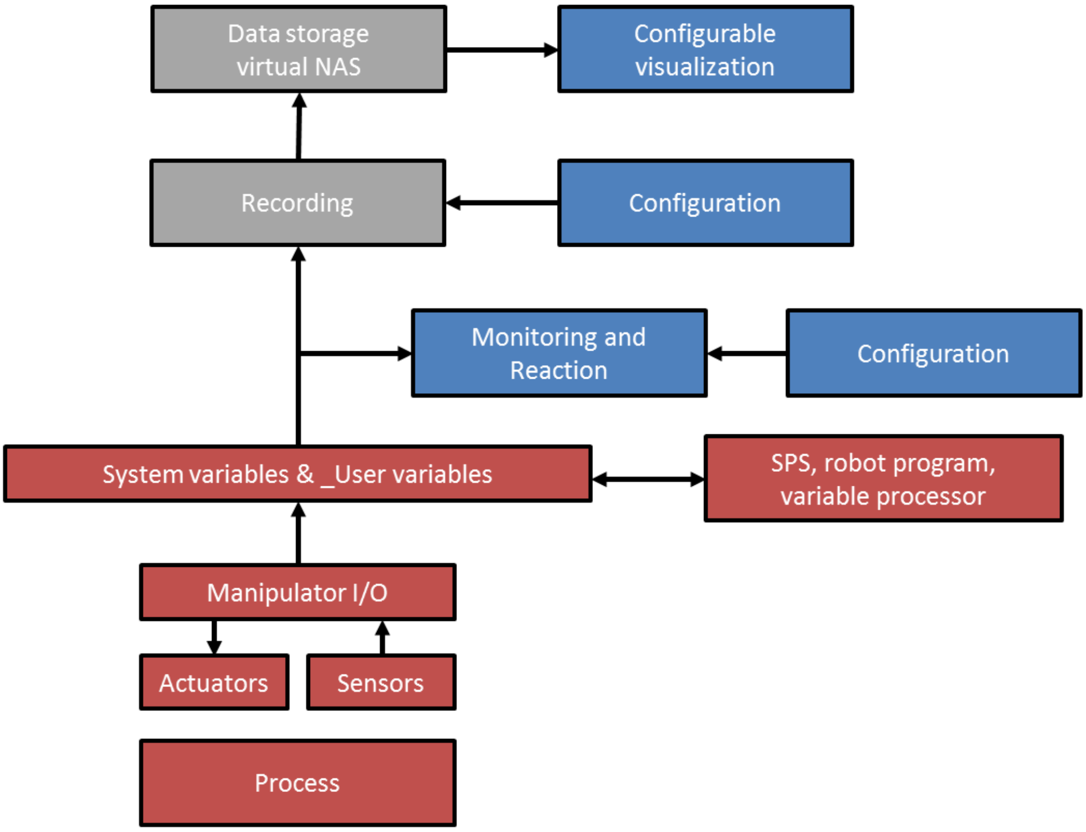


Figure 5.4: *Relation between Scenario 4 and 5*

5.2.2 Implementation

The use case for the fourth MainTelRob scenario defined in the specification document is to detect slowly progressing deterioration. For this, a check on thresholds during the production was realized.

As described above, the operating data is stored in the robot control system, and those system variables can be measured, recorded and evaluated. The check involves two steps: monitoring an error threshold, which leads to the stop of the plant and a warning threshold, which triggers a message on the teach pendant. This is used as an exemplary use case: If a system variable changes a lot during the production process, a time dependent check is needed. In Figure 5.5, a screenshot from the scenario approval is depicted.

Figure 5.5: *Screenshot of teach pendant with warning message*

The monitoring can check selective defined plant configurations or record the whole movement process. If the selective method is used, the user needs to choose a threshold value. In each production cycle the values are retrieved from the system variables, compared to the threshold and stored on an external medium.

The recording of a movement sequence is realized with a sample function. It is parametrized and controlled via the system variables. Thus, an optimal adaption of the recorded values to the movement sequence is realized. The analysis of the data is only possible after the file has been closed. It can be realized by the robot control or with other software (e.g. Matlab). The robot controller GUI "ProVis" also can depict the measurements (can be seen in Figure 5.6).

There is no good method for comparing the recordings of different cycles to one another and to detect abrasion. We come to this problem later in Section sec:CAT.

Figure 5.6: *Screenshot ProVis with depicted sensor data*

5.3 Machinery optimizer

5.3.1 Contextual analysis

The goal of the fifth scenario of "MainTelRob" is "Optimization with external help" (see Chapter 3). Similar to the fourth scenario, this is a complete new development and that is the reason why contextual inquiry is difficult. We decided to ask the service technicians and deciders about their ideas and documented the answers in our qualitative contextual inquiry.

The most important finding was, that all plant parts need to be regarded, although the project focused originally only on the industrial manipulator. As already described above, there are different optimization goals. The most important one is the minimization of the cycle duration, so that more parts are produced in the same time. The operator of a facility also wants to minimize the energy consumption and the abrasion. When we spoke to the service personnel from KUKA Industries, it turned out that the remote expert knows the industrial robot quite well, while he or she is unfamiliar with the plant setting and the other parts of the plant.

According to the interviews with the service technicians, everybody "just does his best to set up the plant". The technicians use their experience with similar plants.

Nobody had ever looked on a combination of the different parameters, because it had not been possible. They also had not tried any additional optimization algorithms, because every part of the plant does the algorithmic optimization internally. If no failures had happened, the technicians had used those parameters.

We asked for the failure and repair history of the plant and similar manipulators. Unfortunately, no conclusions could have been drawn from that data. There has been no attempt to gather continuously data from the robot or the plant because the condition monitoring software has been developed for the MainTelRob project and did not exist before.

At last, we tried to perform a cycle analysis of the plant on our own in order to get a better understanding of the task. We recorded video material from different viewpoints on several parts of the plant. After the manual synchronization, a video with all viewpoints had been analyzed frame by frame (frame: 40ms). The cycle was visualized in a table with a granularity of 200ms. Together with experts from P&G this cycle representation has been supplemented by the involved signals. The chart has been discussed with experts of P&G for optimization potential on the cycle duration, but showed no results. For other optimization tasks, additional sensor information like energy consumption was needed, which could not be provided in the videos. The amount of work needed for the scientist was a whole week while the company P&G provided a highly trained person for two days.

But his video based approach lead to the envisioned flow model in Figure 5.7. We wanted to provide software that allowed the collection of data and video stream simultaneously and store it in one data package, which can be transmitted to experts around the world. Our first idea was, that those experts should be able to display both video and data in some kind of extended video player.

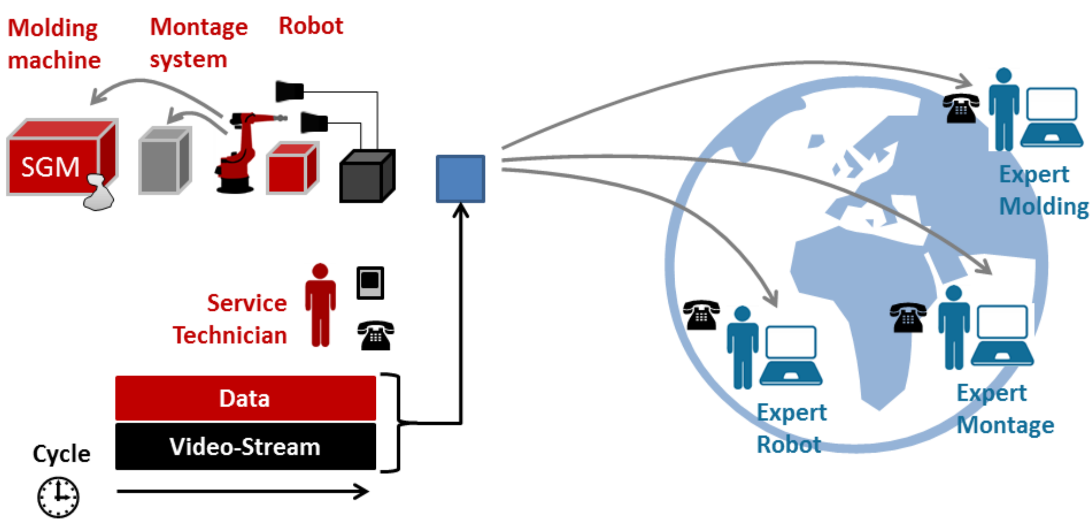


Figure 5.7: *Envisioned flow model for optimizations*

5.3.2 Prototype

First of all, the manipulator hardware needed to be set up. The industrial robot software was modified with the new developed condition monitoring software in order to record all accessible sensor information. The dataset of several plant cycles has been recorded during the biweekly remote access, which will be described later in Chapter 6.

The design challenge was to provide a clearly arranged view on the one hand and provide access to a lot of information on the other hand. We did a lot of paper prototyping (see for example Figure 5.8) which was discussed in detail with the project partners. Next to the video on the upper right, a selection menu for the sensor data should be provided. On the left side several data plots are shown with a common y axis (time). It should be possible to compare the sensor plot to thresholds (upper screen), compare the plot to another with tolerances (middle screen) or just to plot several different values (like position, velocity and current) in a common diagram (lower screen). Several other possible applications had also been visualized with wireframes. This method helped a lot to unite the needs of the end users.

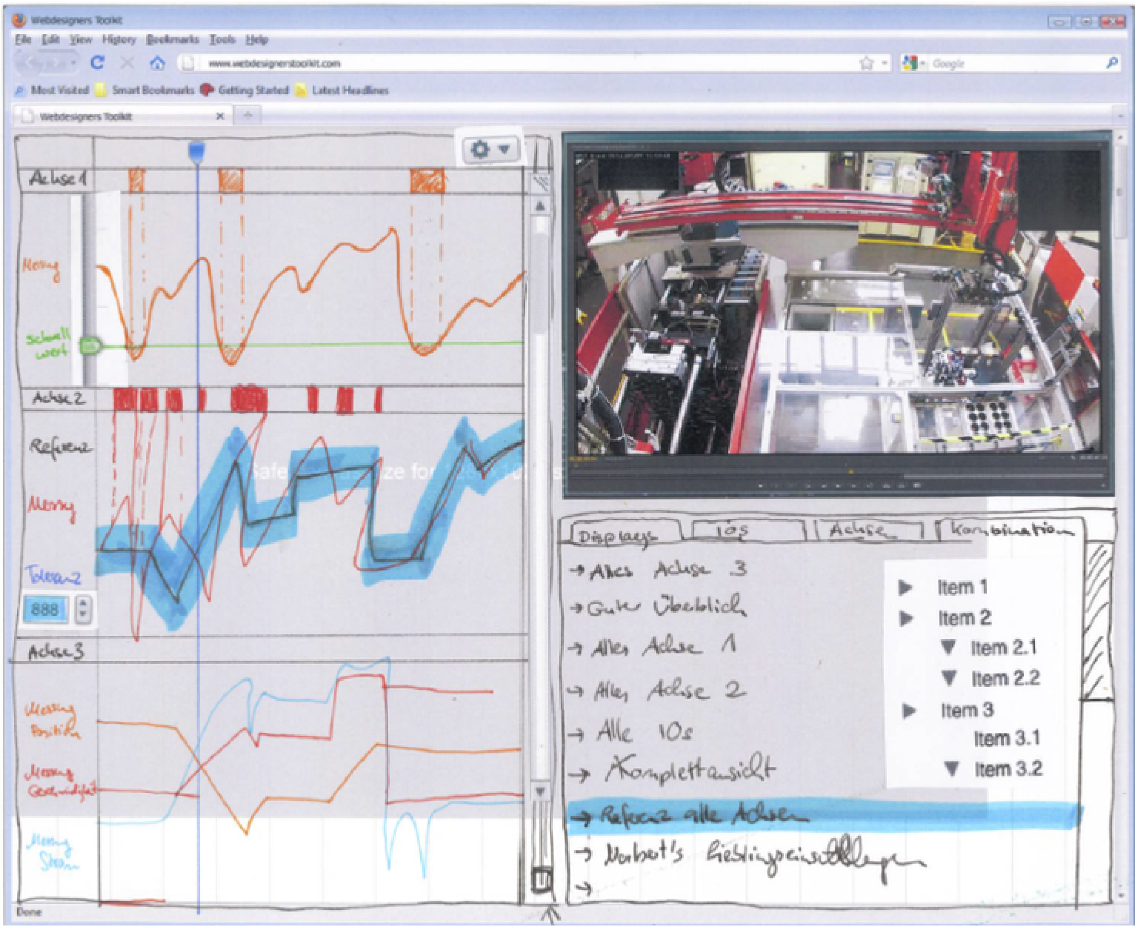


Figure 5.8: *Paper prototype for optimization*

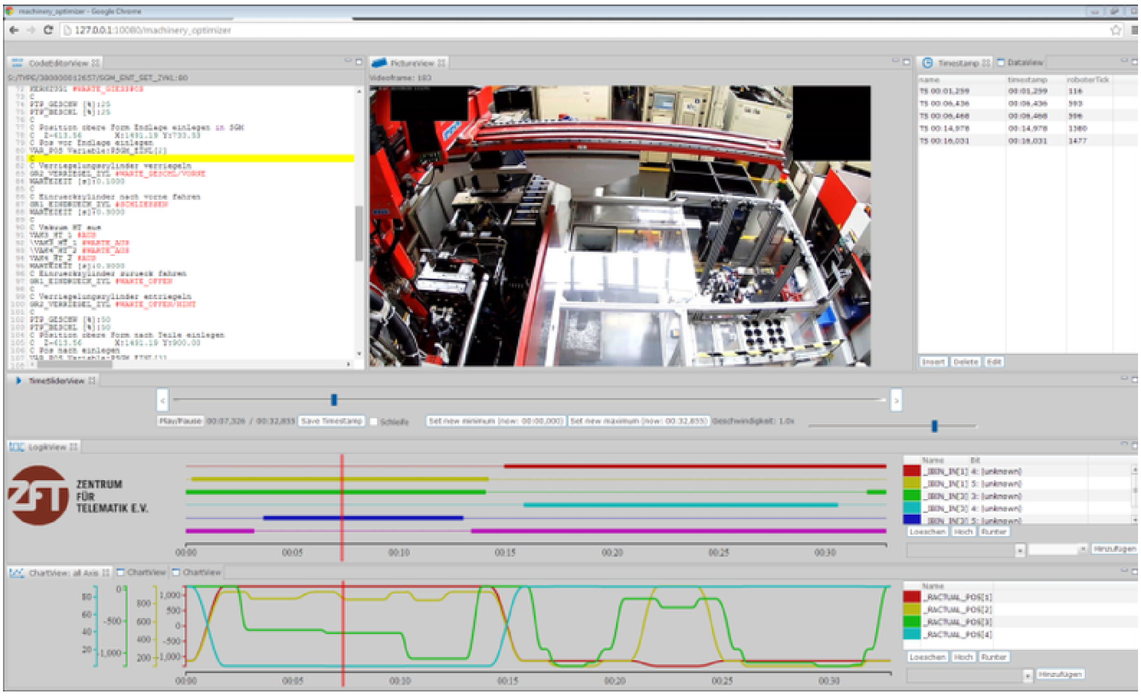


Figure 5.9: *Prototype for optimization*

At first, we describe the user interface presented in the official acceptance meeting. It plays the video data dynamically like any video player. But also the robot data is changed dynamically. The information fields shown around the video player can be configured freely.

A detailed view on the interface is presented in Figure 5.9. In the play bar in the middle of the screen, the recorded package can be played with different velocities. It is also possible to play parts of the cycle in a loop for a deeper analysis. On the upper left part of the screen, the source code of the user program is depicted, while the current code line is highlighted yellow. In the middle there is the video view presenting the view of the top camera with black masked areas (for more information see Section 6.2.2.2). On the upper right, there is the so-called timestamp view, where different interesting timestamps are listed, which can be addressed directly.

On the lower half of the screen, the recorded sensor data is depicted. The current time of the play bar is indicated with a red vertical bar. There is a large number of different system variables and the user needs to configure which data he wants to see. The GUI distinguishes between binary I/O data (e.g. molding machine opened yes/no) depicted above the representation of the continuous data (e.g. position or current). Those two windows can be preconfigured, the configuration can be stored.

The prototype shown in Figure 5.9 attracted a huge interest of both project partners and was successfully accepted in the formal acceptance meeting. But the pluralistic walkthrough revealed a lot of user uncertainty. Some of those remarks could be solved by adjusting parts of the software but the major design challenge of intuitively providing the huge amount of possible sensor data still has no really satisfying solution. Nevertheless, the feedback of both project partners was absolutely positive as this is the first integrated view of recorded robot data together with video information. Both partners plan to use the tool and continue development.

Figure 5.10 provides the final state of the software used for the long-term evaluation. Now the file system on the manipulator software storage can be accessed (top left). Another big progress has been made regarding the timestamps. It is now possible to compare analog sensor values to thresholds (red central line at the bottom view and automatically generate timestamps, when the value passes this threshold. Also upper and lower flags of binary values can be extracted as timestamps. The timestamps are also highlighted in colors on the play bar in the center.

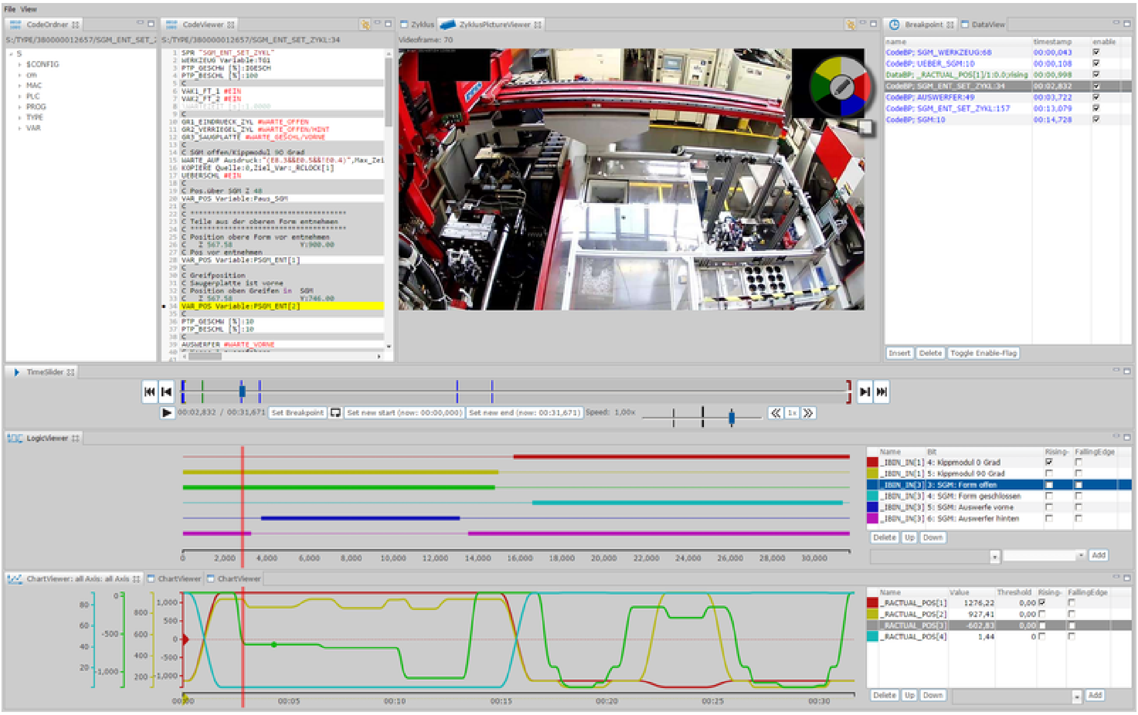


Figure 5.10: *Final prototype for optimization*

5.3.3 Evaluation

It is challenging to design a usability software evaluation for an expansive software tool like the "Machinery Optimizer". The biggest challenge was to find test persons. There is only a small amount of people who have the expert knowledge needed to optimize such a plant. Also there is no reference software, which can be used to compare the results. Our only reference measurement was the duration of the manual analysis of the contextual analysis.

We asked our project partners to provide expert participants for the software. We needed to design an experiment providing those experts with enough flexibility. The software and the data were provided on a web server. The experts scheduled their preferred time and used the software via browser. The participants were supported by a manual and extensive work instructions, in which they filled in their feedback (working steps, time, questionnaires, feedback). A scientist was available on the phone, but due to the spatial distance and the required flexibility no direct observation of the experiment could be provided.

A pilot study had been carried out with another scientist, who did not take part in the project so far, but is accustomed to KUKA industries robots. The main study was performed with nine experts from both of the project partners. Two participants cancelled the experiment due to technical problems (internet connection was too slow) so that their values are incomplete and not included in the following diagrams.

After a familiarization task ("play-around") the participants needed to accomplish three major tasks with randomized order for each person:

- One task was to decrease the cycle duration.
- Another task aimed to minimize the abrasion during each cycle.
- The most complicated task comprised the estimation of hypothetical savings on cycle duration by means of the introduction of a second manipulator.

Figure 5.11 depicts the average time needed by the participants. Time optimization was the task, took took longest (order of tasks was randomized to avoid acclimatization effects).

With the new tool, the recording of a plant cycle can be done within half a day. These results show that the participants were able to complete tasks in less than three hours. If we compare that to the duration of the manual approach (1 week and two days), the tool clearly saved time.

For each task, the questionnaire NASA-TLX was used to measure the perceived load. The test persons can tick values from 0 to 100. The average load depicted in Figure 5.12 is quite normal, while a higher demand of attention for the task with the second manipulator is clearly visible. Considering the separate values measured by the NASA-TLX, there is a clear peak on the mental load as expected (see Figure 5.13).

The perceived load is connected to the task on the one hand, but also to software usability on the other hand. The participants were asked to rate the software after they had finished the tasks. The QUESI questionnaire for the intuitive use was used here with results presented in Figure 5.14. The maximal value (for the best usability) is 5. The values are good, but can be improved. This is why we think, that a part of the measured load described in the previous section can be attributed to usability. So additional work needs to be done.

Additionally the ISONORM questionnaire was used, whose values reach from -3 (very bad) to +3 (very good). The averages can be seen in Figure 5.15. They are positive, the prototype seems to work quite well, but we clearly see a need for improvement in the area of self-descriptiveness. This is also supported by the qualitative data.

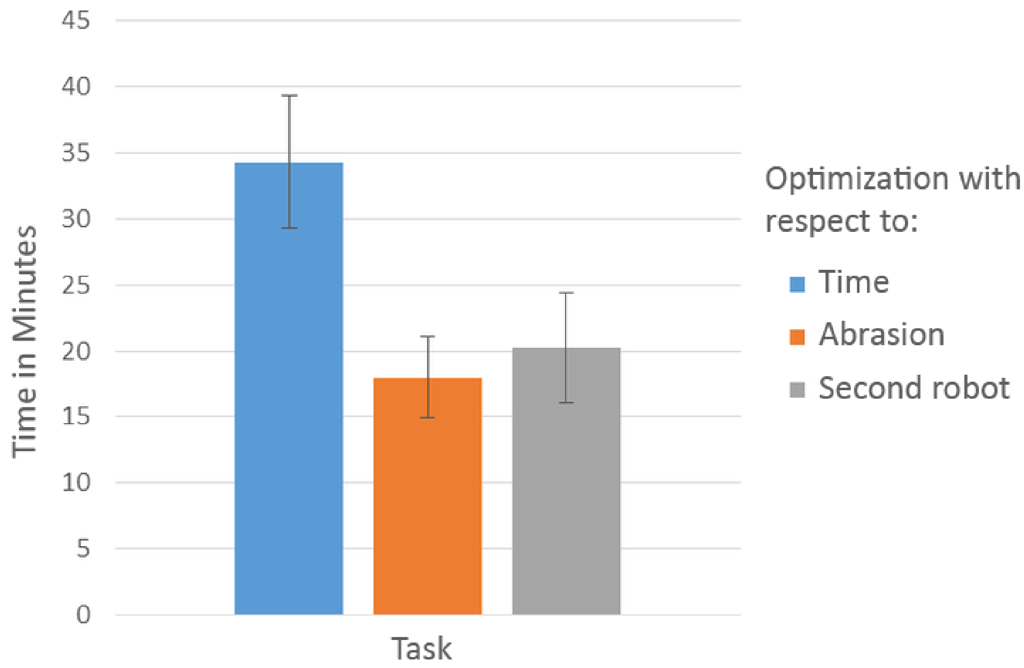


Figure 5.11: *Average TLX values per task (time optimization, abrasion optimization, second manipulator)*

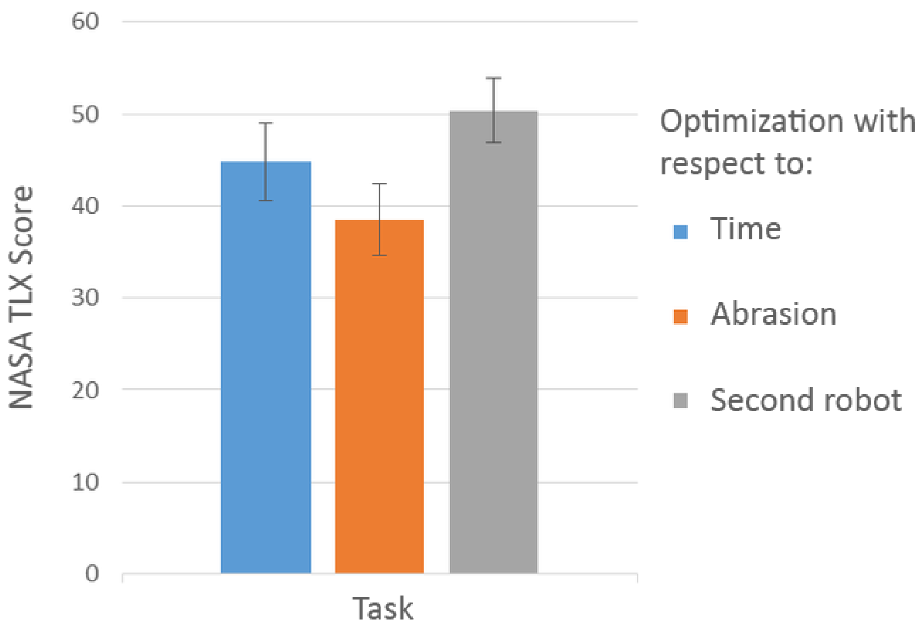


Figure 5.12: *Average TLX values per task*

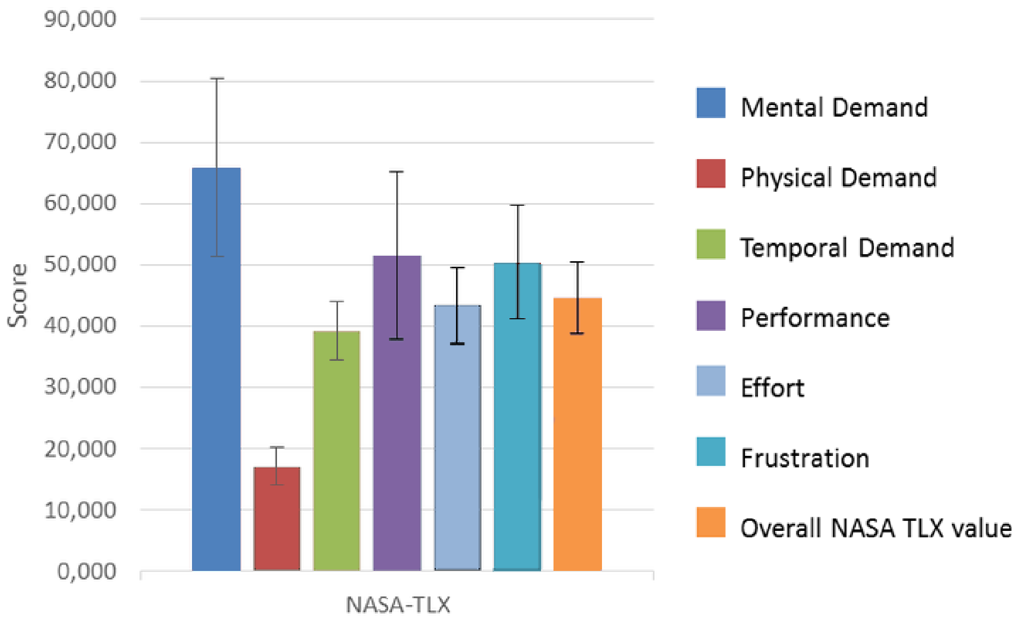


Figure 5.13: *Seperate TLX separate values*

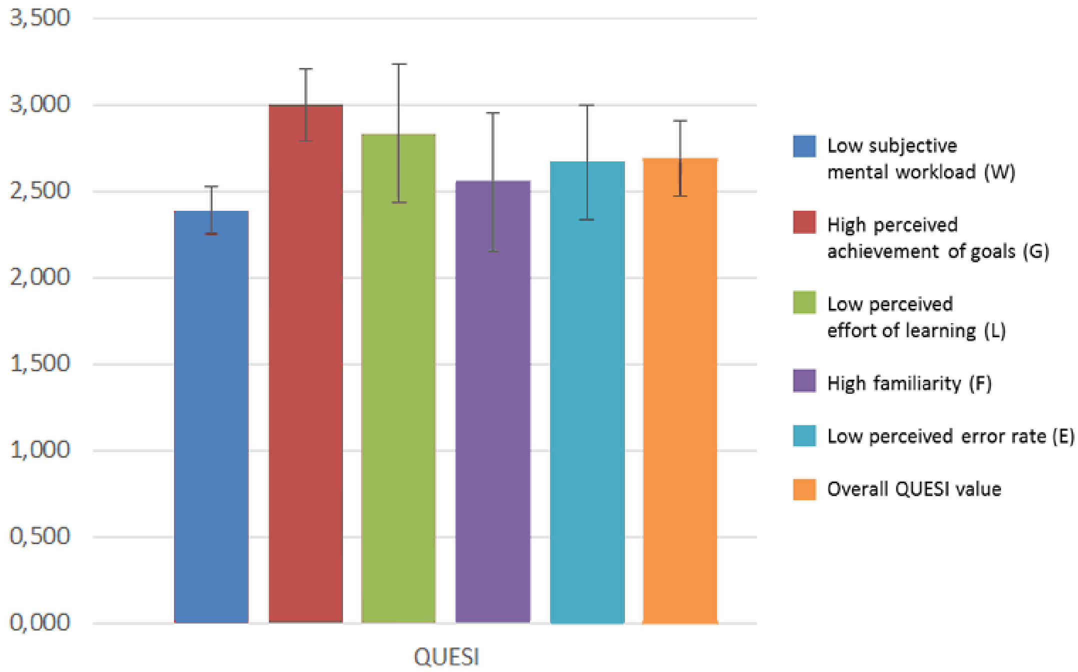


Figure 5.14: *Average QUESI values*

The first part of the qualitative data analysis is based on the comprehension questions which are posed after the initial play-around part. They prove, that

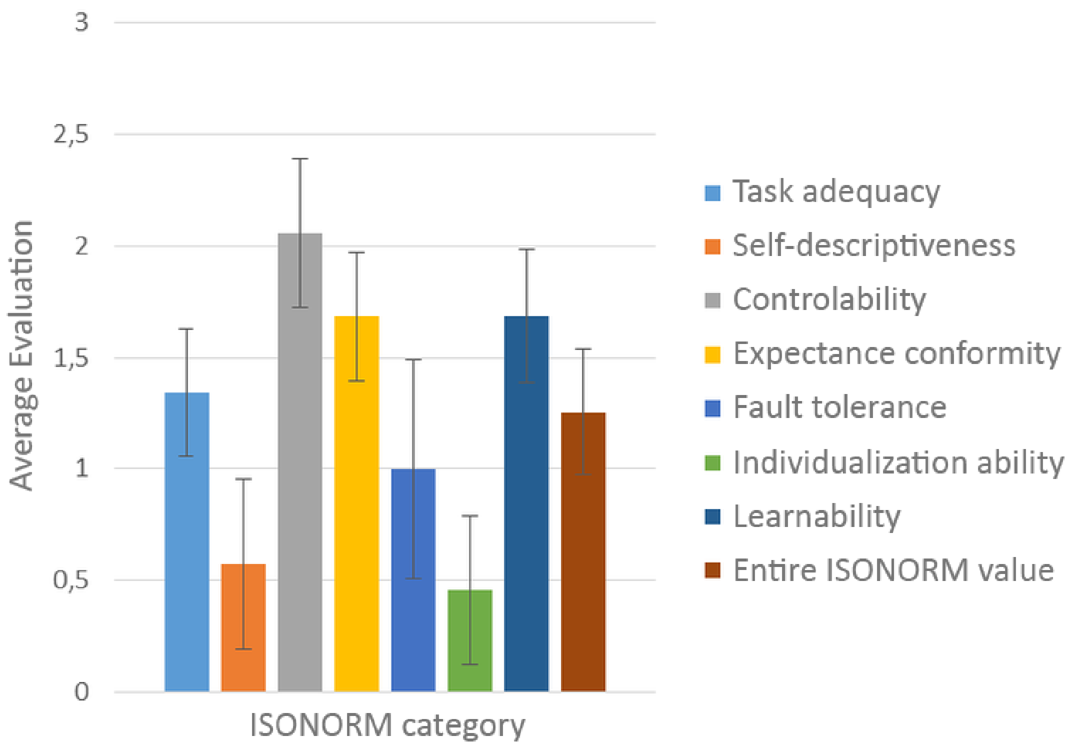


Figure 5.15: *Average ISONORM values*

the tool helps to answer direct questions, like "How long is the cycle duration" (answer: 31,671 seconds), as those questions has been answered 100% correctly by the participants. Also more fine-grained questions like "What is the robot doing on 00:00.727?", "Which code line is executed in this moment?" were answered 100% correctly. We also asked the participants to write down the observed production process. This cannot be directly observed in the video and only one test person, who had not known the process before, was able to answer this question correctly.

If we compare this to the correct process description in Figure 3.7 in Section 3.3.1, it becomes clear, that the second production part (soft plastic) is missing. This naturally results in a false estimation of optimization capabilities, especially on the task with the second manipulator.

Nevertheless, the participants uttered on average three approaches to solve the optimization exercises on cycle duration and abrasion optimization. The people already familiar with the plant were able to make more precise suggestions. Some people were able to quantify their duration optimization result, which has been confirmed in the project team afterwards. While the manual process did not find any duration optimization, the use of the software provided insights which may lead approximately to 2 - 3 seconds less, which is 6 - 9 % of the whole cycle duration.

The free comments on the end of the evaluation questionnaire were equally encouraging: Although performance problems occurred (which partly resulted to the particular Internet connection), the participants still agreed on the suitability of the tool for analysis and optimization tasks. Several participants were able to point out specific failures and possible improvements on the manipulator code. Therefore the tool is also suited for finding programming errors. Two participants from the service area even stated, that such a tool would have been very useful in the past for failure analysis. But several participants noted (according to the usability questionnaire results) the difficulty of handling the large number of possible sensor data.

5.4 Complex analysis tool (CAT)

5.4.1 Concept

The idea of the next application is the comparison of different cycles. This is the result of the problems discovered in the other applications. As described in Section 5.2, the Condition Monitoring itself can not determine useful thresholds for warning or error messages. The service technicians can guess a threshold from their experience. But for some variables, there is no fixed threshold but a tolerance around the optimal data behavior. The Condition Monitoring can plot several cycles but is not able to compare them. This would be necessary for the detection of abrasion. The Machinery Optimizer 5.3 can plot several sensor data of the same cycle in one diagram, but also lacks the possibility to compare several cycles. As described in the evaluation, the participants discovered problems with the huge amount of different sensor data. This problem increases, when several cycles from different plants need to be regarded.

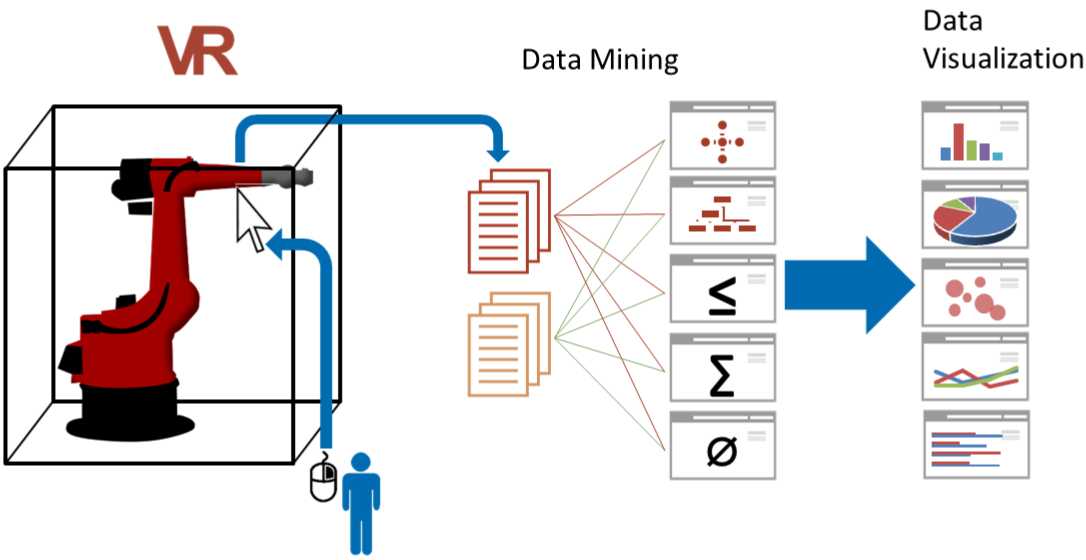


Figure 5.16: *Concept idea of the Complex Analysis Toolkit (CAT)*

The idea of the "Complex Analysis Toolkit" (CAT) is depicted in Figure 5.16. A 3D Model of the plant is used for the interaction with the user. He or she can choose the required data by clicking on the parts of the plant. This results in several data sets which can be processed with data mining methods. For this step, we use the TARDIS software described in the next section. The results are then presented to the user with various visualization methods.

5.4.1.1 TARDIS

The "Telemedical Applications with Rule-based Decision- and Information System" (TARDIS) [168] developed at Zentrum für Telematik was used for the data mining part of CAT. The software allows creation and execution of rules (compare Section 2.2.3.2). The generic telemedicine system is flexible, easy to use, reduces the workload of the medical personnel and involves the patient into the treatment. To meet these conditions, a rule-based approach provides intelligent assistance for the physician and a smart component for the patient adapts to changing environments and gives feedback about the treatment to the patient. The rule-based decision system was intentionally held flexible and platform independent to allow reusing the system in other fields of application such as industrial telemaintenance. It is used to model individual behavior of the system, depending on the patient's disease, treatment and the used devices. It allows to add autonomous functions to the patients mobile device, data preprocessing and feedback to the user even without an active data communication to the medical center. In the medical Tele-Service Center, the framework can be used used to filter and analyze data or adapt the treatment of each individual patient. The system was implemented and tested in real life with patients suffering from chronic obstructive pulmonary disease with a very good feedback. TARDIS comes with a graphical editor that allows an easy creation and editing of rules while showing the flow of data visually.

5.4.1.2 Additional TARDIS modules

To ensure access to data and an essential functionality, TARDIS is extended with several modules:

- **Interval:**
  Two fixed input values are assigned to the module, which set the upper and the lower limit of the interval. Further, a third optional parameter is given as constraint, to define the step length of the interval. 1 is set as a standard value for step length. Since the function is not dependent of data beforehand, the existence of selected data can only be tested later.

- **Array:**
  To enable the selection of specific values, the String module is utilized for arrays instead of introducing a new module. Data can be specified as comma separated values or by an interval diction (e.g. "1-10, 12-20"). Thereby, the String module is used as a constraint and analyzed by the respective modules.

- **Plot:**
  The Plot module serves as data drain and therefore has an input able to process several data sets. Besides complex depictions, the plot function also can visualize simple inputs like a data set from M-Data.

- **Log:**
  Another data drain is generated as a Log module to display single data points. The mounted input signal is passed to a textpad in the user interface to enable the display of single values.

- **Least Square Fit:**
  To control preprocessing of data in the rule editor, a module is generated for Least Square Fit (see Figure 5.17). The module calculates three results for a prompted M-value, which can further be used by selection in the rule editor:

  - Data: A calculated time shift is determined for the first cycle of data from the M-value of the prompt. If "data" is selected as output option, indexes of data are aligned with the time shift and data is returned.
  - Delta T: If "Delta T" is selected as output option, the respective time shift is assigned to each cycle number and returned.
  - Squared Error: Returns the correlation of cycle number to sum of quadratic errors for the first cycle of the transferred M-Values.

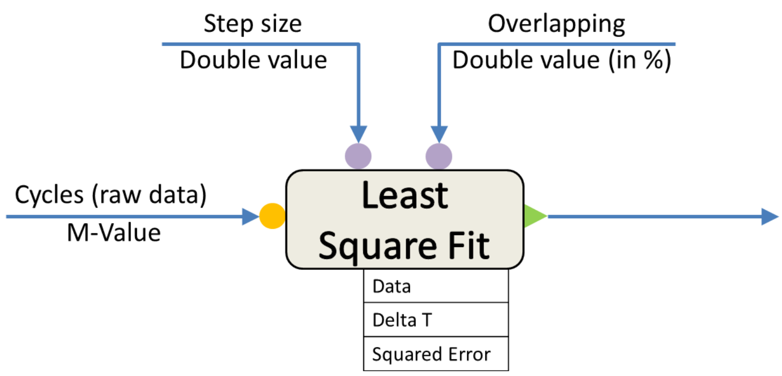


Figure 5.17: *Module Least Square Fit*

5.4.2 Data sources

Handling large amounts of data is a difficult task for every user. Even experts require tools to reduce the amount of data into smaller chunks of data, containing the essential information. The raw data is collected directly from machines and could be collected erroneous. In the MainTelRob project the raw data is collected from two different types of data sources: data is generated from some parts of the industrial robot, while other data is collected by humans at so called Visual Control Points. Both data sets are stored in a joint database.

5.4.2.1 Machine data

The Condition Monitoring stores the several values of the robot's data in a database. The robot can acquire information from other parts of the production unit via the different in- and outputs. The data is divided in three groups:

- Integer variables, like `_ISTEP[1]` (current source code line) or `_ISYSTIME[2]` (current time),
- Binary variables stored in Integer like `_IBIN_IN[1]` (32 bit per variable), like `_IBIN_IN[3] I E8.3` (molding machine is open),
- Position variables, like `_PACTISTPOS[2]` (current position of axis 2) or `_PACTPOS[3]` (current target position for axis 3), and
- Floating point variables `_RCURR_ACT[2]` (current at axis 2) or `_RACTUAL_SPEED[1]` (speed of axis 1).

There are several tables containing the names of the values and a short description, that are used in the robot control. Most recorded values are continuous values, which are sampled time discrete and value discrete. Each measurement records 54 variables. The measurements are repeated in irregular intervals between 10.8 and 12 milliseconds. The robot's clock is 10 milliseconds. There are different delays in the recording of the values. One cycle of the robots process takes about 35 seconds. This results in about 3000 measurements per cycle with 54 variables each. In one hour about 133 megabytes of robot data is recorded.

In order to use the machine data for data mining we define which variables are queried from the database:

- variable x in one cycle i at a time t: $x_i(t)$
- variable x in one cycle i for the entire duration T of one cycle: x_i
- variable x in n cycles at a time t: $\sum_{i=0}^{n} x_i(t)$
- variable x in n cycles for the entire duration and in all cycles: $\sum_{i=0}^{n} x_i$

First the data is combined in the rule editor of TARDIS, then processed in the rule solver. The function M-Data (see Figure 5.18) is a draft for the usage in the rule editor. The variable can be labeled with a text field in the component. The component also needs two parameters containing the input: cycle and time. Both parameters can be specified as single value, interval or array. If some cycles in an interval should not be considered the array can be used to specify all cycles that should be processed. If none of the constraints is defined, the entire period of all cycles will be processed. For every given cycle the function returns an array of $[t, x(t)]$ pairs: $[t, x(t)] \cdot n$, with $t \in \mathbb{N}$ is the point in time, $x(t)$ the value from the database for the chosen variable and $n \in \mathbb{N}$ the number of chosen cycles. The value $x(t)$ is represented as Boolean, Integer or Double variable. This structure can represent all of the machinery data mentioned above and is called "M-Value".

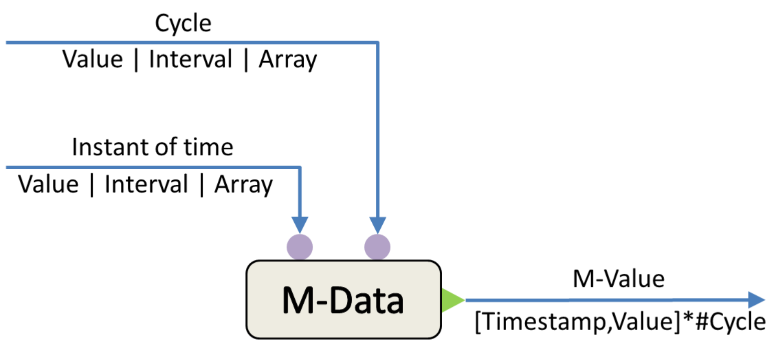


Figure 5.18: *Component M-Data*

5.4.2.2 Visual Control Points (VCP)

As described in Section 4.3, the "Visual Control" application has been realized on a mobile device and tested in the active production environment. The goal for the CAT application was to integrate this data into the data mining, because the human inspection provides a good insight on the status of the plant. At these so called Visual Control Points (VCP), the technician has to perform maintenance tasks. The technician can protocol the current state of this part of the machine and can add a picture of damaged parts or irregularities. The maintenance tasks are repeated in regular intervals. The intervals differ dependent on the VCP. The different intervals are "once per shift", "daily", "weekly" or "monthly". Knowing the different intervals is crucial for using data mining techniques. For the analysis of the data only two states are chosen: "ok" and "not ok". When performing a maintenance task the technician can choose from two other states too: "not accomplished" is used when the maintenance task was not performed in this particular shift. We handle these states as "not ok". An inspection task is marked as "not necessary" for example when it is a weekly task and it has been performed in an earlier shift. The marking "not necessary" is treated like "ok" in this work for simplicity reasons. The data used in this work was acquired during the test phase at a real production line, described in Section 4.3.

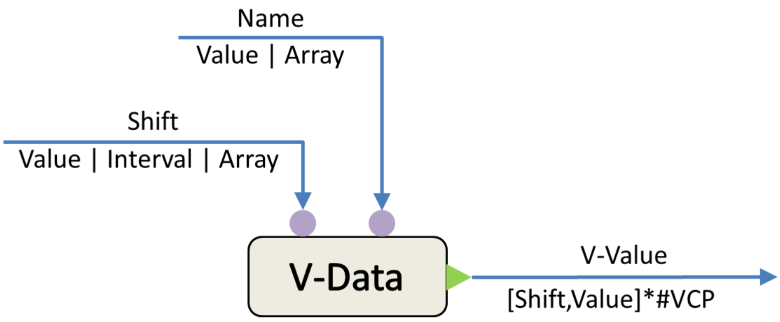


Figure 5.19: *Rule block V-Data*

The data of the visual control points is stored as logical values for the different control points. To be able to use the data with TARDIS, a new block is needed for the rule editor: V-Data (see Figure 5.19). Like the time constraints for the M-Data blocks (see Section 5.4.2.1), a selection of shifts is used as a constraint for the V-Data blocks. Additionally a name can be used to constrain the VCPs. If no name provided, all stored VCPs are used. The result of the block is an array of shifts for the selected VCPs that contains a single binary value per control point: $[t, x(t)] \cdot n$. While $t \in \mathbb{N}$ is the selected shift and $x(t) \in \{\text{True}, \text{False}\}$ is the state of the visual control point in the corresponding shift, $n \in \mathbb{N}$ is the amount of selected VCPs. The value `True` is used, if the control point is "not ok"; `False` is used if the VCP is "ok". The resulting values are labeled as "V-Value" in the following text.

5.4.2.3 Data storage

All data that has been aquired during the project MainTelRob is stored in the same database structure. Access to the data is controlled by the AMS framework (see Section 6.2.4) and is realized using a SQL interface in a Java environment. Besides the data of the robot and the VCP data an additional local database is created for internal communication. In the following we do not distinguish between databases.

5.4.2.4 Data preprocessing

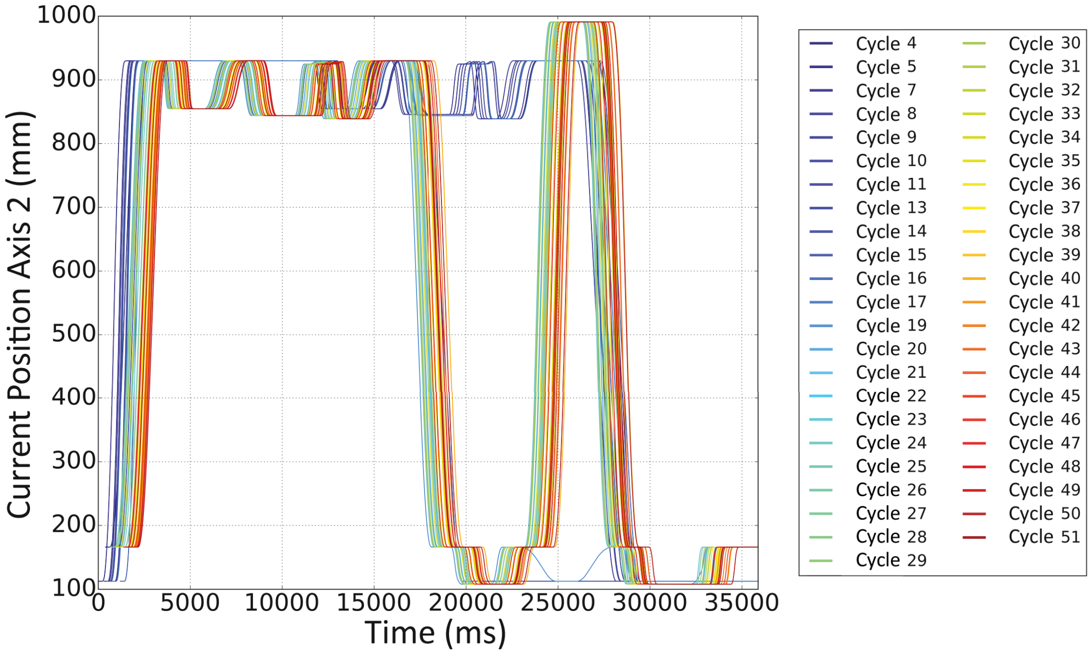


Figure 5.20: *Position of axis 2 in all test cycles*

Preprocessing of Visual Control Points is confined to a hypothesis: If the repetition rate for all values is assumed to be "per layer" (with the simplifications described above), only categorical data remains. It can directly be processed via data mining algorithms.

The machine data needs to be preprocessed. Most of the robot's sensors continuously change, but are read in a pulse of $< 1ms$. The Condition Monitoring is selected once per program sequence and saves data in the database per robot pulse ($10ms$). Measuring time points are set by three variables: date, time in the format hours:minutes:seconds, and millisecond.

Figure 5.20 shows a processed version of variable `_RACTUAL_POS`[2] (position data of axis 2) in the 52 cycles of testing data. Figure 5.20 is the plot of the stored data and only differs in the labeling of the time axis. Since the absolute time stamps in the database show very high values and thereby corrupt the axis labeling, the earliest starting point of all cycles is determined and subtracted from all data. In theory, by using absolute time stamps from the database, all cycles should be displayed successively on the time axis. However, all cycles are indicated in roughly the same time window. Obviously at this point already an error occurs. To compare the course of robot data, the starting points of cycles would have to be changed in the normal case to make all cycles visible in the same time window. The caption of Figure 5.20 shows that not all cycles from 1 to 52 are listed. That is due to the lack of data in the testing database for some cycles.

The example in Figure 5.20 indicates that comparing variables from several cycles is barely possible. The reason for a shifted display of graphs is a mix of several parameters: Cycles have different starting points and various length of complete measuring time. To correct the temporal shift, a slightly modified technique of smallest squares is used: An optional cycle is selected as best-fit curve, to which the remaining cycles are fit best possible. These cycles are shifted on the time axis by a value Δt, until the distance between all points on the position axis and the best-fit curve is minimized. The result is on the one hand a shift Δt, on the other hand it is the minimum of the sum of the quadratic fields on the value axis regarding the best-fit curve. The adapted method of smallest squares below is constituted as "Least Square Fit" (LSF).

But there have been other problems: In the beginning, an equidistant measuring pulse of $10ms$ has been assumed for the data. If together with this assumption, Least Square Fit (LSF) is used to correct the shift of the time axis, a falsified result is produced: With the pulse of $10ms$ being reneged on in already the first cycle, there are extensions and strains on diverse parts of data. LSF assumes equidistant intervals and thus calculates a fictive time shift. By using this shift to correct the time axis of robot data, a false result is received. The problem of different intervals cannot be eliminated by Fourier transformation, because of the low sampling rate and the different sampling distances (see [169] for more detailed information).

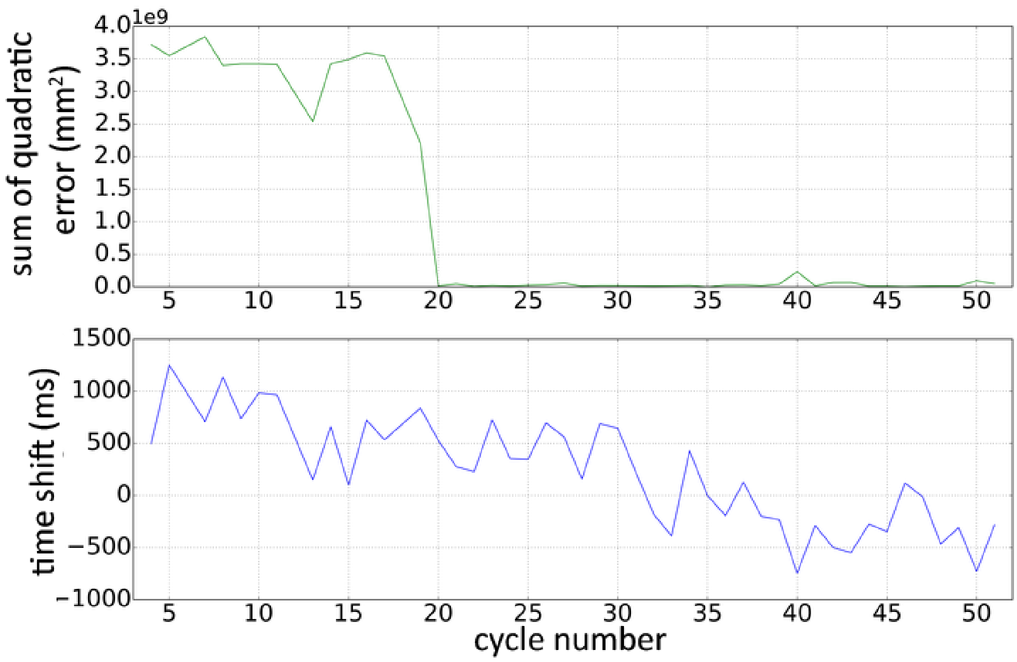


Figure 5.21: *Result of Least Square Fit on cycle 35*

To form equidistant intervals, the data at hand are interpolated on a granulation of $1ms$. Then the LSF is executed on the interpolated data.

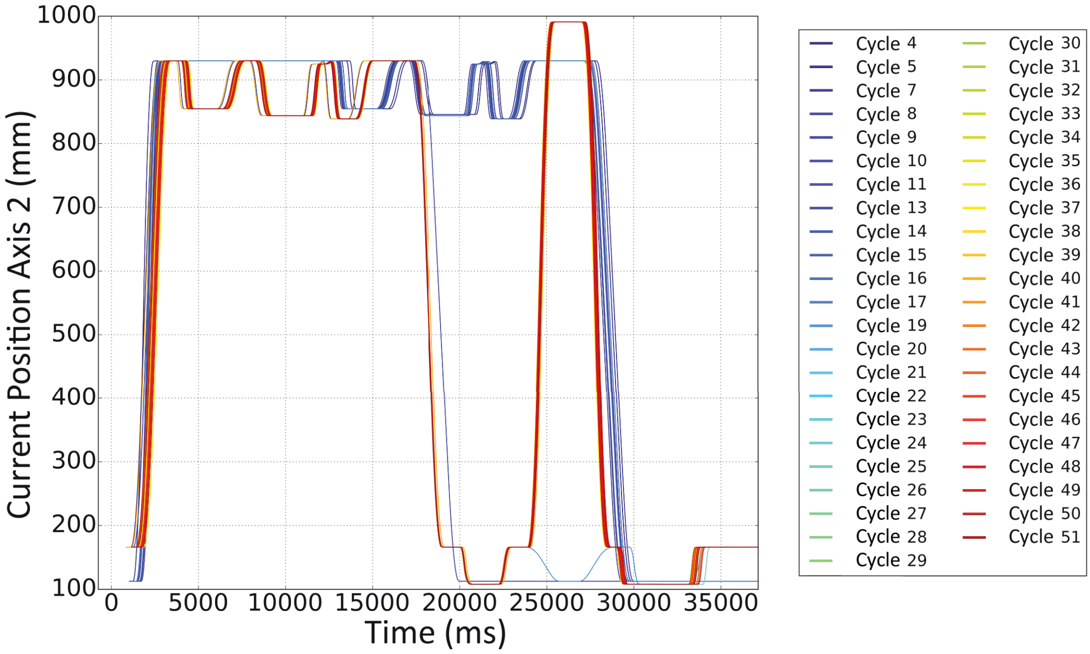


Figure 5.22: *Position of axis 2 after LSF on cycle 35*

In Figure 5.21, the result from LSF for the example of the position of axis 2 can be seen. LSF here is determined with cycle 35 as a best-fit curve. Temporal shifts are within a time frame of approximately -0.8 to +1.3 seconds. The value for the quadratic error shows the extent of deviation between single cycles (x-axis) and fixed cycle 35. Apparently cycles 1 to 19 strongly deviate from cycle 35. This can also be seen in Figure 5.22, which shows the data after executing LSF.

To sustain efficiency during run-time, it seems logical to calculate temporal shift of a cycle for only one variable and use the result for all remaining cycles. To test the similarity of shifts, Figure 5.23 depicts the determined shift for each cycle and variable. A line symbolizes the temporal shift of a variable in machine data per cycle number. If the assumption mentioned above is true and one calculated variable suffices, all variable-lines would coincide with each other. As Figure 5.23 reveals, the shifts in especially the first 19 cycles do not follow a common course. But also the remaining cycles do not show the same curve (see Figure 5.24). For the shift in cycle 35 all tested variables are exactly on the zero point. In summary, comparing machine data over several cycles of the LSF has to be done separately for each variable.

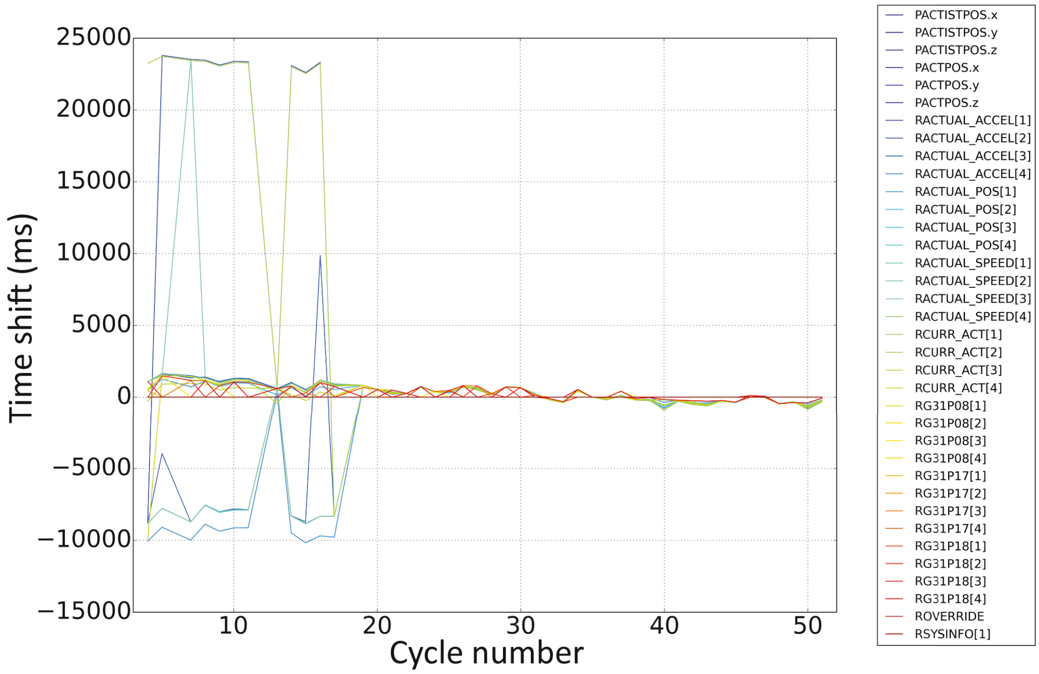


Figure 5.23: *Time shift of all float point variables of all test cycles with respect to cycle 35*

5.4.3 Application scenarios

General use cases are developed, which an expert can comply with the example of an industry robot with the CAT software. Each scenario's application is described, and emerging requirements for TARDIS are explained. Results from the concrete application of scenarios can be found in Section 5.4.5.

5.4.3.1 Scenario 1: Verfication of a value

An expert might want to compare the value of one variable at a specific time $x_i(t)$ to a threshold, i.e. the motor current. The possible over- or undershooting of the threshold value should be tested. This is a function already provided by the Condition Monitoring, and it can be used on a larger data set with the CAT software.

The functionality of different comparisons ($<, \leq, =, \neq, >, \geq$) can already be edited without expansions in TARDIS. Figure 5.25 (l.) shows an example for scenario 1 in the rule editor. On the right side, there is a graphical display for the corresponding threshold test. A single value x(t) (here: t = 5) is compared to the threshold in this figure (blue line, here: 20). The expert receives information about the relative position of selected value versus threshold.

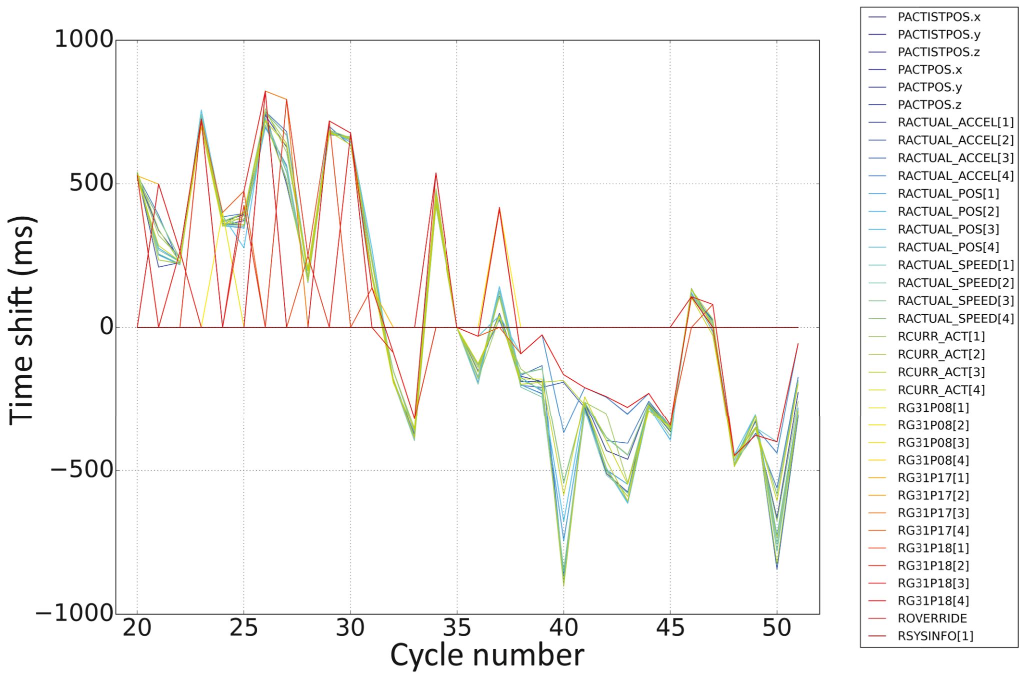


Figure 5.24: *Time shift of all float point variables of cycles 20 to 52 with respect to cycle 35*

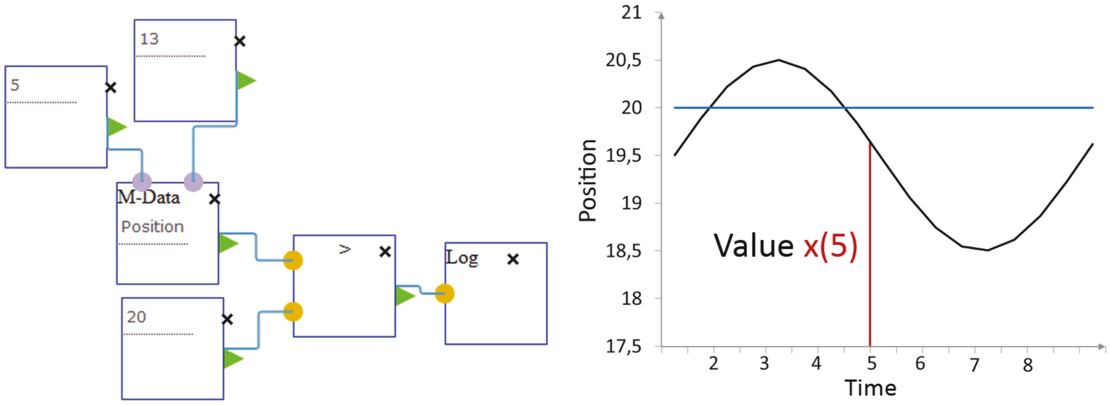


Figure 5.25: *Scenario 1: Rule editor (left) and threshold comparison (right)*

5.4.3.2 Scenario 2: Verification of a cycle

Case of application:

As described above, the Condition Monitoring can also compare an entire cycle x_i to a threshold. A robot's hardware for instance is strained a lot under certain accelerations, leading to an earlier replacement or repair of parts. Therefore the expert analyzes the course of acceleration in the current cycle for a critical threshold. He can thereby locate which movements of the robot are to change to decrease strain.

In scenario 2 all data from one or several complete machine cycles can be analyzed

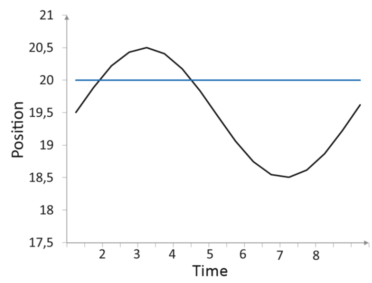


Figure 5.26: *Scenario 2: Verification of a cycle*

for a threshold value. On the one hand, a conclusion about violation of the threshold by at least one value can be drawn. On the other hand, values exceeding or underrunning the threshold can exactly be determined (cf. Figure 5.26). An expert can hence examine the relation of certain data to a critical value. With the exceeding data being returned from the function, the expert can identify the time frame of particular erroneous values. This can also be used to find thresholds which can be applied in the Condition Monitoring.

TARDIS requirements: Comparisons within the rule editor are confined to the usage of single values. For scenario 2, a function comparing several data with a single value is required. Thus a new Threshold module is designed for the rule editor (see Figure 5.27). In the process, one input is occupied by an M-Value and a second one by a double value. The threshold has to set flexible, since for M-Values is not tested if data have Boolean, Integer or Double values. Threshold comparison between integer and double data is without difficulty. For Boolean data, a Double value between 0 and 1 can be used. Each "True Value" for a comparative value above 0 weighs as "above threshold", the remainder counts as "below".

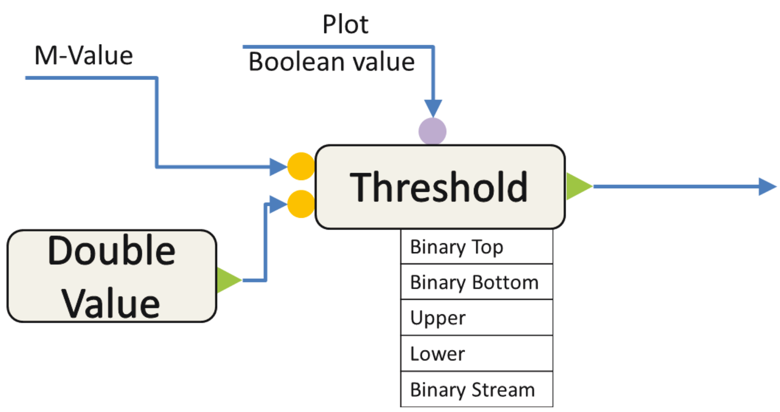


Figure 5.27: *Module Threshold*

The expert can expect various results from the Threshold function. Depending on the application, a graphic display of the threshold test can be a sufficient result.

A "Plot"-constraint can be given as an integer to illustrate a graphic. No graphic is generated if 0 is specified, else it is. But if values from the threshold test should be further applied, they have to be available as input signal for continuing rule editor modules. For example only values above a threshold can be of note. Five versions can be chosen from for the output of the threshold:

- **Binary Top:**
  In the process a single Boolean variable is returned. If the input signal of M-Data has at least one value exceeding the threshold, `True` is returned, else `False`.
- **Binary Bottom:**
  Similar to Binary Top the falling below the threshold is tested. Returns `True` if at least one value falls below the threshold, else `False`.
- **Upper:**
  Here, all data exceeding the threshold is returned as M-Value. Exceeding data is left at its current value, whereas remaining data is set to the threshold at the corresponding time point.
- **Lower:**
  Lower is reverse to the Upper function and sets all values above to the threshold. Data beneath the threshold is left with their original values. Both are passed on as M-Value.
- **Binary Stream:**
  Every single input value is compared to the threshold for output of Binary Stream. 1 is noted for the corresponding time point if the value is equal to or above the threshold. If the value lies beneath, 0 is noted. Thus the output is an M-Value containing binary data at the time points, according to an exceedance or undercut of the threshold, respectively.

5.4.3.3 Scenario 3: A cycle in reference to many cycles

Case of application: It is important to analyze the current data course of an industry robot with respect to an optimal course, particularly concerning Condition Monitoring. Analysis can be run with help of data classification.

Using certain criteria, the course of data can be classified as "Okay" (O.K.) or "not Okay" (n.O.K.). The first cycles during a new robot's operation can for instance be recorded as positive training data, as the robot lacks any signs of wear yet. Current data can be classified according to this training data.

The example of position data of the robot's axis 2 from Section 5.4.2.4 is recharged to substantiate this. Figure 5.28 shows the course of data after minimizing the time shift using LSF. Obviously the values proceed differently. At time point 20000ms a clear division of red and blue colored cycles can be seen. On the basis of training data, the course conforming to the regular machine cycle course can be determined, and subsequently the tested cycles can be classified.

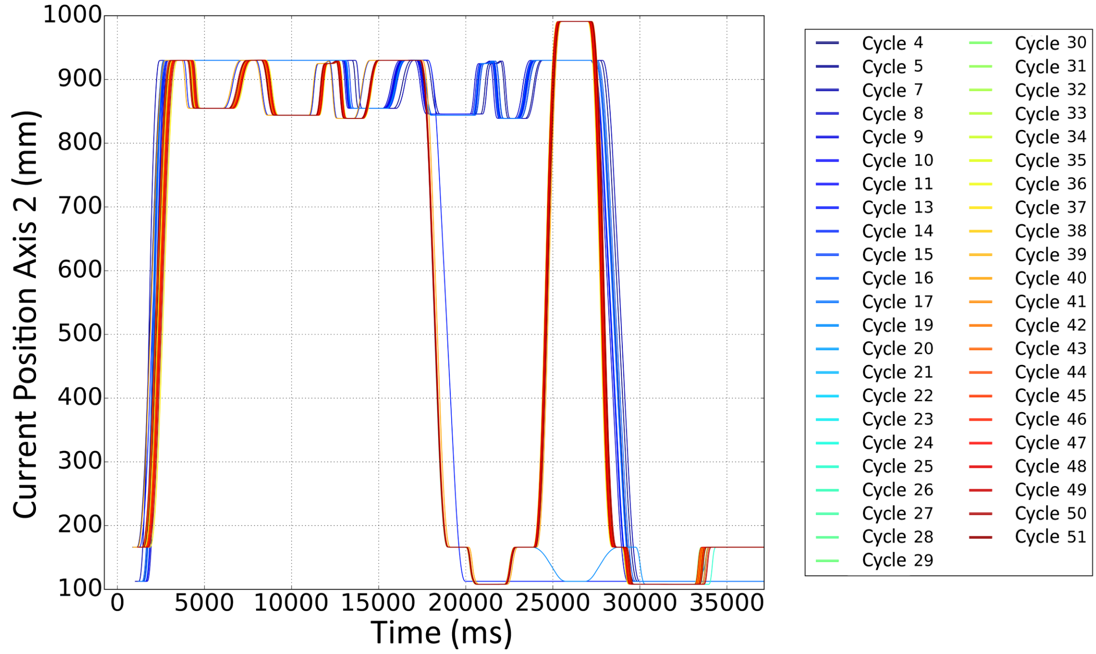


Figure 5.28: *Scenario 3: A cycle in reference to many cycles*

Quantile calculation is applied to determine the deviation. An area containing a certain part of training data can be confined by two quantiles of training data. To examine if data behave like a certain part of training data (*perc* in percent), two quantiles ($q_{1/2}$) are determined by the formula below:

$$q_{1/2} = 0,5 \pm (0,5 \cdot perc)$$

If $perc = 100\%$ is valid, 100% of training data are confined by the quantile (conform to maximum and minimum). A 50% quantile is calculated for $perc = 0\%$ according to the median of training data.

TARDIS requirements:

A new module is required to enable the desired functionality for scenario 3 in TARDIS. Figure 5.29 shows the structural environment of the so-called "Quantile Function" module. Two M-Values are used as input for the module, one as "current cycle" and one as "reference cycle". Initially, it is tested at the outset of the function if input data types are the same. It can be set as constraint in which percentage of data the current cycle should be contained. The quantile area between both quantiles is calculated from the percentage. Quantile Function always gives graphical feedback about the result during execution.

Besides that, a return value can be further used as Binary Stream in the rule editor. It defines a binary row of values as M-Value. Thereby `True` corresponds to a value outlying the quantile area at the respective time point. All values inside the quantile area are labeled as `False`.

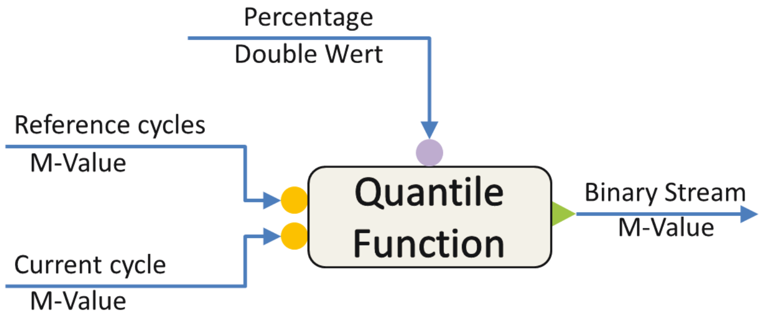


Figure 5.29: *Module Quantile Function*

5.4.3.4 Scenario 4: Visual Control Histogram

Case of application: Human knowledge about the state of the industrial unit to be tested is recorded with help of Visual Control Points. Analyzing several shifts of all VCPs is helpful. To determine the number of "ok" or "not ok" classifications of a certain part of the unit, the graphical function of the histogram is employed (see Figure 5.30 (right)). An expert can easily identify the location of frequent problems in the VCPs using the added values. Likewise the VCPs with error-free classification can easily be identified.

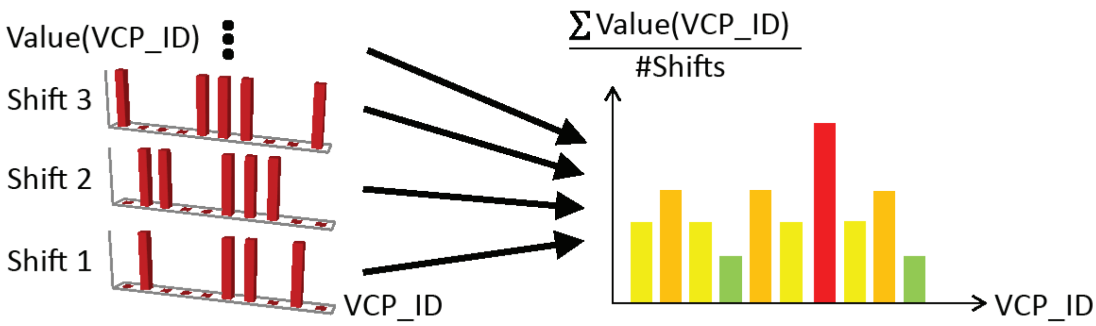


Figure 5.30: *Scenario 4: Visual Control Histogram*

TARDIS requirements:
The Plot Function, introduced in Chapter 5.4.1.2, is expanded by a selection of "Histogram" to attend to scenario 4 (see Figure 5.31). The function is used to identify the one VCP among several, that very often or very seldom scores certain values. With help of this function, it is for instance possible to examine, if a particular machine part is erroneous, because is causes problems in many work shifts. In reverse, the correct function of machine parts or units can be ensured with the

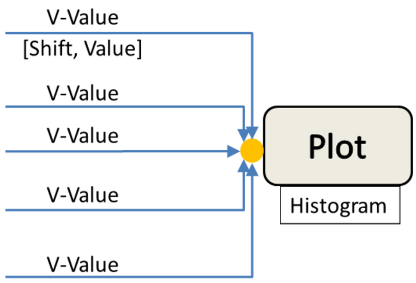


Figure 5.31: *Module "Plot" with selection Histogram*

VCPs being labeled as error-free. The critically outstanding VCPs are easily visible in graphic display. An optional number of different V-Values is put into the Plot Function, and can be compared to each other.

5.4.3.5 Scenario 5: Correlation

Case of application: A correlation matrix can be generated for several variables or trace correlations of different data. On the one hand, it allows detection of any direct influence of a variable on another. On the other hand, plausibility tests can be run to confirm the recorded data's accordance to physical laws. This can for example be used for the correlation of position and speed of a robot's axis.

TARDIS requirements: A further selection slot "correlation" is inserted into the plot module to make the correlation matrix available in the rule editor. Moreover, a discretionary number of input signals can be connected to the plot. A line and column are added to the correlation matrix for each signal, and correlation to other variables is determined.

5.4.3.6 Scenario 6: Complex cycle comparison

Case of application: The other scenarios have provided us with classification methods, that can be used for one heuristic (like "does the current exceed this threshold?"). The aim of this scenario is to run several tests on different criteria simultaneously and to plot the result. In addition to classification, also the quality of comparison between cycles can be determined and analyzed.

TARDIS requirements: Scenario 6 is designed to demonstrate the combination of all functions generated for TARDIS. Figure 5.32 illustrates an abstract situation with an expert trying to detect abnormalities in test cycles using several heuristics. Up to now, the modules Threshold and Quantile Function can be used as heuristics, but TARDIS can easily be expanded to other functions. With selection of Binary Stream as output of the heuristic, binary values are saved for the time points of the test cycles, informing about compliance of the heuristics. Then the binary values are evaluated using the plot function as a histogram. In the latter the number of violated heuristics and corresponding time points can be seen.

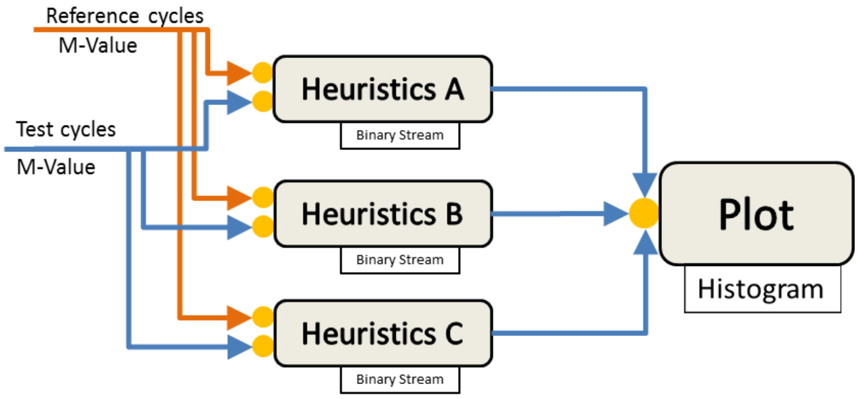


Figure 5.32: *Scenario 6: Complex cycle comparison*

5.4.4 CAT architecture

This section introduces components of the designed framework CAT. At first, the structural composition of the designed software is described. Further, the communication of CAT with TARDIS and the data mining functions are explained. Finally, an interaction model for the use of this software is constructed.

5.4.4.1 Architecture Survey

Two existing software frameworks are expanded and adapted for the Complex Analyzing Tool (CAT). In Figure 5.33, the concept visualization of the architecture can be seen. The AMS framework [15] includes components about communication, database administration, and compilation of graphic user surfaces. The same framework has been used for the final implementation of scenario 2 (Visual Control), scenario 5 (Machinery Optimizer) and scenario 6 (Teleoperation). All data from the Condition Monitoring system and the inspection experiments are stored within this framework.

With the rule editor, TARDIS [168] offers a graphic surface to establish various rules and executes these via the Rule Solver. The following chapter deals with architectural components relating to the used programming languages and interfaces.

Preparations for data visualization with the help of Data-Miners are depicted in Chapter 5.4.4.4. The user's interaction are explained in Section 5.4.4.5 through all working steps.

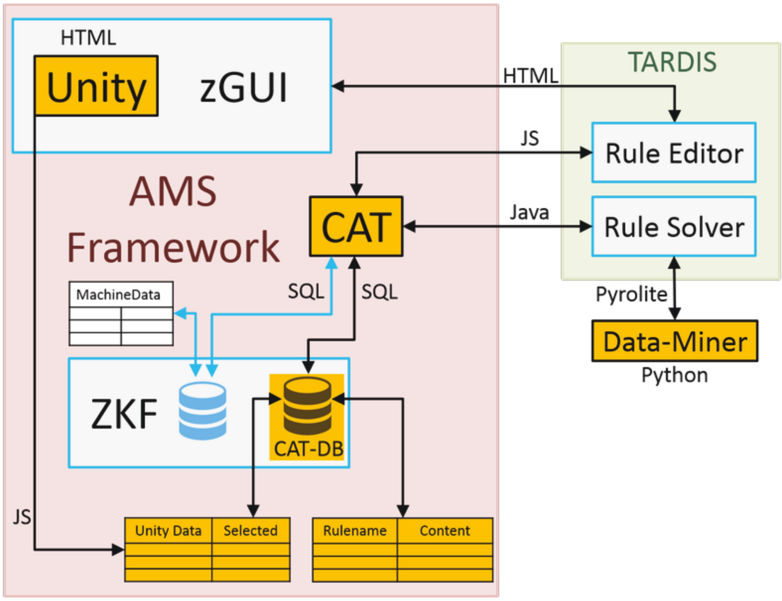


Figure 5.33: *Architecture software extensions*

5.4.4.2 Architecture interfaces

Figure 5.33 shows the interfaces connecting the different software components. Since the AMS framework GUI (zGUI) is completely based on javascript, the CAT software is also implemented in javascript. The communication framework "ZKF" [170] facilitates communication between the components within the AMS framework. The databases are operated by SQL. In zGUI there are web-based user surfaces that can communicate with components of the AMS framework. Besides implementation of HTML sites, javascript and CSS files can be linked in zGUI. Both Unity implementation as well as the rule editor are linked to the zGUI as an HTML site. Data chosen in virtual reality is sent to a local database (CAT-DB) by Unity via javascript.

The "CAT service" functions as a central data interchange point from AMS framework to TARDIS. Rules generated in the rule editor are saved in the CAT-database through a javascript command to the CAT service. Since the Rule Solver is started as an independent javascript process, a Interprocess Communication is used on base of sockets. The Pyrolite Library implements the Python-based data mining algorithms [171]. It enables the communication between a Python server to a javascript class.

The orange colouring in Figure 5.33 highlights the extensions included within the frame of this work.

5.4.4.3 Integration of TARDIS

When combining TARDIS with this work, an emphasis has been set on the communication via clearly defined interfaces and a preferably modularized work, because TARDIS is also used for other applications. The designed groups of scenarios (Section 5.4.3) are integrated in the rule editor, and further rules can be applied. The rule editor is integrated as an HTML file into the AMS framework and communicates with the CAT service via javascript. The rule solver's data exchange with the CAT service is executed by Interprocess Communication between the two javascript processes.

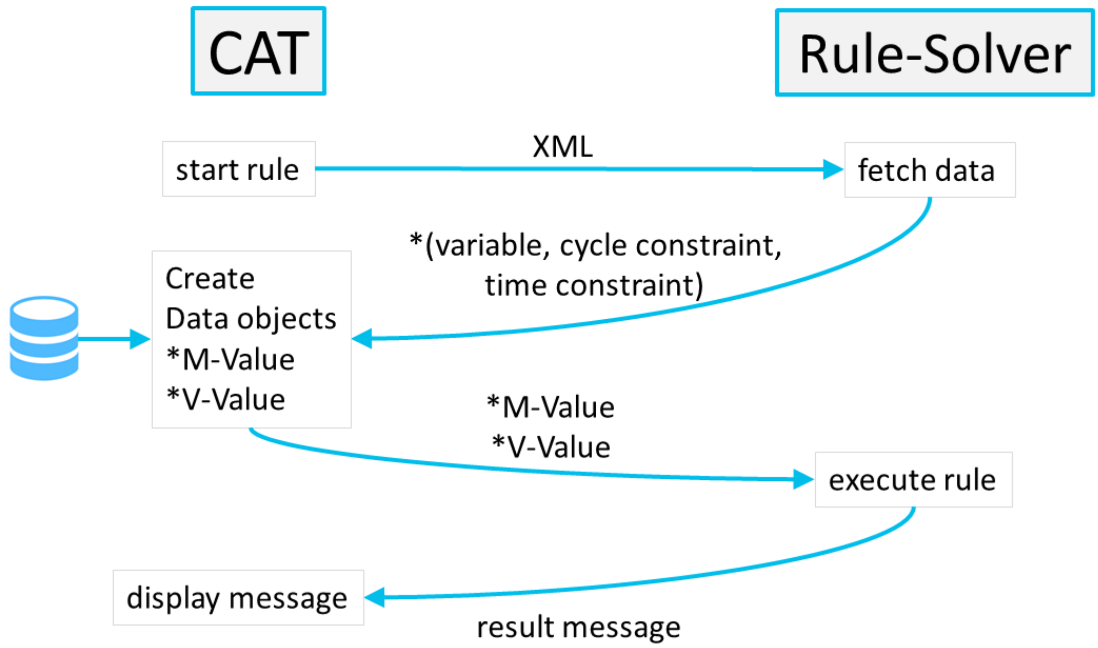


Figure 5.34: *Interaction of the rule solver with CAT software*

Communication between CAT and the rule solver is depicted in Figure 5.34. First, a rule designed in the rule editor is started. Then the rule solver prompts correspondent data from the database of CAT. Data is defined by variable name, cycle constraints, and time constraints. Finally, the rule can be executed including data, a result is returned as a message to CAT, and accordingly a plot can be generated by the Data-Miner.

To translate the application scenarios from Section 5.4.3 to TARDIS, specific modules are designed, which enhance the functionality of TARDIS. These modules partially require structural alterations of the rule editor: Modules regularly have a fixed number of input and output signals. However, newly added modules need a flexible number of input signals. Besides the regular input and output signals, so-called "Constraints" determine auxiliary conditions for some of the rule editor's functions. Constraints are managed as a new signal of a module and marked with a violet circle on the upper side of a module. In case of a constraint not being chosen, the function automatically uses a standard value. Further, some modules require a flexible type of output signal, because the subsequent use strongly depends on the module's output. This is put into effect by a drop-down menu within the respective modules (see Section 5.4.3).

5.4.4.4 Integration Data-Miner

The "Data-Miner" module designates a Python server connected with TARDIS via the Pyrolite interface. The library `matplotlib` is used for the visualization of translated data [172]. Data structures are managed with the help of the `pandas}` library [173]. So-called `DataFrames` arrange data in an order of lines (Index) and columns (Columns), similar to tables. Single columns from `DataFrames` can be handled as `Series`, which respectively assign a value to an index.

Robot data (M-Data compare Section 5.4.2.1) are codified from the database via cycle number to a tree map of time points with values.

Python uses the data as dictionary, where a number of time points with values is assigned to a cycle number and then it either can iterate by the entries or convert the whole dictionary in a `DataFrame`.

Data of the visual control points (V-Data compare Section 5.4.2.2) is managed as a classification of layer number (`Long`) to a binary value (`Boolean`) from the database. In the process, data is assigned per `String` to the name of the corresponding VCP. In Python, data is received as dictionary.

If data is transferred directly from javascript in a `DataFrame`, for every index (in case of the robot a measuring time point) a line is added to the `DataFrame`. With the intervals between measurements not being equidistant, a new index is added for every previously unknown time point in the `DataFrame`. Values for the other cycles rarely exist for the new index, therefore `pandas` backfills missing values as "Not a Number" (`NaN`) in the `DataFrame`. Single `Series` have own indices for the values. Regarding the graphic display of data from the database, this means that every column is managed as `pandas.Series` to avoid falsified display by `NaNs`.

The AMS framework already manages to embed a graphic dynamically in the website. Yet, to enable the functionality of shifting and magnifying the graphic, a new window showing the graphs generated by Python is opened. The scenarios in Section 5.4.5 depict implementation of the graphs in detail.

5.4.4.5 Interaction model

In this study, a new perspective for the AMS framework is established and utilized as a frame for the designed software parts (see Figure 5.35). Figure 5.36 shows the design of the user interface in a wireframe [138]. For the interface, various arrangements of the components were compared in several wireframes. By means of the wireframes, thoughts about user interaction and communication between the components were discussed.

The designed interface has four components: VR implementation with Unity (upper left), the rule editor from TARDIS (right), a favorite list for rules, as well as a description field for system feedback.

Choosing the first component for interaction is essentially the user's decision. There are three starting points provided, depending on demand:

- **Unity:** The robot's VR model is used to select parts of the robot and to transfer correspondent data to the rule editor. For this purpose, the user

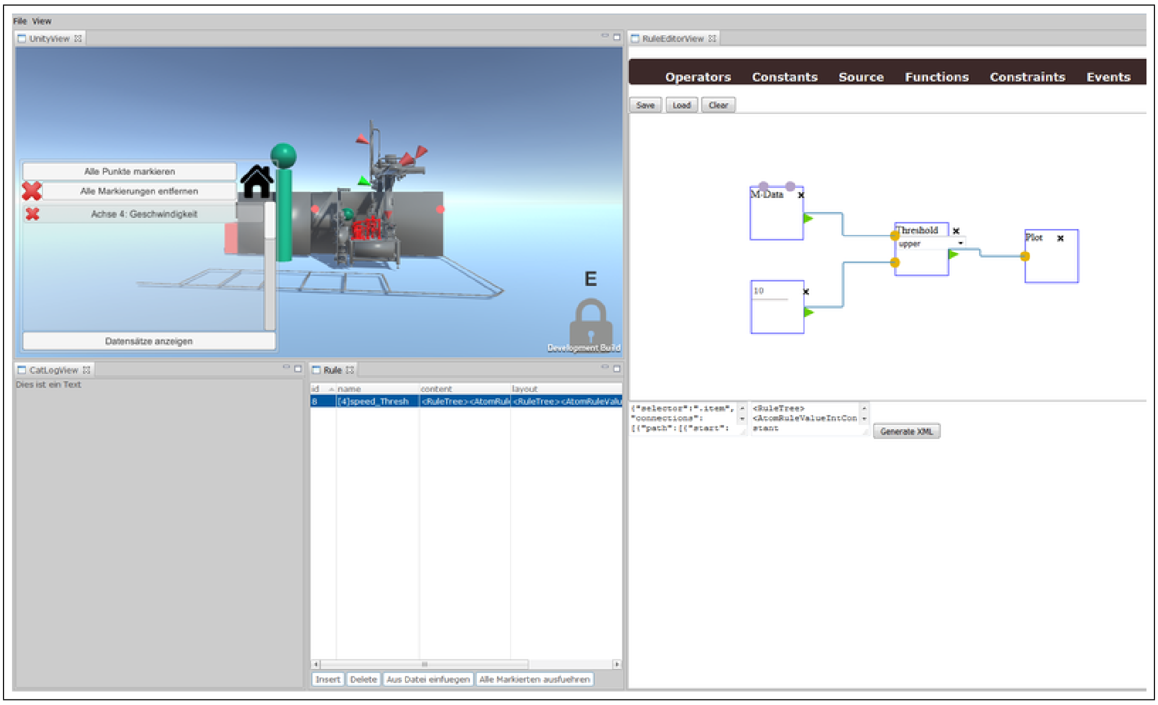
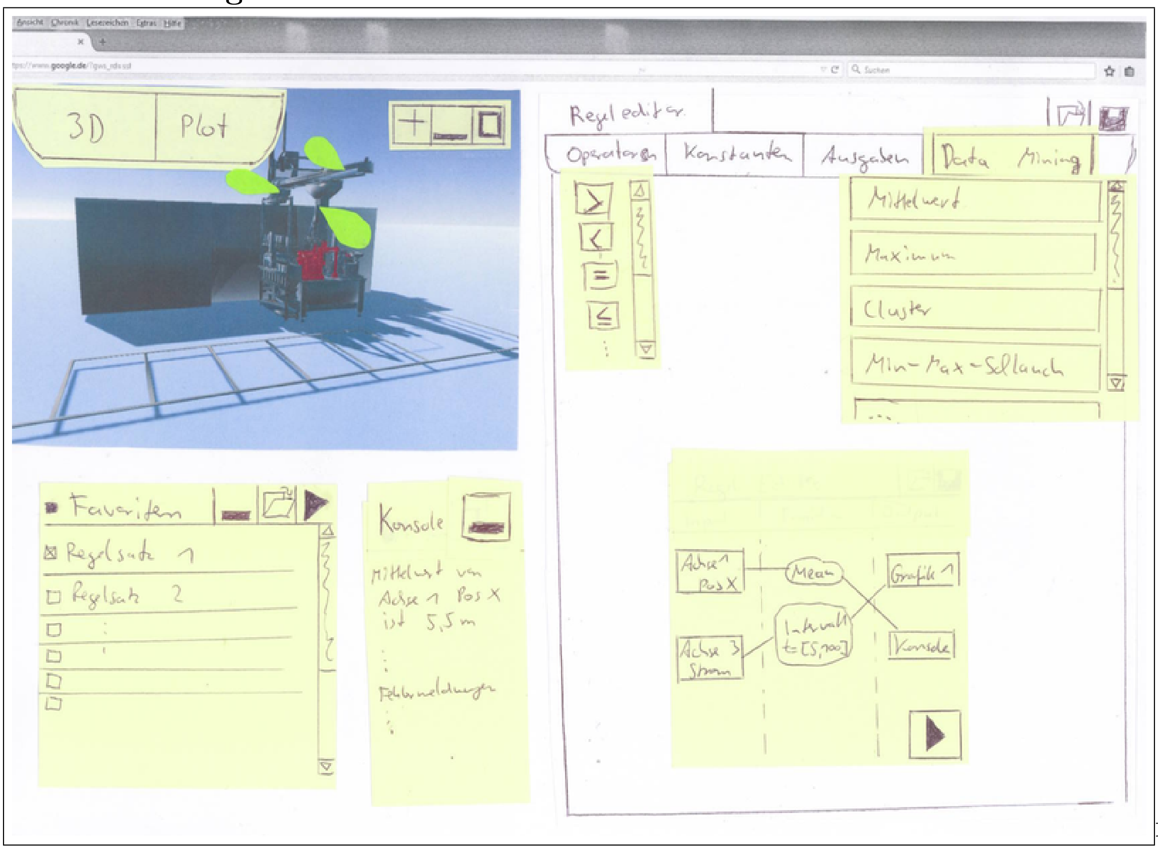


Figure 5.35: *CAT interface in AMS web environment*



f

Figure 5.36: *Wireframes for CAT perspective*

interacts with the interaction points designed as cones, as depicted in Figure 5.37.

- **Rule editor:** The user compiles various data sources in the rule editor and uses functions to analyze these. Moreover, generated rules can be saved, thereby directly forwarding them to the favorite list. Execution of the rules is operated by the button "run all tagged" in the favorite list. So several rules can be run consecutively.
- **Favourites:** Since the user worked with the software beforehand, a rule is existent in a file on the local storage. He imports it via the button "Paste from file" and intends to directly run it.

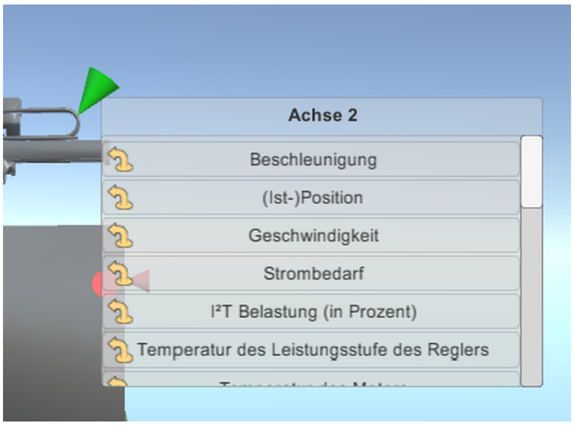


Figure 5.37: *Left click interaction with an interaction point in Unity*

If saving and loading of rules remains out of consideration, an interaction model is designed according to Figure 5.38. The 360° arrows symbolize the possibility of multiple execution of the respective action. For instance, a user-defined amount of data can be selected in Unity. For the selected data, correspondent modules are inserted automatically in the rule editor and can be processed at will via functions. As a result, a graphic plot or an output into the description field (Log) is returned respectively, according to the rule.

5.4.5 Data mining results

The following section describes implementation of the Virtual Reality with the help of Unity software in detail. Afterwards, the results from the specific use cases for the CAT software are illustrated.

5.4.5.1 Design und Umsetzung in Unity

Getting across a full comprehension of the situation to the user and ensuring an easy employment of the interface is important for the implementation of the robot unit in VR. Figure 5.39 shows the user interface in Unity.

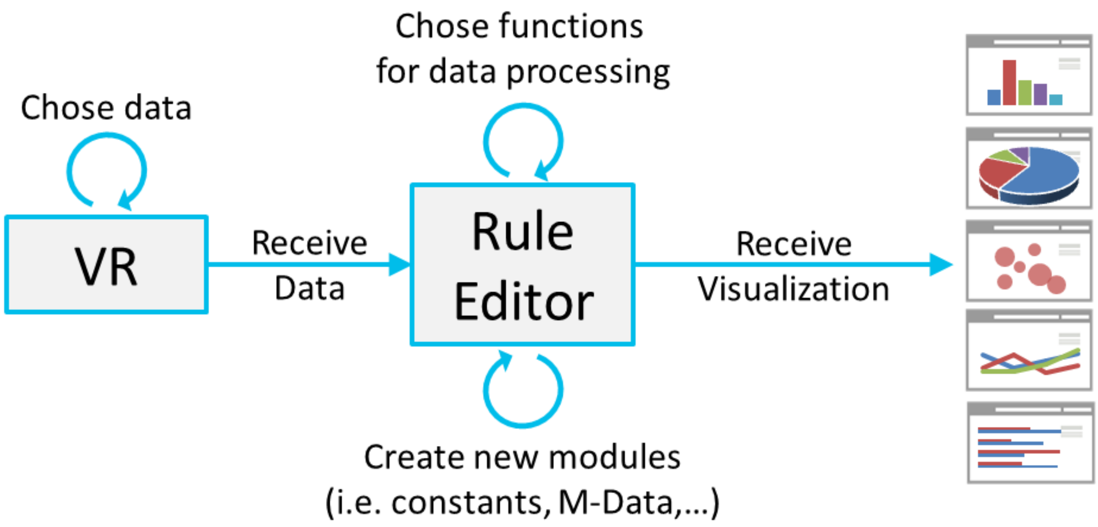


Figure 5.38: *Interaction model*

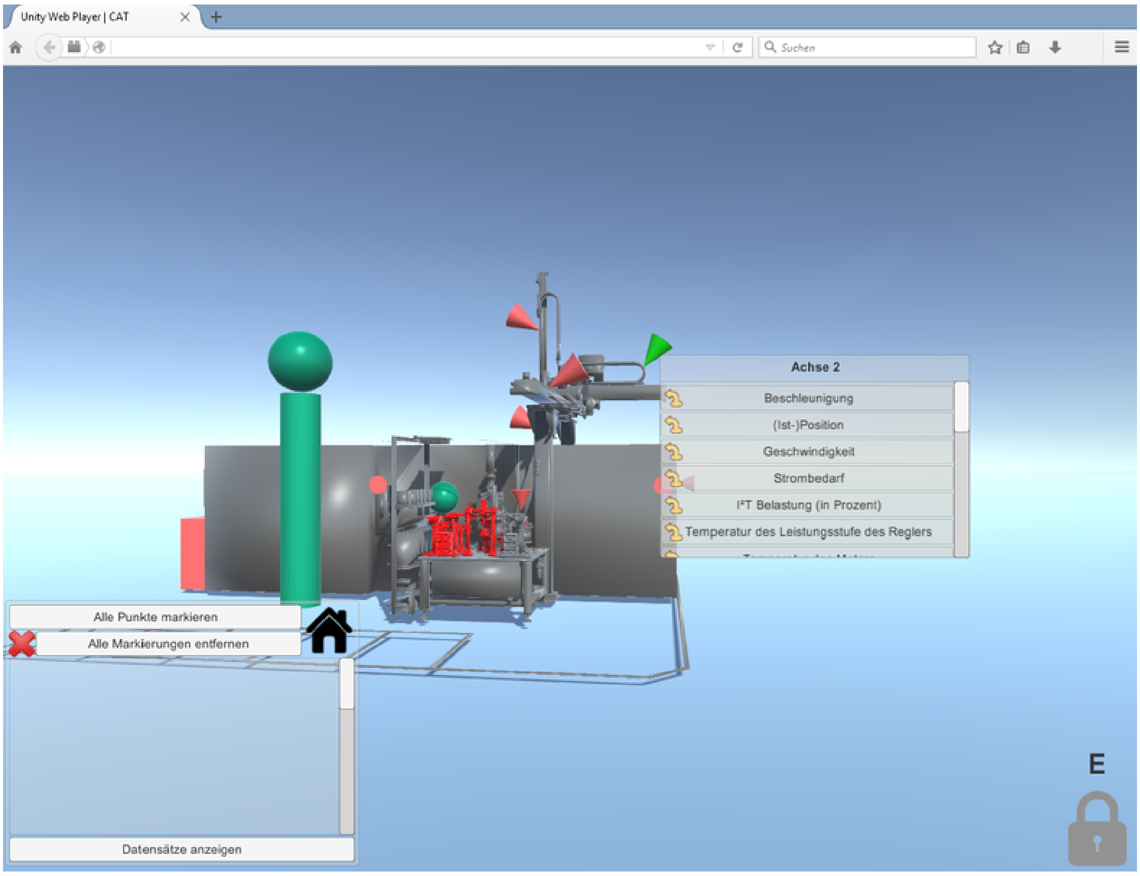


Figure 5.39: *Interactive Unity environment*

To give the user a good overview over the unit, controls are implemented in first-person-perspective. An expert can move his character around the unit with mouse and keyboard. The character's camera trails the cursor movements. To facilitate controls for the user, there are built-in comfort functions: The small black house in the lower menu on the left repositions the character on the starting position, and by pressing "E", the camera movement is blocked to move the cursor freely (see Figure 5.39 lower right).

The green spheres symbolize permanently installed cameras. The pedestals beneath were designed from conceptual considerations to illustrate the cameras' spacial position. The user can shift via mouse click to another camera perspective to observe the unit from another angle. The cameras' positions are correspondent to those of the real cameras that are placed in the factory of P&G.

Available interaction points on the unit are illustrated by red cones. As soon as the cursor moves across them, interactive objects appear in yellow, to indicate the selected part of the unit. The user can interact with objects by left clicking (cone turns green). Next to the unit, there is a red cuboid on the left side(symbolizing the robot control) with data that cannot be assigned directly to a part of the unit, for example the system time. In addition to the robot's four axes and the robot control, further cones are shown for the Visual Control Points. These are treated like the other objects. By left-clicking a menu appears for the interaction points (selection menu), including data labels of the selected part of the unit. Left of the labels, there is an arrow for transferring to the permanently visible menu on the left (data menu).

All selected data is listed in the data menu and can be sent as javascript command to the AMS framework via the button "show data sets". For each entry, the variable name and a readable name is transferred. In the next step, a module is generated for each entry in the rule editor. Here, the readable name is shown for better usability, while internally data is assigned according to the variable names.

5.4.5.2 Configuration of objects

To label the objects and the connected context-menu, a configuration file `UnityConfig_MTR.xml` is imported. The structure of the XML file is defined by a DTD file, as shown in Figure 5.40. `object` thereby indicates an object in Unity, which should include data labels. `object` contains the readable object name in `value`, whereas the attributes `type` and `name` refer to the type and the object name used by Unity. A `object` element must at least have one entry of `data`. The latter only contains the three specifying attributes. The `data` elements are separately assigned to the `object` in the `UnityConfig_MTR`, since for instance all axes have the same values in `data` and thereby doublings in the XML file can be avoided. Using the attributes from `data` within the `object` elements, an assignment of `data` elements to each `object` is possible. Display in Unity is extracted from a `value` element of a `dataEntry`. For Visual Control Points, the attribute `environment` and the element `cil_type` are introduced. With some VCPs being specified by the same number, `environment` is used for the assignment of VCPs. As a preparation towards more detailed information about the objects, `cil_type` and `description` are installed in Unity.

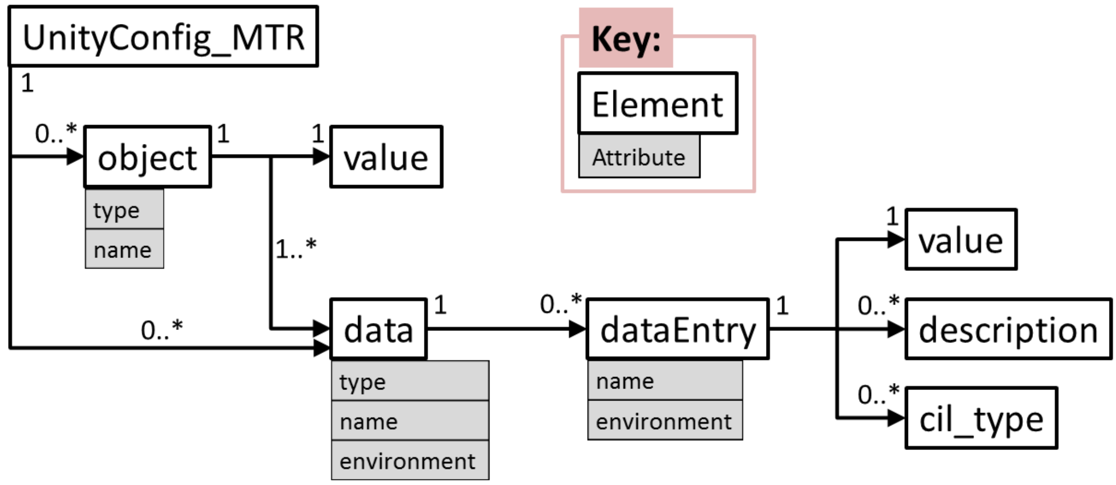


Figure 5.40: *Visualization of the UnityConfig_MTR.dtd*

Reading of the XML file is implemented, because Unity cannot be changed in the HTML file. Thereby the flexible labeling of objects in the XML file is possible. Naming the `name` attribute of `object` identical with the objects from the Unity editor is important to provide consistency. Likewise, the `name` attribute of each `dataEntry` element has to correspond to the exact name of the variable in the AMS database. To assign the VCPs, the `environment` has to be considered accessory to the `name`.

Unity has a safety model for the import of external data to prevent wiretapping of foreign data over a unity application. Since the XML file is placed on the local server of our application, the safety model does not block import. Access from the XML file to an external DTD file though is not allowed just like that. The XML file is not validated during import by the DTD to avoid the constraint of removing the link to the DTD file from the XML file.

Up until version .NET 3.5 it is only possible to either permit or completely prohibit a DTD comparison, in .NET>3.5 one can ignore the DTD.

5.4.5.3 Scenario 1: Validation of a value

So-called breakpoints can be set with the Machinery Optimizer software (cf. Section 5.3). These identify time points within the robot cycle and can be determined with the robot's implementation code.

Validation of I^2T load: The expert defines time point $t = 10s$ in the code using a robot action. At this time point the experts wants to test, if the I^2T strain lies above the threshold of 20% of the acceptable constant strain (see Figure 5.41).

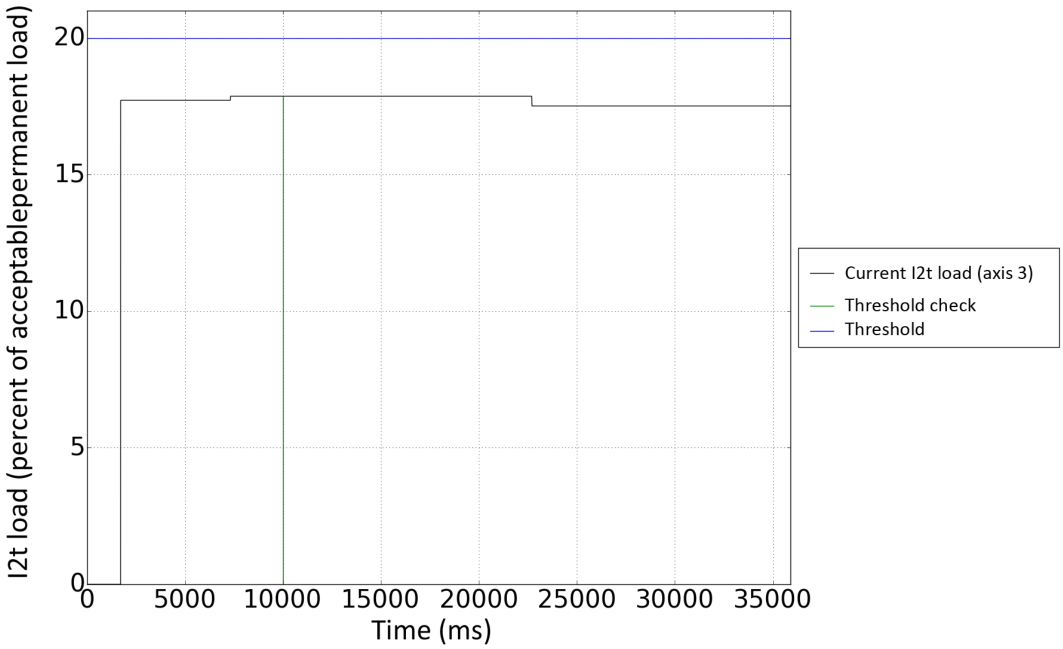


Figure 5.41: *I^2T load (axis 3) of cycle 35*

Returning a `False`, the expert is informed via Log that the threshold is not exceeded at time point 10 sec.

Validation of motor temperature: Alongside to breakpoints, the time point to test threshold values can also be determined by binary variables. Variable _IBIN_IN[3] indicates if the profiles of the injection molding machine are open. Transition from open to closed is determined at time point $t = 14,5s$. The expert tests at this time point if the motor temperature of axis 1 exceeds 30°C and receives the message `True` in the Log as a result. Contrary to the previous example, the threshold is not adhered to. Figure 5.42 exemplifies the test.

5.4.5.4 Scenario 2: Validation of a cycle

Estimation of current peaks: An expert wants to take a threshold test for the whole process of cycle 35 in the motor current of axis 3 (cf. Figure 5.43). The threshold of 6A to be tested is an empirical value of the expert and returns Figure 5.44 as a result. Using the `Threshold` function in TARDIS, different return values can be employed for further modules. Thereby, the expert can for example illustrate only the values above the threshold, to filter out irrelevant values as shown in Figure 5.45.

Separating warm-up and regular operation: To closer examine the course of position of axis 2, an expert wants to separate the initial cycles from the remaining

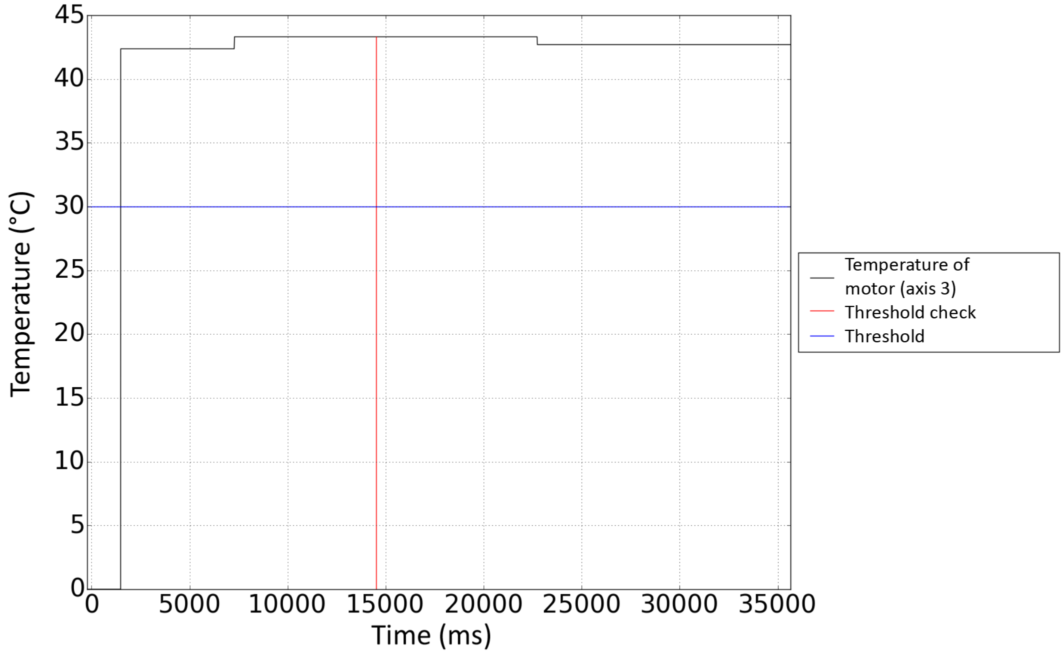


Figure 5.42: *Temperature of the motor in cycle 35*

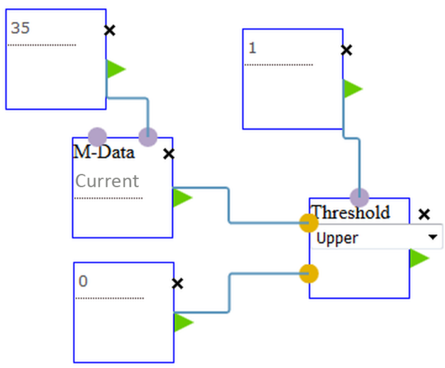


Figure 5.43: *Example rule for testing on current peaks displayed in the rule editor*

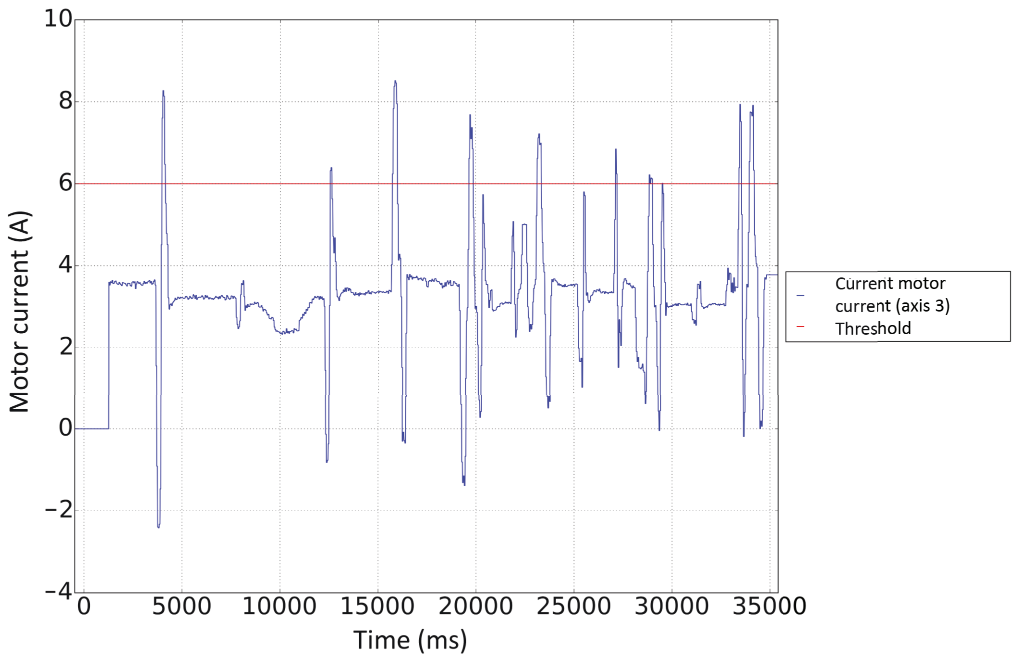


Figure 5.44: *Current trend of motor of axis 3*

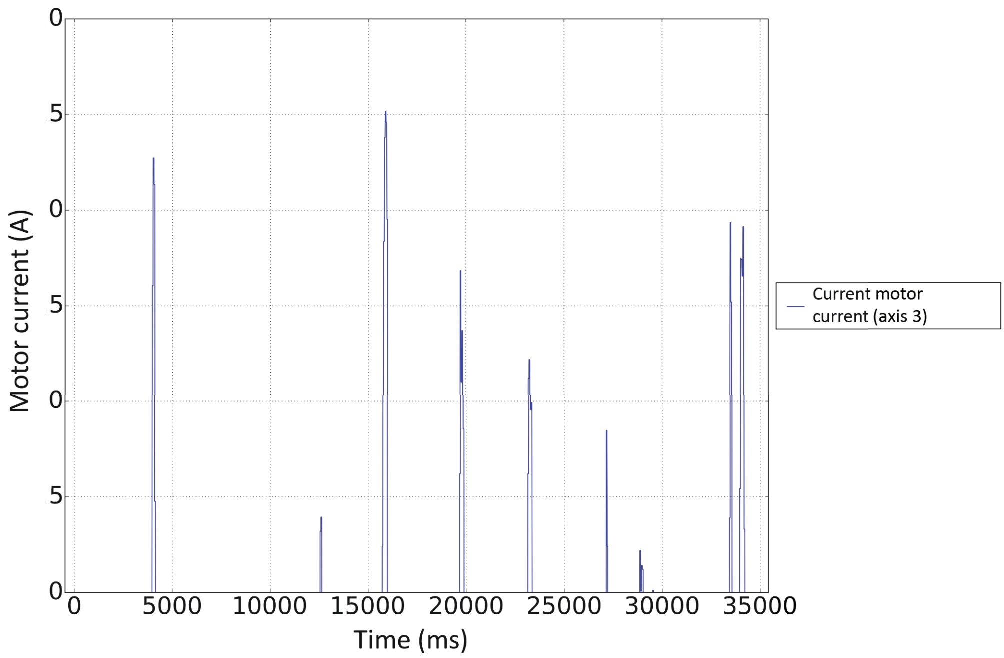


Figure 5.45: *Motor current of axis 3 with threshold 6 Ampere*

cycles from a dataset. He applies the Least Square Fit function to gain a heuristic stating which cycles do not fit into the normal process (see Figure 5.46). For each cycle the sum of the quadratic errors to the reference cycle (here cycle 35) is identified by the LSF. On base of the display of quadratic errors (Figure 5.47), the experts chooses a threshold (here: $5{\cdot}10^8$mm^2). Above it cycles are not classified as operating phase.

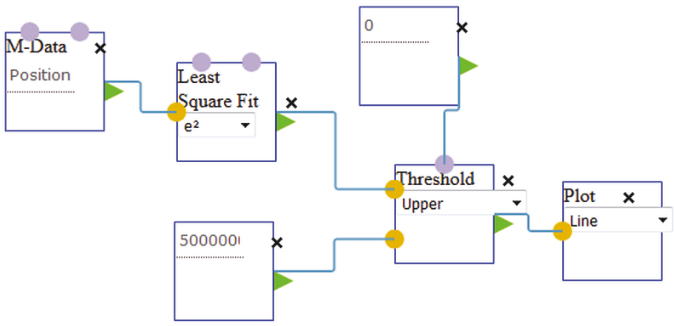


Figure 5.46: *Separating warm-up and regular operation in rule editor*

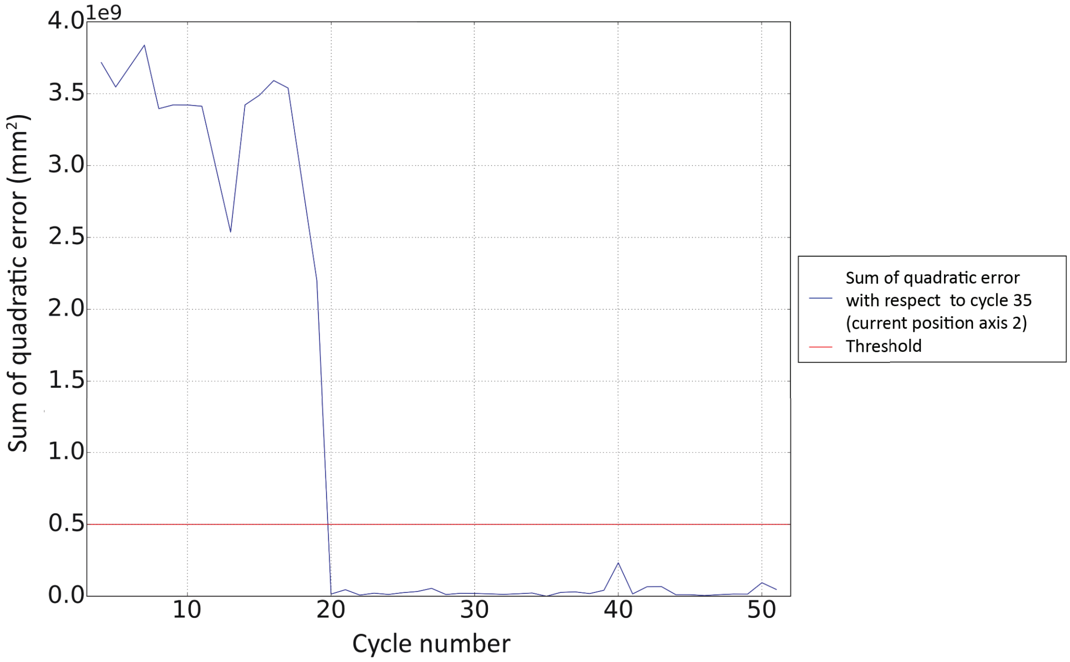


Figure 5.47: *Sum of quadratic error for cycle 35 axis 2 position*

5.4.5.5 Scenario 3: A cycle in reference to many cycles

Validation of acceleration (Axis 4) of cycle 40: While separating initial and operating phases, the expert notices cycle 40 to show a high deviation at the sum

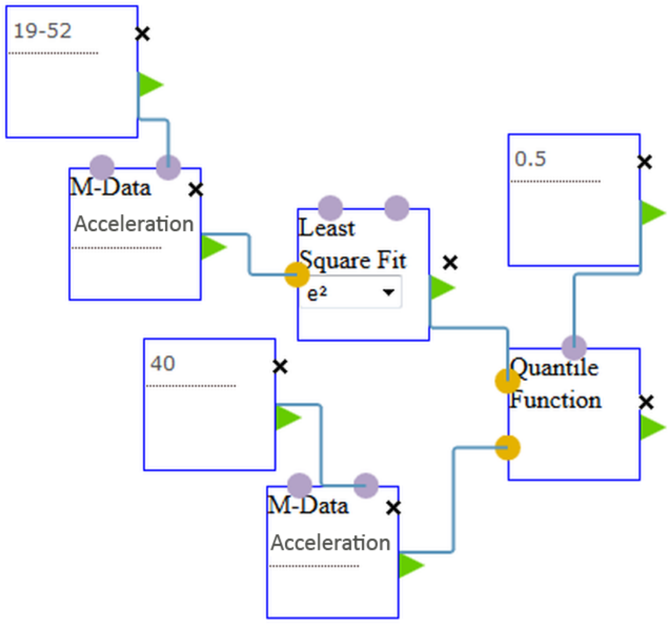


Figure 5.48: *Validation of acceleration (Axis 4) of cycle 40 in rule editor*

of quadratic errors when compared to cycle 35 (Figure 5.47). First the experts decides to use cycles 19 to 52 as training data. That means that the training data is classified as "following the usual process" (or "O.K."). The expert compares the training data with cycle 40 to accelerate axis 4 using Quantile Function from the rule editor (cf. Figure 5.48).

At first, the level of deviation is tested for a share of 50% of all training data (Figure 5.49). A part of Figure 5.49 is magnified in Figure 5.50 for better comprehension of the generated graphic. The images show training data (reference cycles) in light grey and the consequential quantile area in dark grey. The course of cycle 40 is coloured in black, as long as it is inside the quantile area. Values lying outside are indicated in red. In Figure 5.49, the reason for cycle deviation can barely be seen, because the bigger part of cycle 40 does not correspond to the quantile area, thus colouring a wide area in red.

The expert decides to test cycle 40 for 100% of all training data (Figure 5.51). Especially the deviations in the two strong accelerations at approximately 3 and 17 seconds strike at this point. In the first acceleration, deviations are on the left, whereas they are on the right side for the second acceleration. That implies that the first acceleration of cycle 40 starts earlier, and the second later.

Temperature area for controller power stage: Additionally to cycle testing, Quantile Function can also be used to identify the data range. Therefore identifying the maximal and minimal range is beneficial for hardly comparable data, such as temperature courses.

Figure 5.52 illustrates the temperature range of the modulator's power stage on axis 4 for all 52 cycles. Minimal values are displayed as blue lines, maximal values

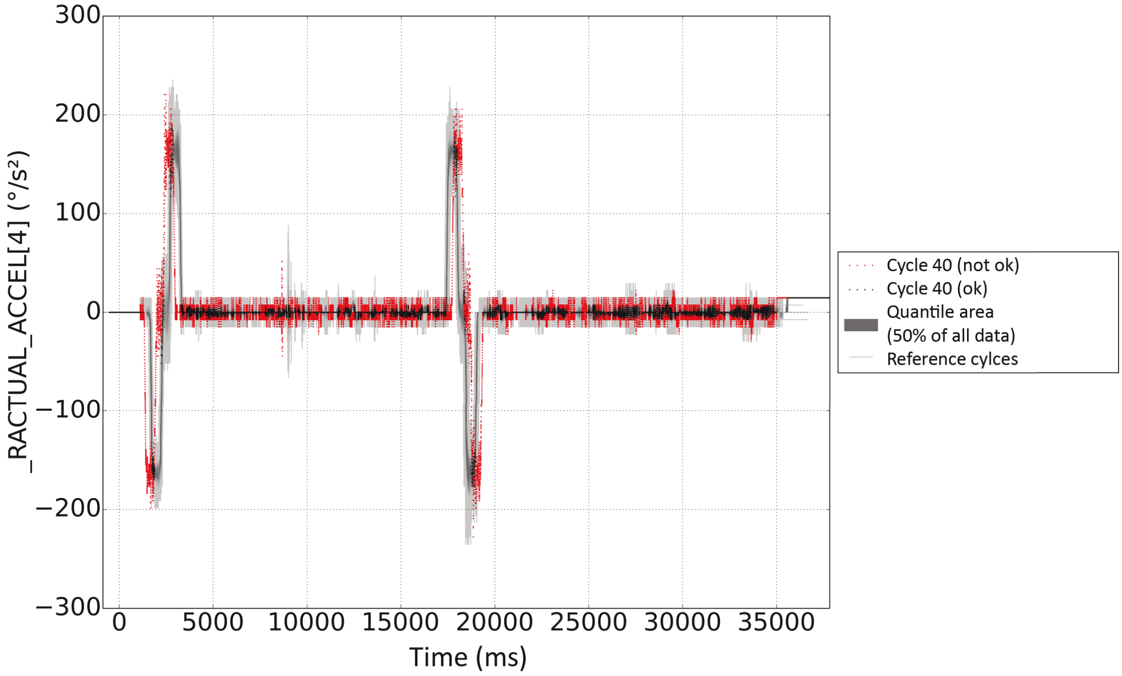


Figure 5.49: *Acceleration of axis 4 (50% Quantile)*

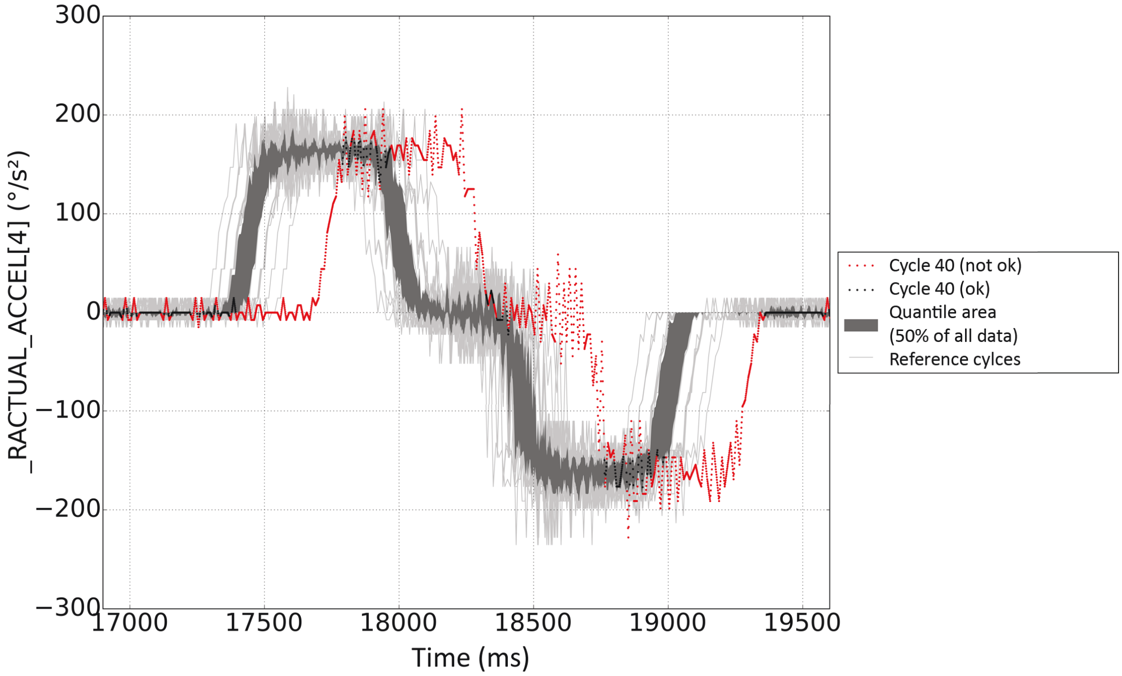


Figure 5.50: *Acceleration of axis 4 (50% Quantile, zoomed)*

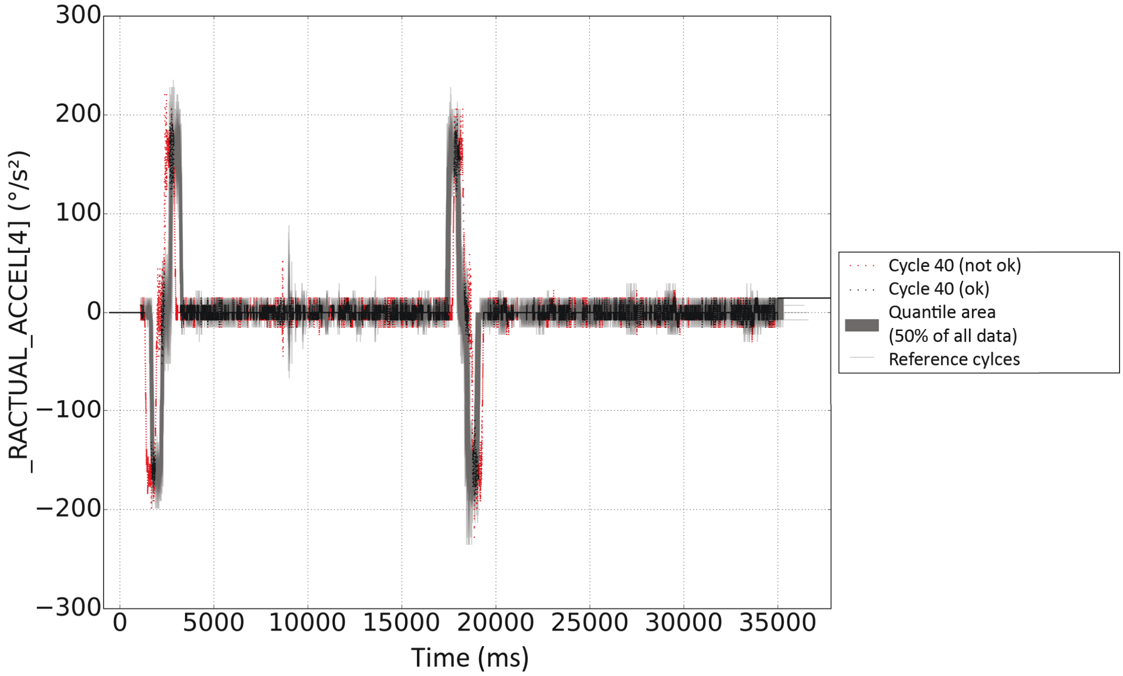


Figure 5.51: *Acceleration of axis 4 (100% Quantile)*

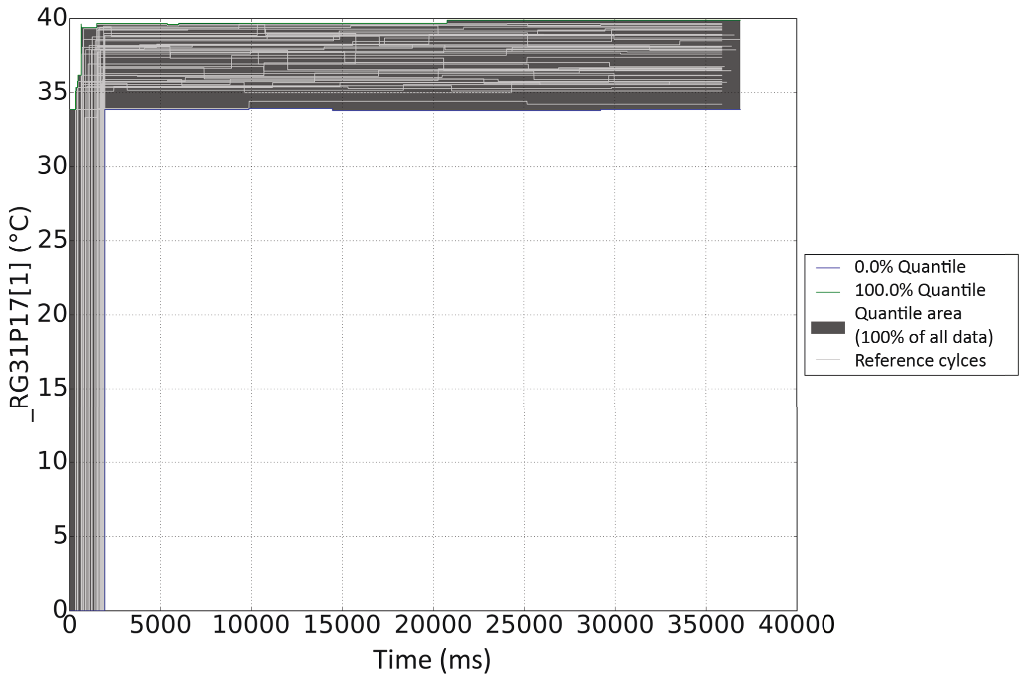


Figure 5.52: *Temperature area for controller power stage*

in green. Contrary to the previous requirements, reference cycles here are drawn in the foreground. This helps to distinguish data referring to the quantile area.

TARDIS implementation M-Data's structure allows to transfer both "reference cycles" as well as "current cycle" with optional constraints to the Quantile Function. In general, M-Value indicates any number of cycles which are labeled with `[timestamp, value]` pairs. This improves the handling of rule editing, rather than transferring each individual reference cycle to Quantile Function.

Functionality of Quantile Function is not limited by time points or cycle number, meaning that any cycle number can be indicated. An expert can thus for instance specify various cycles as current and test these for a reference cycle. This would presumably lead to many deviations, but is the user's decision. Likewise, input parameters can be chosen for single time points. This makes sense when only certain time spans are being observed with corresponding data being tested.

The result of Quantile Function generally strongly depends on input data. The case of M-Value entries including binary values is not sensible for the Quantile Function.

Python Implementation To gain a reasonable comparison between reference cycles and the current cycle, Quantile Function has to run a Least Square Fit prior to all calculations, like in data preprocessing (see Section 5.4.2.4). For the "current cycle" (or an optional number of these), temporal deviation from one of the reference cycles is determined by LSF.

5.4.5.6 Scenario 4: Visual Control

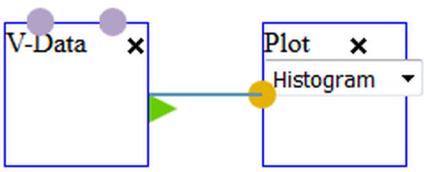


Figure 5.53: *Rule example for histograms of all VCP data in rule editor*

To analyze the technicians' rating of each Visual Control Point in several layers using all VCPs, a histogram is generated over several shifts. A colour spectrum is used for chromatic visualization of evaluation, from red (VCP in all shifts "not O.K."), through yellow to green (VCP in all shifts "O.K."). Figure 5.54 shows the result generated in the rule editor (Figure 5.53). The result reveals that many VCPs in the SGM area are labeled "O.K." in the 17 tested shifts. Testing points "SGM: 11", "SGM: 12", and "SGM: 13" on the injection molding machine show the most "n.O.K."-labels. Remaining VCPs were marked respectively three times as "n.O.K." by the technicians.

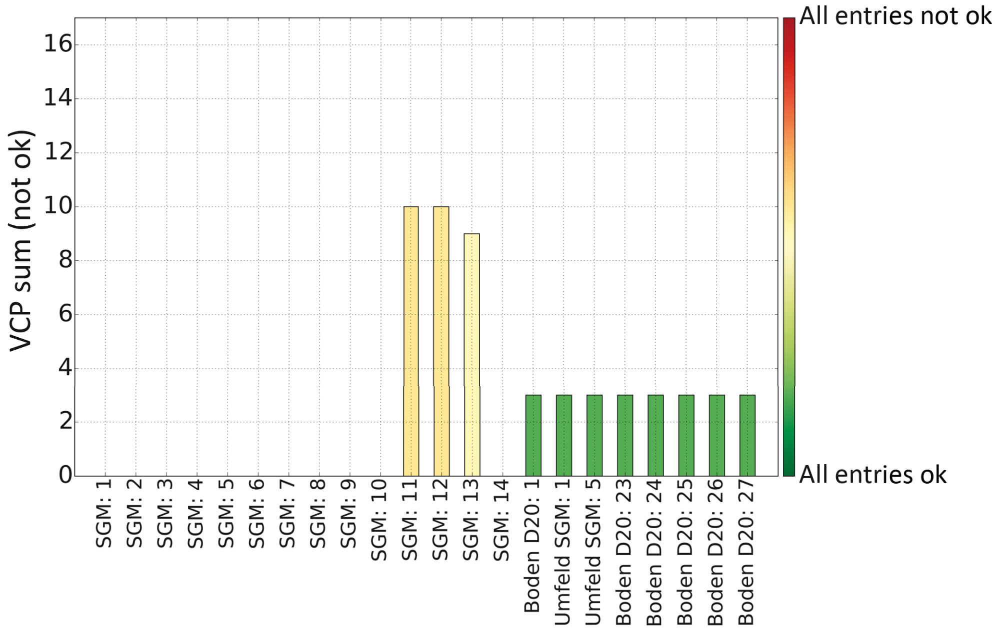


Figure 5.54: *VCP Histogram*

Python implementation

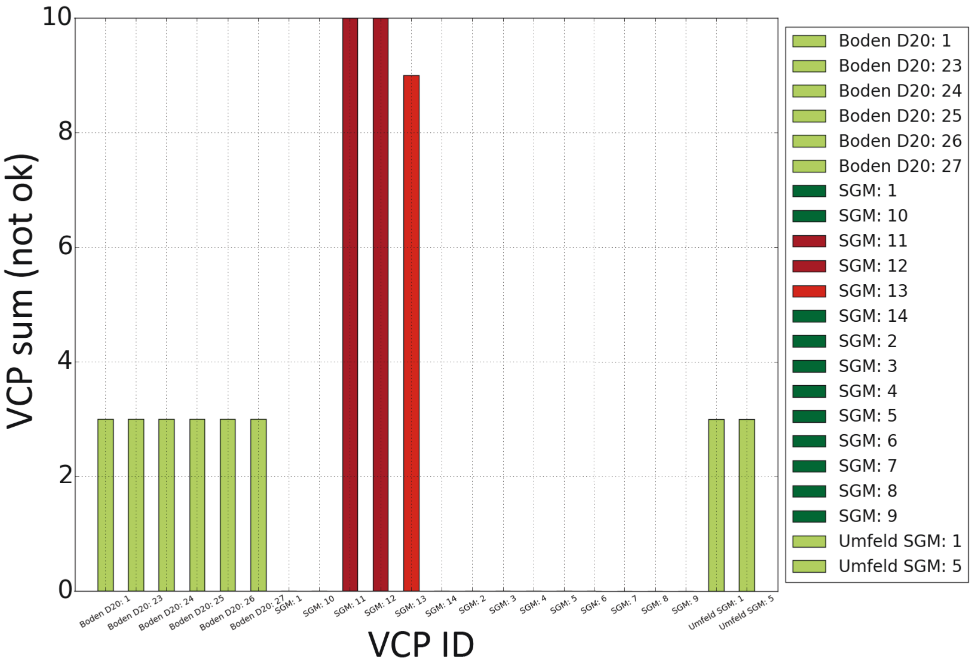


Figure 5.55: *First approach for VCP visualization*

The module V-Value describes binary values of a VCP, assigned to a correspondent working layer. In Python VCPs are managed as an index of a `DataFrames`. The VCP's value in column "Status_sum" is incremented for each "n.O.K." entry. In the result this value is implemented as bar height.

Figure 5.55 shows an early version of visualization. In this version the VCPs are mentioned in the caption as well as in the 45° angle of the abcissa. Apparently the labels on the abcissa appear shifted. `matplotlib` centers captions under the respective point on the abcissa by default setting. Due to the rotation, correlation to the middle part of graphic is almost impractical for the reader. Therefore, a rotation of 90° is selected for labeling the abscissa. The display in Figure 5.55 implies that VCPs SGM: 11 bis 13 are n.O.K. in all layers, since the ordinate scale is not shown up to the full number of layers. Given that this is not true, the ordinate maximum is set to the total number of layers.

5.4.5.7 Scenario 5: Correlation

Application

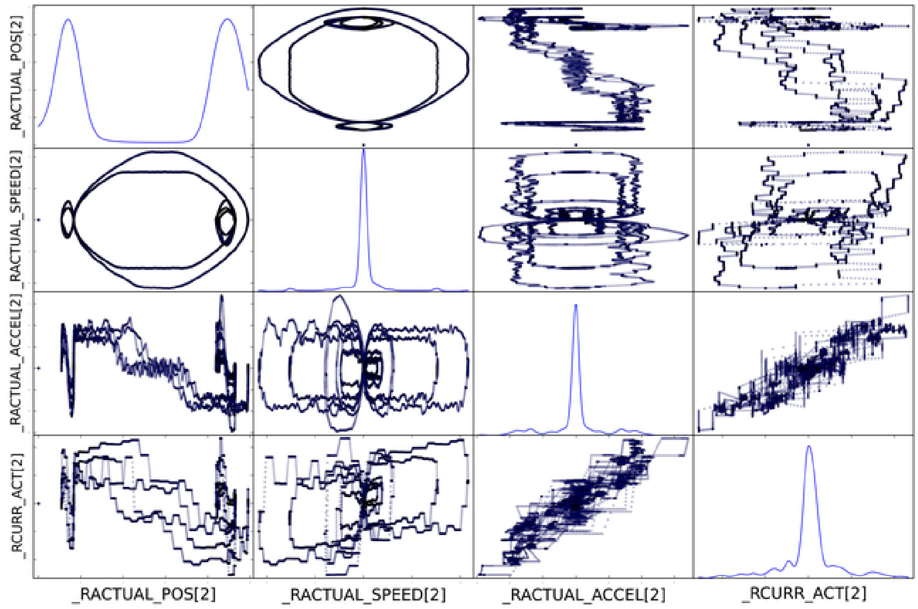


Figure 5.56: *Correlation between position, velocity, acceleration and motor current of axis 2*

To specify correlation of position, speed, acceleration, and motor current of the robot's axis 2, an expert generates a correlation matrix with help of the plot function (see Figure 5.56). Relations between the variables can hardly be seen therein, prompting the expert to examine parts of the matrix separately in the subsequent applications.

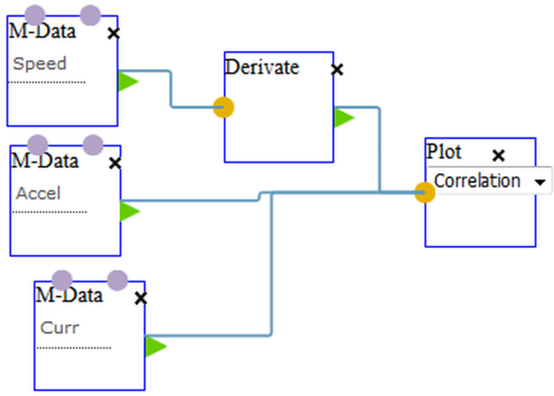


Figure 5.57: *Example rule for correlation of velocity, acceleration and motor current in the rule editor*

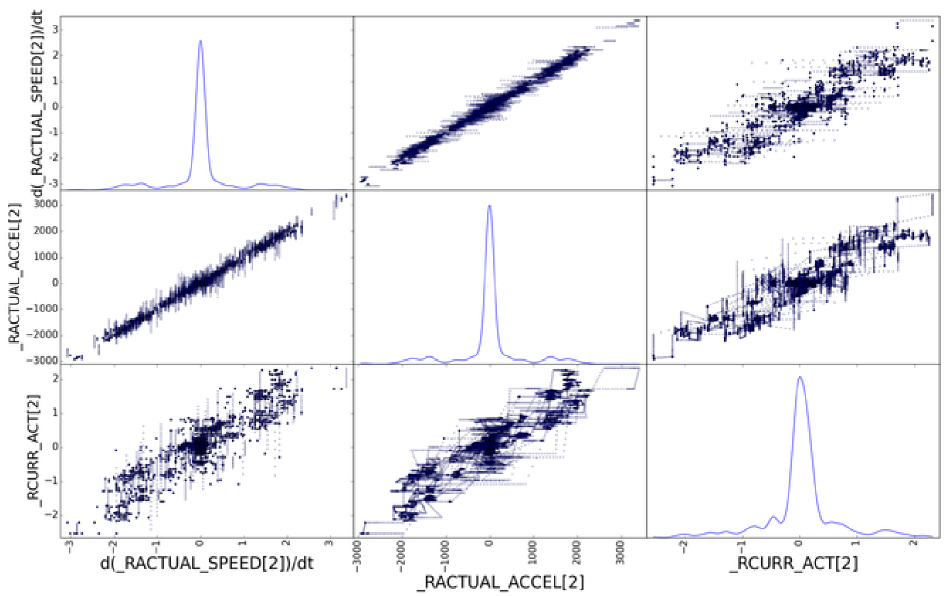


Figure 5.58: *Correlation of velocity derivation, acceleration and current of axis 2*

Plausibility test on position and velocity: A plausibility test is run as a practical use case for data correlation. The expert wants to test whether the formula $v = \frac{d}{dt}x$ is valid for axis 2 of the robot. Thus he tests, if the derivation of position data (x, `_RACTUAL_POS[2]`) is positively correlated with the measured speed data (v, `_RACTUAL_SPEED[2]`) at the time (t). He designs the rule from Figure 5.57 using the rule editor. The plot function delivers Figure 5.58 as a result. The expert's hypothesis is confirmed by the clearly positive correlation of both variables.

Plausibility test on velocity, acceleration and motor current: The expert wants to detect a correlation of speed (v, _RACTUAL_SPEED[2]) and acceleration (a, `_RACTUAL_ACCEL[2]`) of the robot's axis 2 in measured data. Demonstrating the correlation demands the derivation of speed to time (t) ($a = \frac{d}{dt}v$). In addition the experts investigates if the motor current (`_RCURR_ACT[2]`) of axis 2 correlates with acceleration of axis 2. Data is sent to the plot function by TARDIS (see Figure 5.57). Obviously all three variable correlate positively. Thereby the connection of acceleration and speed of the robot's axis 2 is verified. Furthermore it is visible that the correlations between motor current and both other variables roughly meet a positive linear slope.

TARDIS implementation Since the derivation of variables is a crucial part for the requirements of scenario 5, the rule editor has to offer a `Derivate` function. The function has exactly one input and one output signal and provides the input deviation according to the index variable. I.e., a data structure manageable as `DataFrame` is assumed to ensure a smooth execution of the derivation by the pandas library (cf. Chapter 5.4.4.4). Further, the selection "Correlation" is added for realization in the plot function.

Python implementation The `pandas` library uses functions of the `matplotlib` library for graphic display. With the correlation matrix being a composition of several plots from `matplotlib`, `pandas` offers a pre-assembled function to generate the correlation matrix from a `DataFrame`.

From the linear interpolation of data in preprocessing (Section 5.4.2.4), the derivations produce imprecise results after some time. As an example, in Figure 5.59 the first two derivations of position to time are illustrated for the position of the robot's axis 2. In Figure 5.60 the display of the time sector between 2.8 and 4.0 seconds is magnified. This clearly shows interpolation errors already had occurred during the first derivation. It is hence not possible to give a realistic statement for the correlation of derived variables .

5.4.5.8 Scenario 6: Complex cycle comparison

Application

Comparing cycle 40 to certain reference cycles regarding the acceleration of axis 4 does not suffice for the expert. He wants to examine for various variables, at which time points there are deviations from the 100% area of the reference cycles.

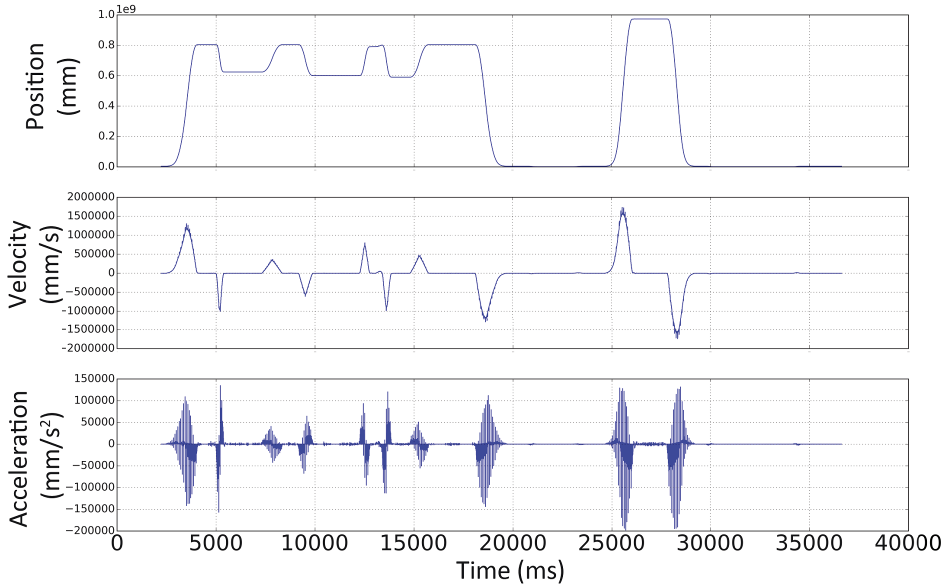


Figure 5.59: *From up to down: linear interpolation of the position of axis 2, first derivation of position with respect to time, second derivation of position with respect to time*

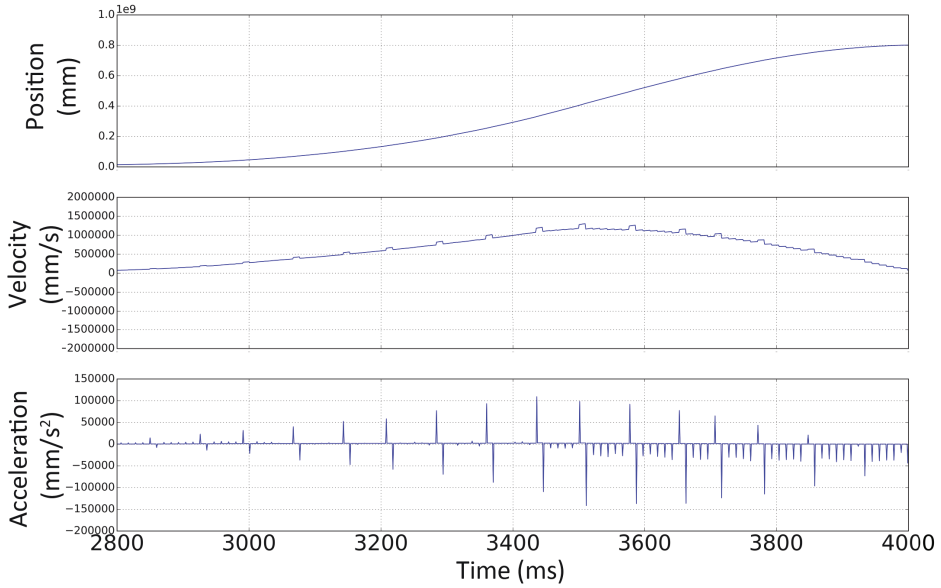


Figure 5.60: *From up to down: linear interpolation of the position of axis 2, first derivation of position with respect to time, second derivation of position with respect to time*

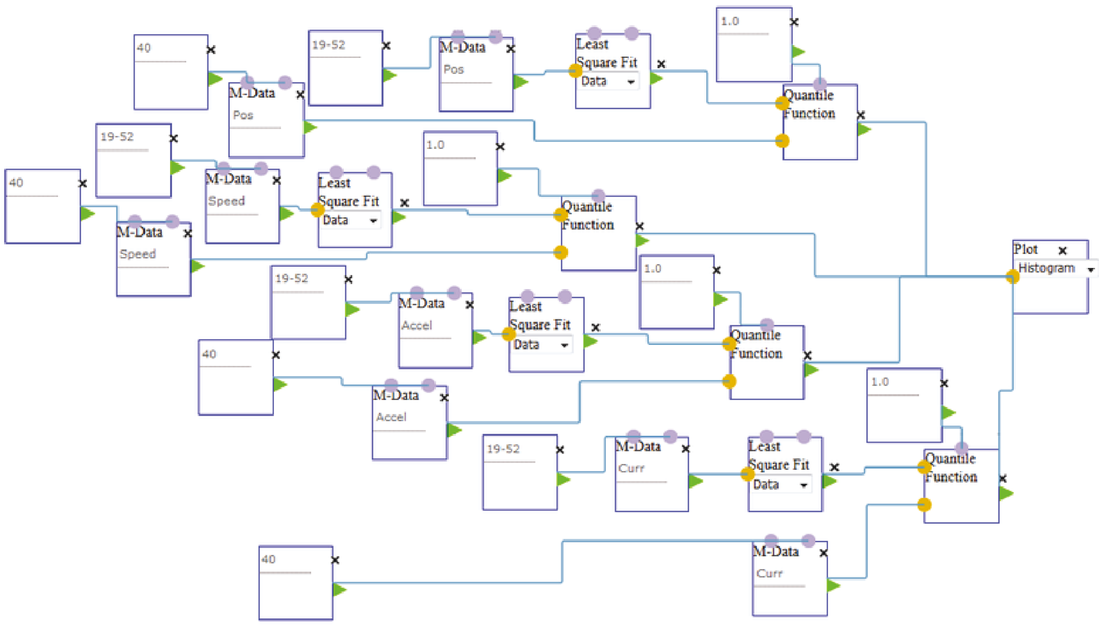


Figure 5.61: *Example of a rule for histogram out of several Binary Streams in rule editor*

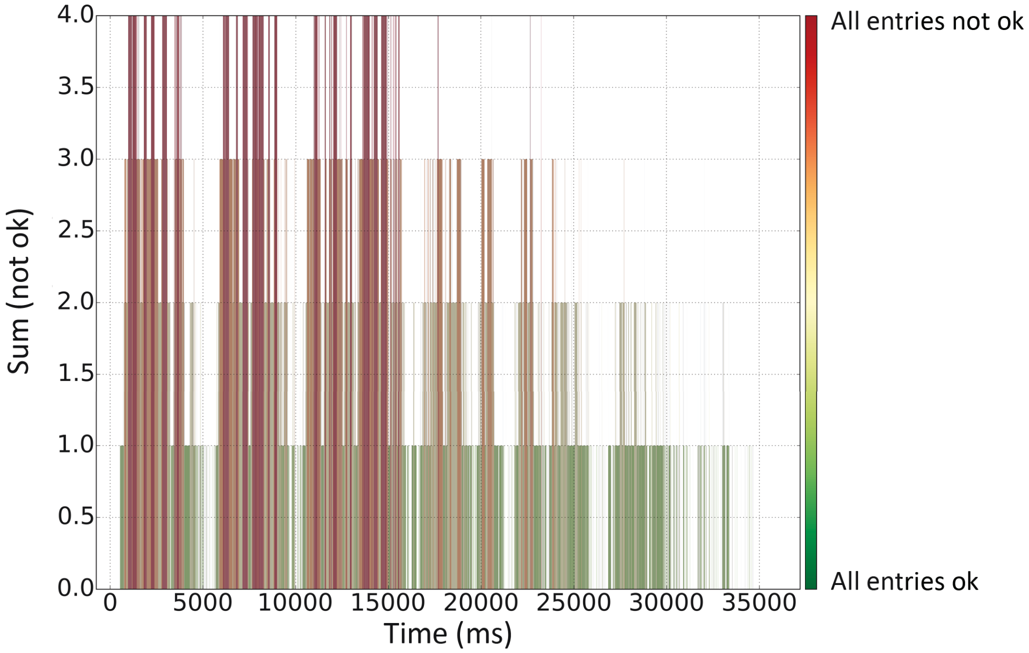


Figure 5.62: *Validation of cycle 40 with four heuristics*

Figure 5.61 depicts the rule to be analyzed by the expert. He uses two times four M-Data modules, equivalent to position, speed, acceleration, and motor current of axis 2. One M-date symbolizes cycle 40, the second one is utilized for the reference cycles of the same variable (here cycles 19 to 52, except cycle 40). Time deviations of reference cycles from the database must initially be corrected via the Least Square Fit function. Then Quantile Function is run according to the respective variable, and the result is transferred to the plot function as a Binary Stream. The result of this rule is seen in Figure 5.62.

The number of heuristics including deviation of cycle 40 from training data is determined for each time point. Since altogether four heuristics are employed, cycle 40 can at most derive from four heuristics. Figure 5.62 exhibits that cycle 40 in some phases differs from all of the heuristics. But still there are time points with data from cycle 40 residing within the 100% quantile area of training data for all heuristics. This is for instance seen from time point at 34 seconds.

Python implementation With Python's extreme flexibility when working with data types, there are no problems to edit data consecutively with various functions. For this scenario, four variables of cycle 40 are tested for the respective 100% area of reference cycles. Employment of different parts of reference cycles with the Quantile Function regarding the same variable implies a correlation of results, and thereby falsifies the result in histogram display. Thus the Quantile Function should only be used independently from each other, when giving different percentages regarding a single variable.

5.5 Process visualization (ProcVis)

As shown in the last section, a 3D model of the plant can prove useful for the analysis of several plants. When we consider the procedure visualization of one plant in the machinery optimizer, it might also be of use to use an animated 3D model. In order to achieve this, a process visualization was implemented in Unity on base of the same condition monitoring dataset.

5.5.1 Concept

It is difficult to identify the production process in complex plants. The evaluation of the Machinery Optimizer in Section 5.3.3 had shown, that only one out of six expert not familiar with the plant context were able to describe the production process correctly. But this type of situation awareness is strongly required in a remote teleanalysis scenario. The evaluation showed that some remote analysis tasks had not been possible without a proper knowledge of the process. The promising approach of using a static 3D model for data mining of the last section has been further developed in the ProcVis application.

As described above, the 3D model of the plant was provided by the production factory of P&G. KUKA Industries also uses animated 3D models in the software

Provis, but they do neither visualize the whole plant nor the production process. Goal of the work was to animate the 3D model in Unity with the recorded Condition Monitoring data. The developer environment Unity was used once again, because it is able to produce applications for all established operation systems.

The following steps were required:

1. Reconditioning of the CAD data in Blender
2. Integration of the 3D model in Unity
3. Animation of the 3D model with the Condition Monitoring data set
4. Schematic visualization of the produced workpiece with colored markings
5. Visualization of control and sensor data

The production process has been introduced in Figure 3.7 in Section 3.3.1. In order to understand the problem and to introduce the colored markings, Figure 5.63 describes a handover diagram of the parts between the components. It also shows that always two parts are produced at the same time: The first part is produced in the molding machine (red), is transported to the montage system, where it turns green. As it has been brought back for to the molding machine, another red part has been produced by the molding machine. The green part stays in the molding machine and turns blue after the second injection molding. It is then release to the "ready parts" deposit.

5.5.2 Implementation

After the 3D models had been reconditioned for the use in this application, they are imported in Unity. For the transformation of Unity objects, scripts in C-Sharp [174] or Javascript can be used. In this application, we use Csharp. The Condition Monitoring data has been exported as CSV file (comma separated values [175]), but also a network connection could have been used.

Figure 5.64 shows a Unity environment with the project. In the center, the 3D model is visualized. If the central "Play" Button is hit, the simulation starts. But in Unity, the controlling scripts are not self-explaining, thus a short documentation is added here.

Figure 5.65 shows an UML diagram of the main scripts used in this implementation. They all inherit from the UnityEngine method "MonoBehavior" and will be described in detail in the following sections.

5.5.2.1 Machine Controller

The class `MachineController` itself does not process incoming data. Depending on the data source, a class like `CSVMachineController` inherits from `MachineController` and implements the specific realization (like reading from a

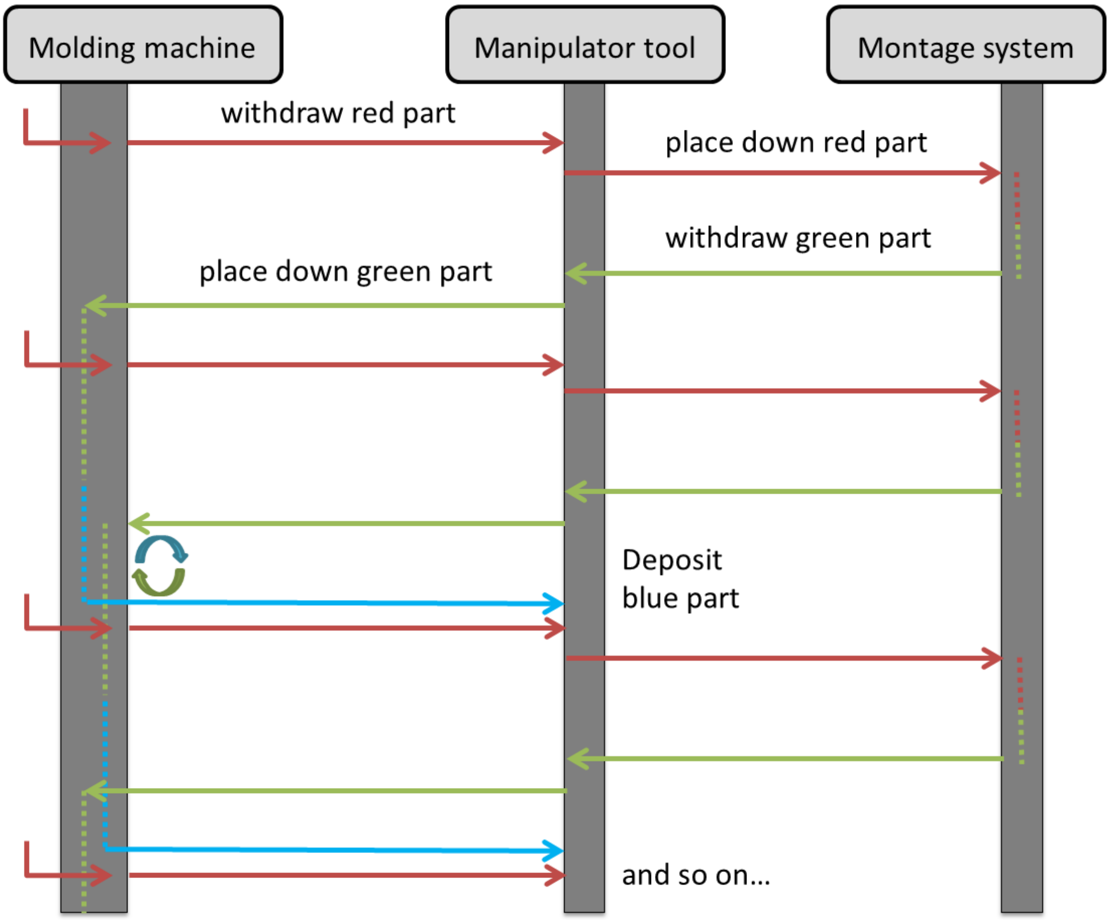


Figure 5.63: *Production process visualization for ProcVis*

file). Another Controller could be used to implement a UDP network connection and write the received data into the `currentSignal` array. The signal names and value are represented as String arrays. `MachineController` provides the access methods (by name or by position). In order to enable time triggered events, `MachineController` also provides several time management functions. It provides access to the time in seconds, that has already passed since the begin of the simulation (`float _currentTime`) and the current time slot (`int currentTimeSlot`).

5.5.2.2 CSVMachineController

The script `CSVMachineController` is represented in Unity as depicted in Figure 5.66. The CSV file can be added per drag & drop. The first line of the CSV file provides the list of signal names. The other lines are the signals in float, binary or integer (as described in Section 5.4.2.1). Each line represents a time step of $11ms$. As described above, the robot cycle theoretically is $10ms$, but the values are not stored in equidistant intervals and the distances are most of the time larger than $10ms$. For simplification, equidistant intervals of $11ms$ were chosen. At every

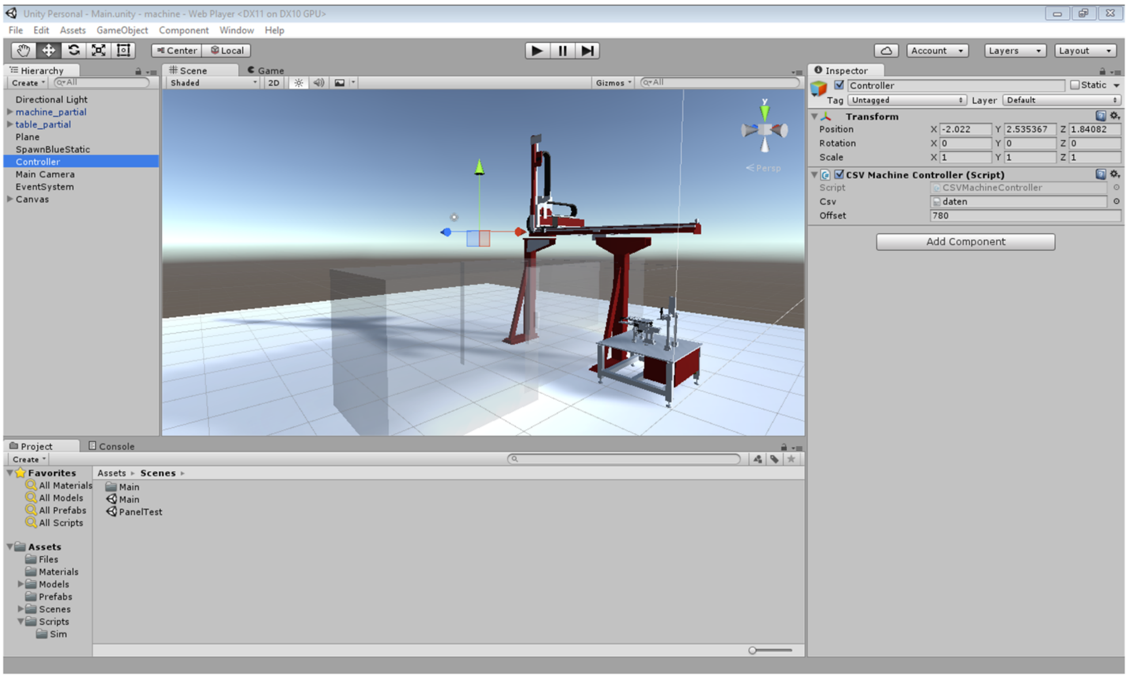


Figure 5.64: *Production process visualization for ProcVis*

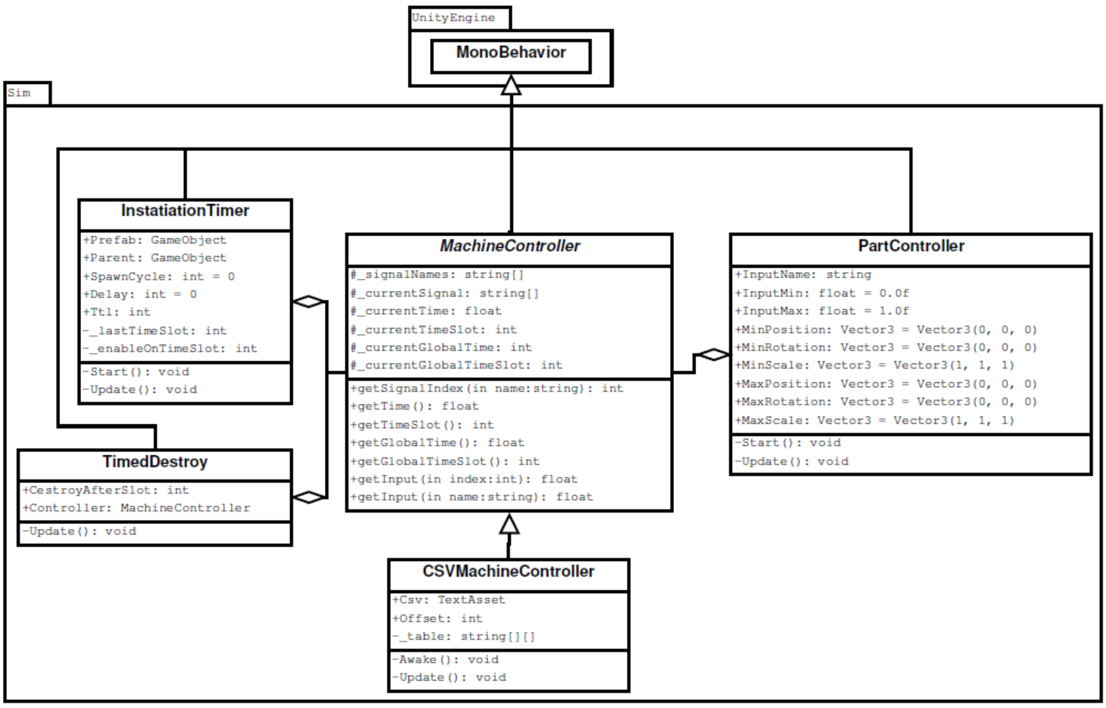


Figure 5.65: *UML chart for ProcVis classes in Unity*

Figure 5.66: *Unity GUI element for CSVMachineController script*

instance of time, the according entry is drawn from the table and presented to the other scripts. When the bottom of the table is reached, the first signal entry is used again, because the process is played in a loop.

5.5.2.3 PartController

This script is responsible to provide the current position, rotation and size of the corresponding `GameObject` like a manipulator axis. In Unity this is displayed like in Figure 5.67.

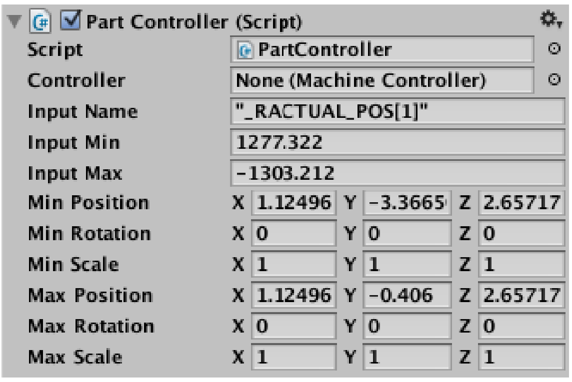


Figure 5.67: *Unity GUI element for PartController script*

The script needs a `MachineController` as data source. If this field is empty, the first `MachineController` of one of the parent objects is used. It further needs the "Input Name" of the signal, which should be used (in the example the current position of Axis 1). The values provided by the Condition Monitoring are not scaled and it depends on the manipulator configuration, where the maximum or minimum positions of a movable axis are. In order to provide a simulation, the scaling is inserted manually here. The maximum and minimum of the applied signal are inserted and also the position of the object's minimum and the maximum positions. The script will restrict the movement of the axis within those borders and normalize the Condition Monitoring data according to those restrictions.

5.5.2.4 InstantiationTimer and TimedDestroy

In order to visualize the process, additional objects need to be inserted and removed to specific instances of time (the produced parts). This is realized by the script `InstantiationTimer`, whose GUI element is depicted in Figure 5.68.

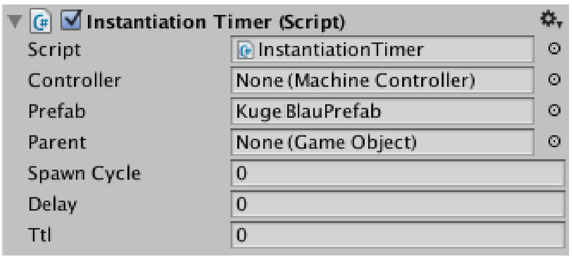


Figure 5.68: *Unity GUI element for InstantiationTimer script*

The script also needs a `MachineController` as data source and uses the first Controller of the parent objects, if no pointer has been set here. The Prefab specifies the Object, which is generated, in the example it is the blue sphere (third part state). The parent is the object, who will be parent to the newly generated object. The "Spawn Cycle" specifies the machine cycle, the delay to the beginning of the simulation and the "time to live" after its generation. A `TimedDestroy`-Script is added, if that is greater than 0. This script will destroy the object after the preconfigured time.

5.5.2.5 Util

The class `Util` stores some small helper functions, which can be accessed from several functions, like a csv-parser, a convert function and a search function for the `MachineController` on the parent objects.

5.5.2.6 GUI classes

Additionally, there are several GUI scripts, which will not be described in detail. One example is `PinOutPanel` which is used for showing the binary sensor values in a list next to the moving plant, which is shown in Figure 5.69. If a binary value is 1, the ID is highlighted green, else it is grey.

5.5.3 Results

Figure 5.69 shows a newly started simulation process. The red part has been generated with the first injection molding process. On the left, some of the relevant binary IOs are displayed with their values. The red part is transported by the robot from the molding machine (transparent large block on the left) to the montage system on the right.

Figure 5.70 presents another viewpoint, like of someone standing in front of the montage system. The red part has been inserted here and becomes a green part after the insert of the "gore membrane".

The green part is then transferred to the molding machine, where another red part has just appeared due to the completion of its molding process. After the second injection molding process, the green part turn blue. After each molding process,

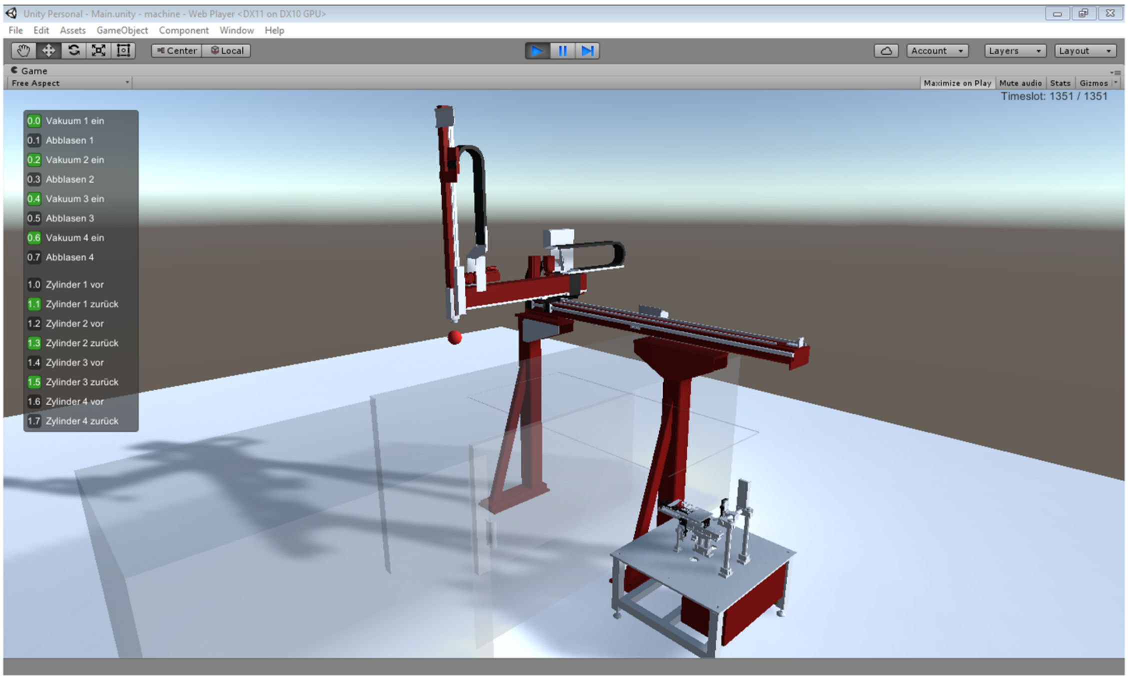


Figure 5.69: *Begin of process (red part spawns)*

Figure 5.70: *Second working step (red part transforms to green part)*

a green part is inserted into the molding machine and a red and a blue part are obtained. The red part is inserted in the montage system and the blue part will be put into the "ready parts" deposit. This situation is displayed in Figure 5.71 from a top perspective, showing all three parts together.

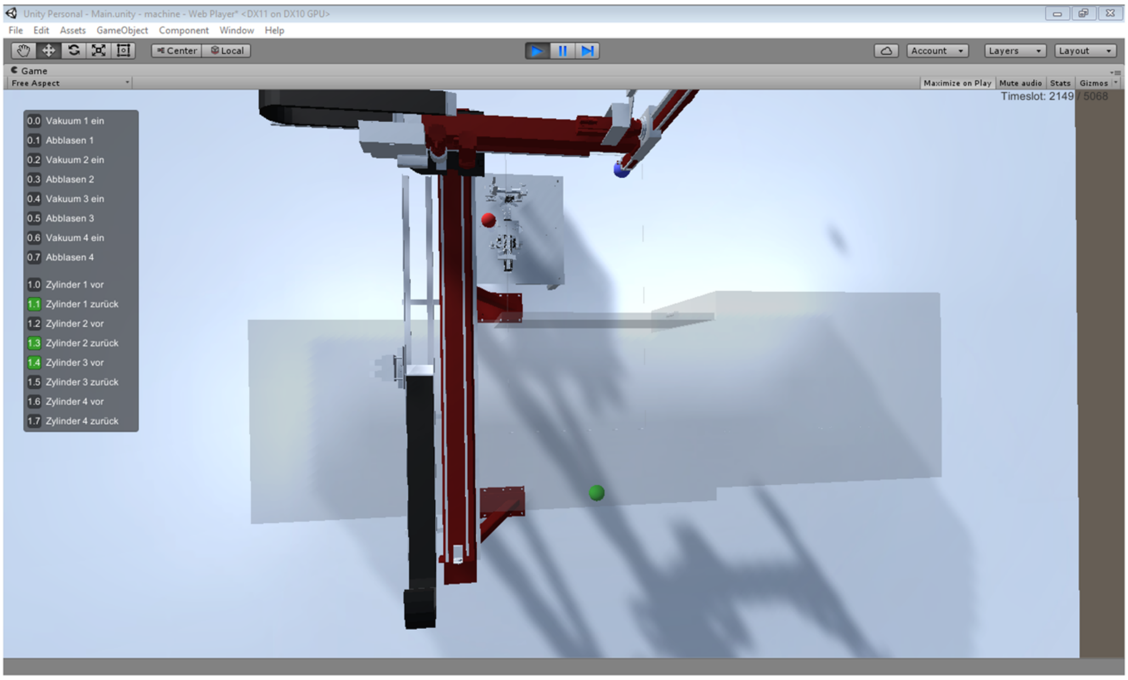


Figure 5.71: *All three working status parts visible*

5.6 Summary

This chapter has described several innovations in the field of remote analysis of production plants, which are summarized in Figure 5.72. The base for remote analysis provides a condition monitoring system which collects and saves data of the production plant. The chapter provided several applications, which enable the analysis of recorded data with respect to one production cycle, with a view on several cycles or even plants and a 3D model process visualization.

5.6.1 Condition Monitoring

With the help of the manipulator manufactorer KUKA Industries, Section 5.2 describes a simple condition monitoring system in the robot control was established. With the help of the new software, several variables can be saved at runtime and written in a file. The robot can acquire information from other parts of the production unit via the different in- and outputs. Each measurement records 54 variables. The measurements are repeated in irregular intervals between $10.8ms$ and $12ms$. There are different delays in the recording of the values. One cycle of the robots

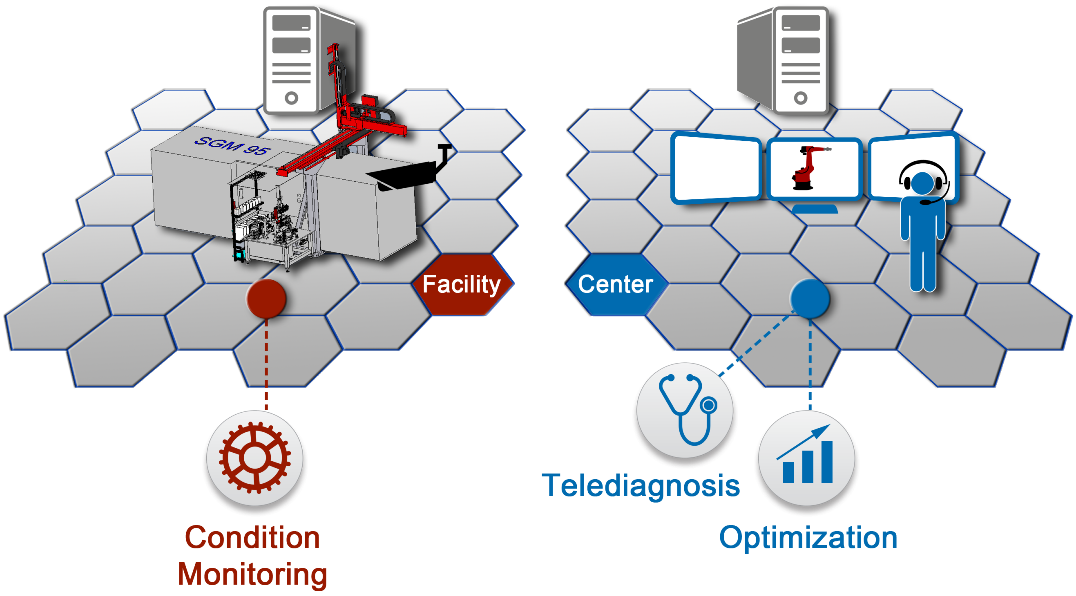


Figure 5.72: *Summary of the innovations: Asynchronous remote teleanalysis*

process takes about 35 seconds. This results in about 3000 measurements per cycle with 54 variables each. In one hour about 133 megabytes of robot data is recorded.

The system is able to check variables for threshold on runtime and produce warnings and errors on the manipulator teach pendant.

5.6.2 Machinery optimizer

The software "Machinery optimizer" described in Section 5.3 uses a data package generated by the condition monitoring system together with recorded video data of the robot. The idea of the software is to implement a interface similar to a video player showing one production cycle of the plant. For each timestamp of the video, the related sensor data and the active command line in the source code is provided. A user can watch the cycle in slow motion or even backwards and select the interesting variables, which are displayed together in a graph. It is also possible to set specific breakpoints based on variable threshold.

Section 5.3 describes the development of the idea for this software, shows the realization of different stages of the prototype and finally presents the results of an evaluation performed with specialized end users out of the industrial domain. The results showed, that the use of the software for optimization substantially saves working time and can provide new insights, which have not been possible before. For example, several experts showed concrete approaches to shorten the plant process duration for at least two seconds.

5.6.3 Complex analysis tool (CAT)

A further development is presented in Section 5.4. As the "Machinery optimizer" only presents the view on one plant cycle, this does not suffice for the detection of slowly prevailing changes due to abrasion. It also occurred to the facility staff, that the statistical comparison of data of different but similar plants might be useful. The main problem of the big data processing in automation industry is that specialists either have deeper knowledge on data mining or on the industrial context. The "Complex Analysis Tool" addressed this problem by providing data processing with a rule-based system and special data mining scripts which address the problems in the plant. The goal of the CAT software is industrial specialists not having to touch the source code and operating only in the interface. The data selection can take place with a 3D model in Unity, the data is then processed with a rule based system and plots are generated. In six application scenarios derived from the industrial production setting, the usefulness of this software was proven.

In the provided example, the sensor data collected by the Condition Monitoring system is used along with the human estimation derived from the inspection test of the mobile device (data of two weeks test mode).

5.6.4 Process visualization (ProcVis)

The CAT software used a static model of the plant. The "Machinery optimizer" focused on one cycle and the production process. In order to bridge those two ideas, another part project with the name "ProcVis" was created, which uses a dynamic 3D model animated with the sensor data (described in Section 5.5). It can also be used for simulating different robot motions as well as for visualizing the produced parts.

The stationary 3D model provided by the production facility was reconstructed in Blender in order to achieve the three Cartesian robot axes in Unity. The movements have been realized in Unity with an external included CSV file containing the position information derived from the condition monitoring system. Other parts of the sensor data have been realized as interface objects in Unity (like "on" and "off" highlights for binary data). At last, the production process was realized with three product states depicted in three colors. The produced parts are generated in the molding machine, moved by the manipulator and finally disposed as ready parts.

5.6.5 Economical benefits

In the project "MainTelRob" an economical analysis of the scenarios 4 and 5 was performed on base of the project results in the plant of P&G.

An important goal of planned maintenance is the ideal timestamp of the maintenance measures. The ideal maintenance cycle uses the whole effective period of the machine elements. By using Condition Monitoring (scenario 4), the planned maintenance cycles can be extended. On base of the manufacturer recommendations for the component exchange, plant specific change cycles can be estimated. Instead of time-based maintenance, a cycle-based maintenance is strongly required. Costs for material and personnel can be reduced by approximately 5-10 percent. The precondition for predictive maintenance is Condition Monitoring. There is a high potential for reaching ideal maintenance cycles.

The largest savings potential is seen when using the optimization tools. Different specialists can work offline on the ideal process and the ideal parameters. By using ideal parameters, the productive time of the robot axis can be extended. By using optimal parameters for the interfaces and velocities, the cycle time can be reduced. Different specialists can discuss the optimization approaches and implement them in different similar plants. Breakdowns can be omitted. In the evaluation of the machinery optimizer, the actual cycle time of 33 seconds could be reduced by 2 seconds. This means a direct saving on the part price.

5.6.6 Conclusion

This chapter showed the different possibilities of using recorded machine data for asynchronous analysis of a plant and its production process. This is especially important, if several experts are working together remotely in order to overhaul and optimize a production process. This often requires special knowledge from different domains. The most important role of teleanalysis software is to visualize the process and the plant context in order to provide a holistic understanding of the problems that should be solved. We showed, that with the described methods, optimization processes can be faster and more effective compared to the approach used in the production environment.

Chapter 6

Telemaintenance as Human Supervisory Control

This chapter describes the development and the installation of a telemaintenance architecture for live remote support. This especially includes the teleoperation of the industrial robot in an active production environment and the video interaction between a remote expert and a service technician. This architecture is evaluated in a proof of concept experiment of a remotely guided motor exchange. The research focus was to provide the remote user with a maximum of situation awareness. Finally, we evaluate different approaches on enhancing the situation awareness with virtual and augmented reality.

6.1 Survey

This chapter broaches the issues of the whole Extended Human Supervisory Control model introduced in Section 2.6.3. Together with the innovation covered in the last chapters, the development process in this chapter focuses on the architecture and the human collaboration as depicted in Figure 6.1.

The sixth scenario of the project "MainTelRob" covers the telemaintenance task and therefore is an integration of all the other functionalities under real-time conditions. In this complex case, the following questions will be regarded:

- Which infrastructure and architecture are required for telemaintenance?
- Which functionalities help to get an appropriate problem solution?
- How can the results of the other scenarios be used?
- Which features help the expert to gain situation awareness?
- What should be considered in order to optimize the virtual team performance?

6.1.1 Motivation

When a fault occurs at a plant, the service technicians immediately try to fix it. A failure often leads to an unplanned breakdown, but at least to a lower product quality. Both is not acceptable. Inspection measures can detect indications prematurely

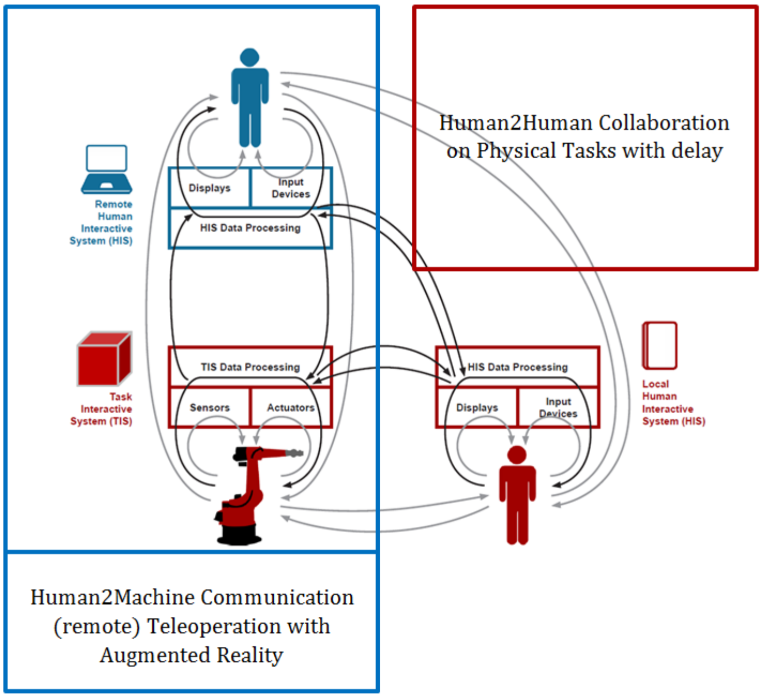


Figure 6.1: *Context of this chapter visualized in Extended Human Supervisory Control Model*

to prevent an unplanned breakdown. At least a good inspection history facilitates the fault diagnosis. Otherwise, the fault diagnosis itself can take a long time, in which the plant does not produce. This is very expensive, especially if the plant is on a critical path.

If the service technicians either cannot succeed in the fault diagnosis or in the necessary repair tasks, the advice of specialized personnel of the robot manufacturer is required. At the moment, service is provided via telephone. A lot of cases cannot be supported via mere audio connection and technicians need to visit the production site, in order to provide proper assistance. This should now be substituted by a telemaintenance solution in order so safe time and money.

6.1.2 Goals

The expert personnel would provide the best support, if they were present at the erroneous plant. Consequently, the telemaintenance interface needs to provide the "feeling of standing right in front of the manipulator". This is on the one hand the mental state of situation awareness (cf. Section 2.5.2), described by numerous researchers. It involves the acquiring and transferring of all relevant information and providing it in an intuitive and not overwhelming fashion. On the other hand, this also should provide the same scope of influence for the remote application, particularly the possibility of controlling the robot.

The goals for telemaintenance can be described as follows:

- Acquiring all relevant data from the manipulator and the plant (scenario 4 Condition Monitoring, Section 5.2), as well as the inspection history (scenario 2 Visual Control, Section 4.3) and the previous repair operations.
- The expert needs the possibility to consult a knowledge base, this can be documentation (scenario 1 Maintenance, Section 4.2) but also a data base from several plants (CAT, Section 5.4).
- All relevant data needs to be provided in real time via the Internet. This presumes an **effective network architecture**, which also emphasizes computer and network security.
- **Teleoperation** of the industrial robot has high safety requirements, i.e. compensating network delay and to drive in a safe state in the cause of network loss.
- The remote expert and the service technician need to have a **shared visual context**, this includes **video material** but also may include repair instructions (scenario 3, Section 4.5) or a representation of the plant and its production process (scenario 5 Machinery Optimizer, Section 5.3 or ProcVis, Section 5.5).
- The data from the different other scenarios need to be accessed in a **common software architecture**.

6.1.3 Related work

The first studies in the area of collaborative work [176] investigated the conversations and information exchange of two workers who worked side by side on a montage task. Those were mainly the identification of goal objects, the instruction for tasks and the confirmation of finished tasks. Further observation studies [177] introduced a common work field, which can be used for sketches and writings. It has been shown that this common view helped to support the work process and to convey information. For a remote constellation, research [178] showed, that collaborative groups on the same workplace performed better than spatially separated coworkers, but also showed the potential of multimedia systems.

Another study focused on the question, which visual information provides a benefit for a coworking team. The study [179] measured the performance of a person working alone on a bicycle repair task with a group of a remote mechanic and a local worker on the same task. The experiment provided evidence for the importance of shared visual context for remote collaborative work. Further investigations [180] on puzzle tasks showed, that shared visual context was also important for situation awareness and grounding.

Teleoperation of robots has been discussed and implemented a lot since the beginning of robotics research. For an overview of applications see [124]. We focus on teleoperation of industrial robots, which have been of early [181] and late [182] research interest. The application for teleoperation of industrial robots lies mainly in the context of hazardous environments [183], but in our scenario we cover maintenance scenarios similar to those for tunneling machines covered in [184]. As we aim to facilitate teleoperation via the Internet, the control perspective with time delay over the Internet [185] has to be integrated as much as human perception of time delays [186].

There have been several proposals for teleoperation architectures [187], [188]. Reis Robotics (now KUKA Industries) provided an architecture that includes basic teleoperation functionalities [189]. However, there is a need for an enhanced architecture that also includes further services needed for Human Supervisory Control, such as synchronized video feedback. A lot of research interest lies in the data flow in maintenance scenarios and how to manage it [190], but this approach has not included teleoperation with video feedback. The closest related work has been provided by [191], who covers multimedia streams, too [192], but does not include other kinds of data needed for maintenance. To our best knowledge, there is no integrated teleoperation and maintenance architecture yet.

6.2 Telemaintenance architecture

6.2.1 Contextual analysis

Again, interviews with both sides helped us to understand which services are needed to fulfill the scenario of telemaintenance. We asked about the current situation in case of service and also collected ideas for an envisioned situation with telemaintenance.

The service technicians at P&G are responsible for prohibiting breakdowns of the plant. They need to find the fault and fix it as soon as possible in case a breakdown happens nevertheless. P&G has a well-trained team of specialists for this purpose, who can handle most occurring problems on their own, but other customers of KUKA Industries may not have such a team. We therefore consider a situation with the facility technician not being able to find the failure and to fix the problem. His shift supervisor decides to call the customer support of KUKA Industries (depicted on the left side of Figure 6.2) to ask for help. An expert connects to the robot and communicates over a mobile device with the technician, so that they both work together to find the failure and fix it.

In practice, "telemaintenance" is provided mainly via telephone communication without any visual feedback or remote access. Furthermore, the expert is specialized on the industrial manipulator, but is not familiar with the specific plant context, as robots are used in a variety of settings. That is why some tasks can simply not be supervised remotely, if the service technician has not been specifically trained on the task in advance. In such cases the expert service personnel travels to the facility and performs the required repair tasks there.

Of course, remote access to the plant and teleoperation of the robot are absolutely

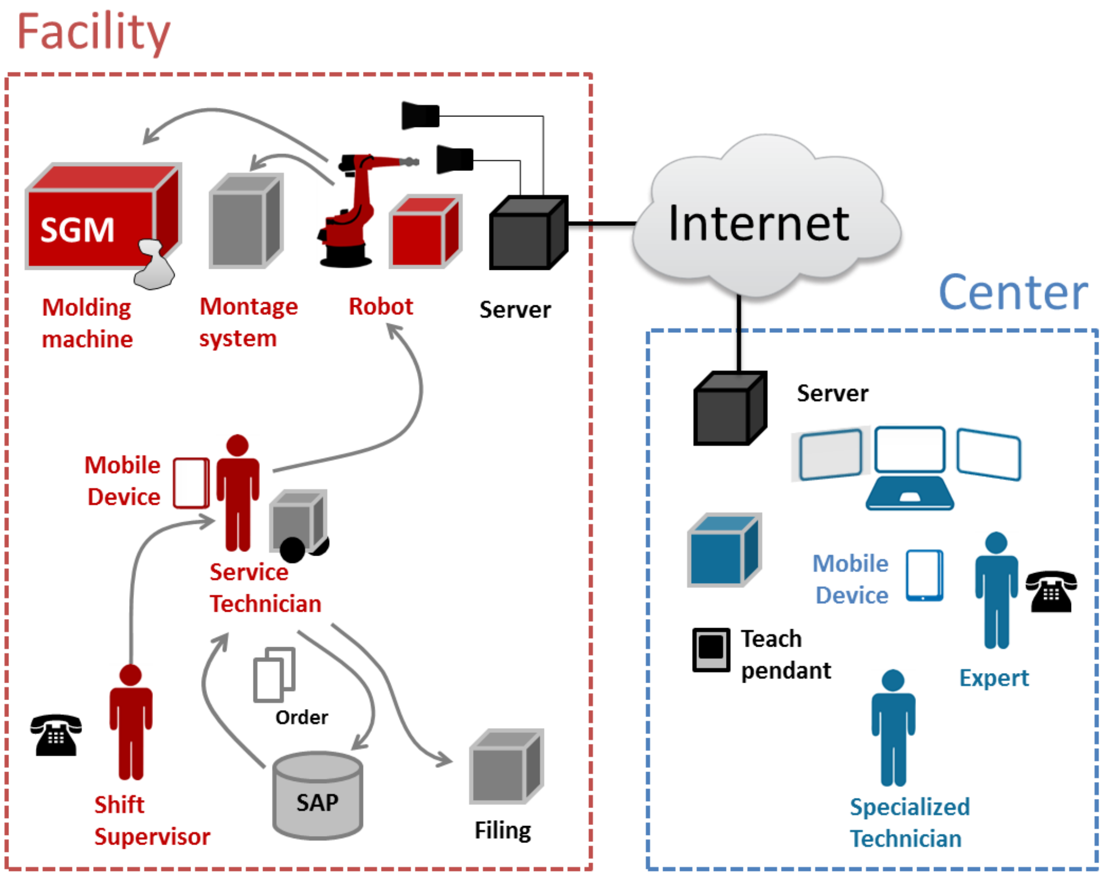


Figure 6.2: *Flow Model for telemaintenance*

mandatory. This was obvious for the project team to involve a lot of organizational tasks during the start-up of the network connection. The interviews also revealed the importance of a reliable human communication channel and the possibility of file-exchange. It also turned out that the workers on both sides considered video-based interaction a very important feature. The expert, who knows the robots very well, but is facing the context of the specific plant for the first time, needs appropriate situation awareness. Also, any kind of teleoperation is only feasible with proper feedback information. The technician has a need for videos too, in which the expert explains tasks, the technician has never conducted before. The telemaintenance scenario therefore involves a specific combination of different services like teleoperation, communication via chat and voice, video streaming, file transfer and Augmented and Virtual Reality.

In the case of telemaintenance, both the local technician and the remote expert perform supervisory control. Due to safety issues in Europe and other parts of the world the legal permission to teleoperate an industrial robot in an automation facility cannot be granted without a local supervisor. The latter needs to maintain control of the industrial robot and his or her control inputs need to have priority over remote commands. In this extended Human Supervisory Control setting the communication between both partners in the virtual team plays an important role for the task achievement.

6.2.2 Infrastructure

For the development of prototypes for scenario 6, extensive preliminary work was necessary which is described briefly in this section. The goal was to develop the entire architecture depicted in Figure 6.3 within several steps.

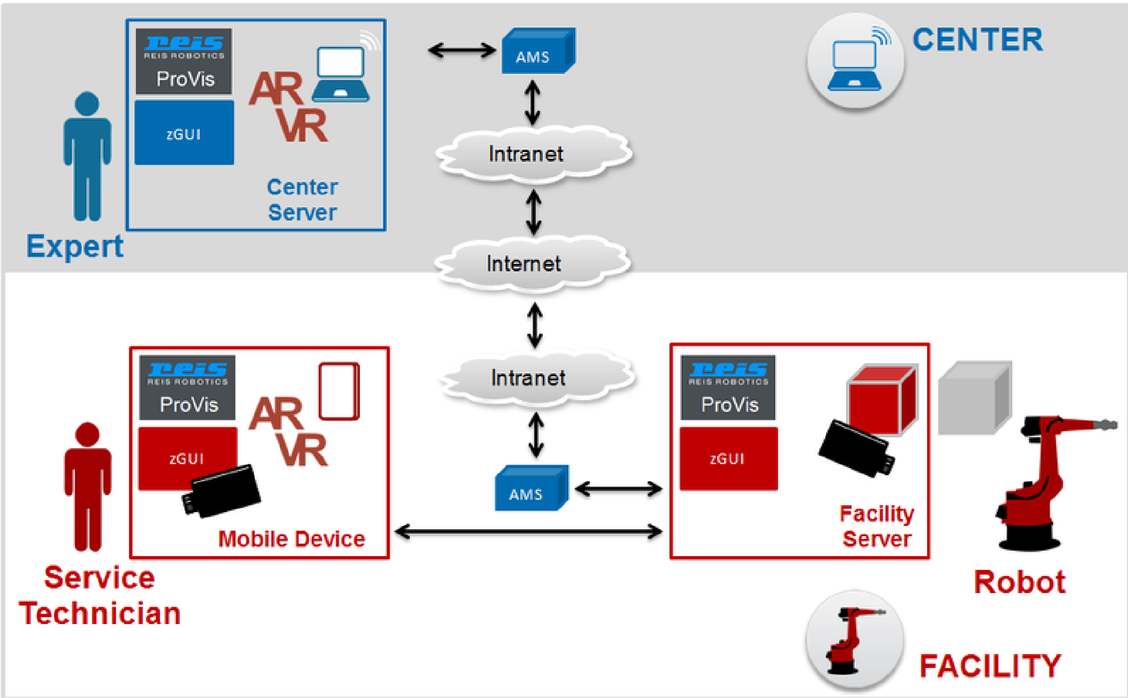


Figure 6.3: *Schematic overview to the telemaintenance architecture*

6.2.2.1 Network connection and security

The biggest challenge was the network access from KUKA Industries or (for testing purposes) from the Zentrum für Telematik to the plant at P&G. This required the application and establishment of a Nortel VPN to the P&G VPN Gateweay (in Brussels), which is shown schematically in Figure 6.4.

On base of the network infrastructure various measurements of the connection quality have been performed. Several security concepts have been tried out. The secure access from the telemaintenance center to the plant is performed via a second VPN tunnel (shown in Figure 6.4 and referred to as AMS VPN Tunnel).

At the plant on the right side a test server ("Facility Server") had been provided by P&G, which can be connected to the manipulator controller. For the remote control

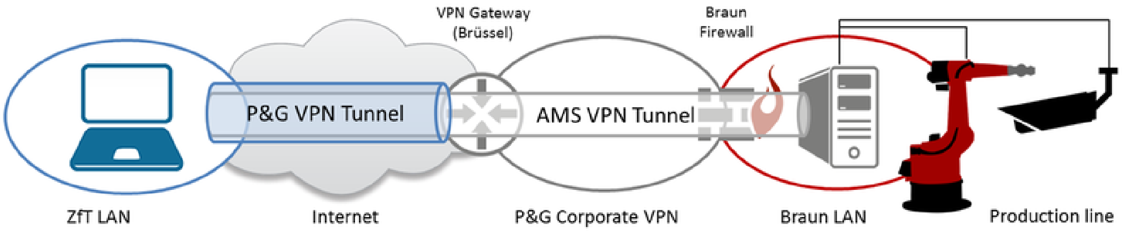


Figure 6.4: *Model of the VPN Tunnel*

a specially developed manipulator firmware was used. For this case a disclaimer of warranty had been arranged.

A remote access took place every week since the May 8th, 2014 from the Zentrum für Telematik, while a service technician assisted on the facility side. To ensure a safe telecontrol (nearly biweekly), the manipulator was secured with safety wedges. Also for this case a disclaimer of warranty had been arranged.

6.2.2.2 Video cameras

As mentioned above, a video feedback infrastructure is highly necessary for teleoperation. For the prototype setup, two cameras were installed at the P&G facility. The first camera provides a top view, as shown in Figure 6.6 (on the right side). The second camera is installed in the scaffolding of the handling process and can be rotated, so that it is able to monitor a large area on this side of the plant. Both devices are IP-cameras that provide multiple encoded HD video streams.

The installation process of the cameras was carried out under the supervision and with the consent of the works council of P&G. One special matter of importance was insuring the personal privacy of the workers. That means monitoring as little personnel as feasible, and especially not observing people on their way from or to the break room adjacent to the prototype facility. We agreed on installing covers to restrict the range of the turning camera, as well as to mask several areas showing the corridor (the masks are seen in Figure 6.6). In addition, we installed several information signs and a light, which glows red when the cameras are addressed externally. The entire video infrastructure setup was completed in December 2013.

To provide a feedback to the actual activity on the robot to the external expert, several video cameras have been installed. The cameras needed to be purchased, installed and included in the network. Figure 6.5 provides a schematic overview to the plant and the used video camera positions (1 and 2).

In Figure 6.6 additionally the photos of the cameras (left) and the camera perspective of camera 1 are depicted.

6.2.2.3 Test environments

1. **Control test environment** (in Figure 6.8)
   The purpose of this testbed was the demonstration of the transmission path

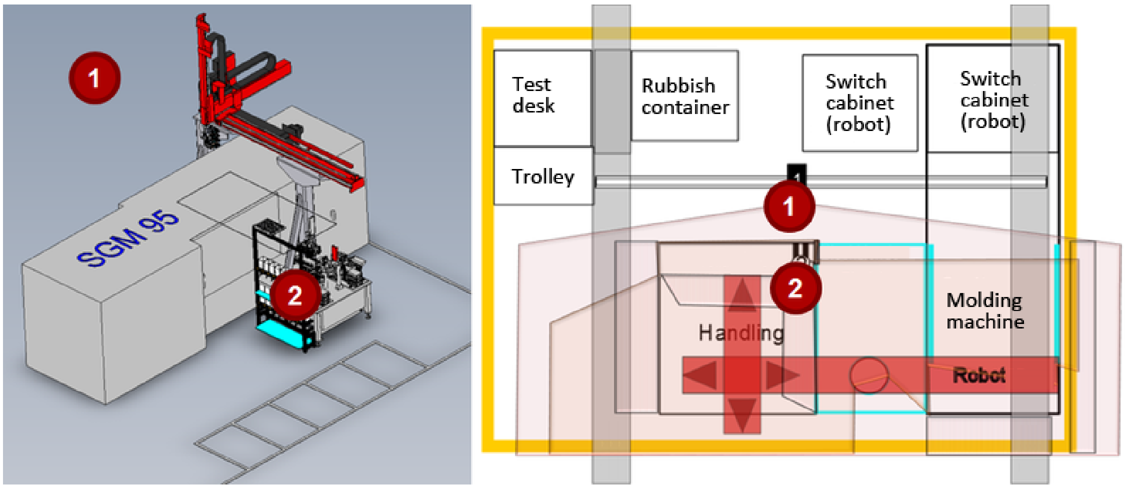


Figure 6.5: *Camera positions*

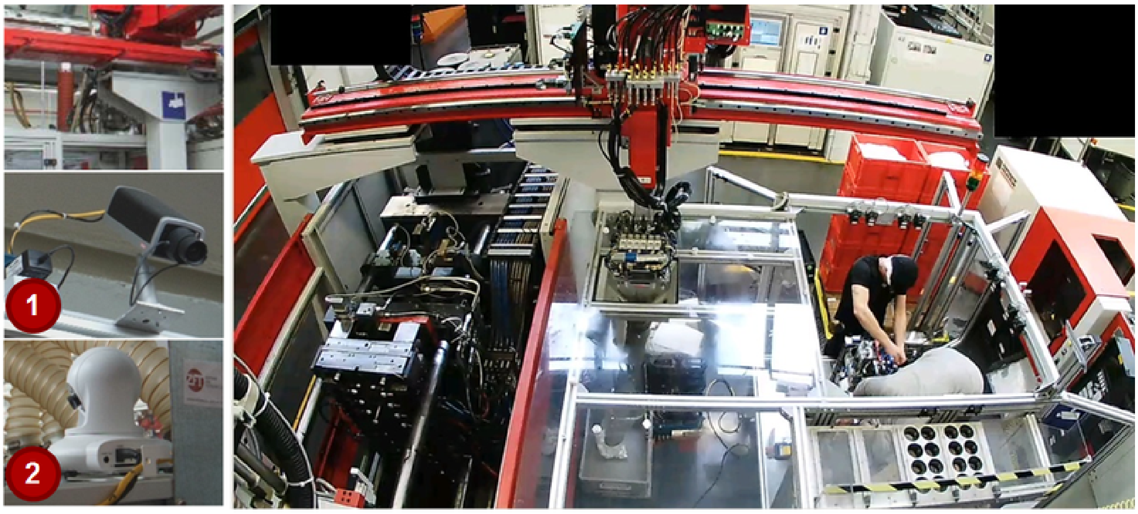


Figure 6.6: *Photos of the cameras (left) and camera perspective of camera 1*

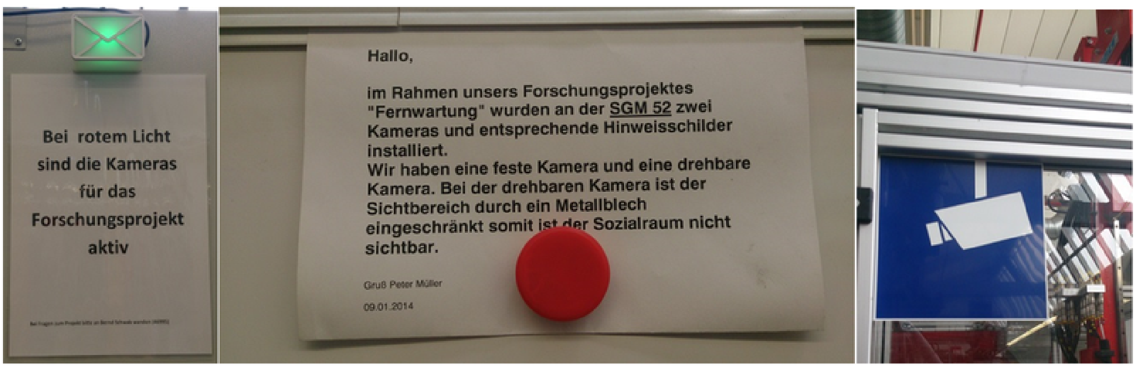


Figure 6.7: *Indicator plates and bulletins for camera surveillance*

and a simple PI controller with two computers. The first computer served as an network emulator and the second modeled sender and receiver with Matlab.

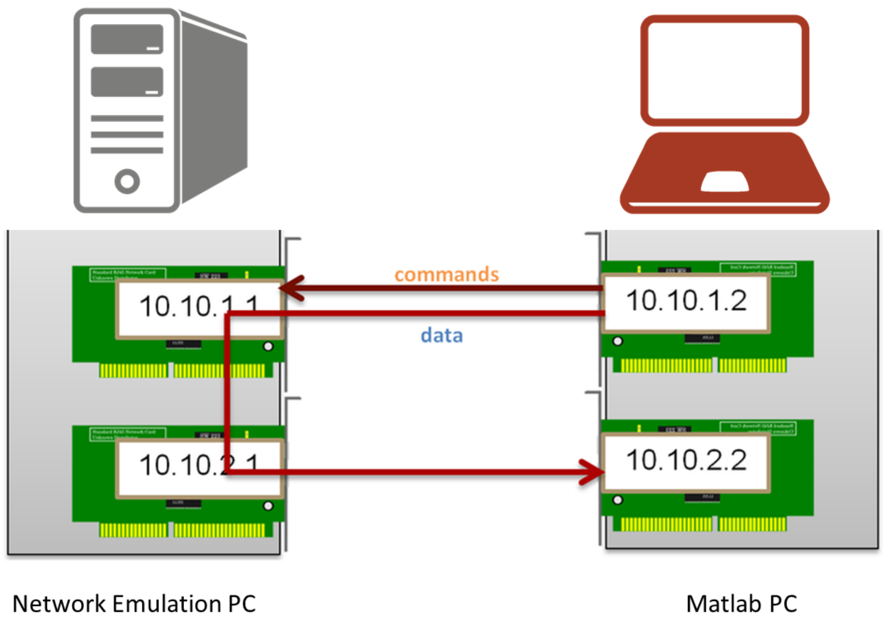


Figure 6.8: *1. Testbed: Control test environment*

2. **Emulator Testbed for virtual manipulator control** (in Figure 6.9)
This testbed was used to demonstrate both sides of the telemaintenance setting with one computer each. In the middle of Figure 6.9, the emulation PC serves as the transmission path. On both computers the Reis Software is used – it simulates a robot and the appropriate interaction devices in software (RobOffice und ProVis).

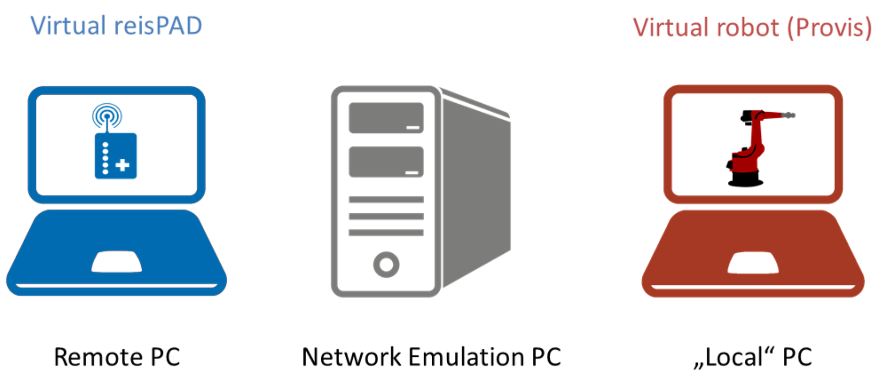


Figure 6.9: *2. Testbed: Emulator testbed for virtual manipulator control*

3. **Local tests with KUKA Industries control system** (in Figure 6.10)
A KUKA Industries mini control system without robot was provided and con-

nected to the ZfT network. It works just like the manipulator control systems, but misses the hardware robot. Several algorithms can be tested with real hardware in that way. Also the teach pendant can be used in hardware. The software visualization of the robot can be used on the monitor.

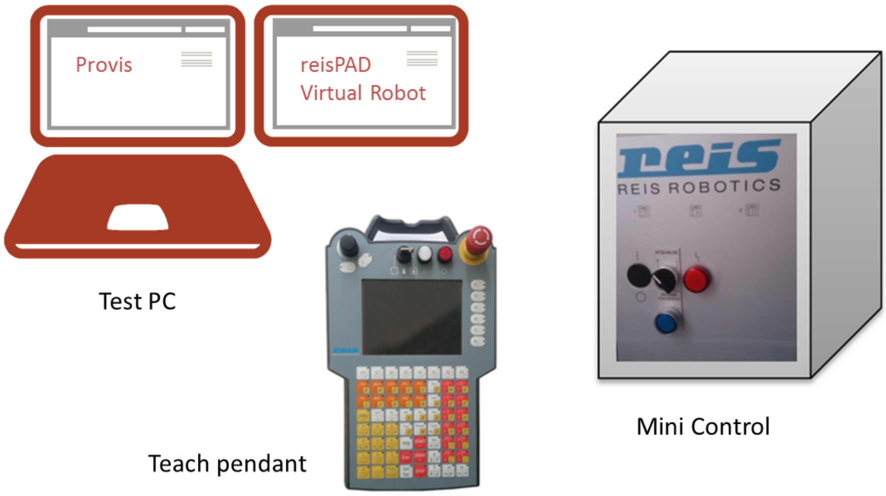


Figure 6.10: *3. Testbed: Local tests with KUKA Industries control system*

4. **Prototype 1** (in Figure 6.12 and 6.11)
   The prototype development for the teleoperation scenario evolved in several steps. After frame conditions were estimated, an offline testbed was installed to enable the testing of all components. As depicted in 6.11, it consists of two computers, one tablet PC and several cameras. A KUKA Industries mini control (without the actual robot) and a robot simulation was provided by KUKA Industries and included into the testbed. The network connection and quality of service changes were emulated with a Linux machine (Ubuntu 12.04 LTS) and scripts using iproute2/netem. With this testbed, we can evaluate the influence of different network conditions and also the influence of other components' network activities, on the individual modules of our system.

5. **Prototype 2** (in Figure 6.13)
   Additionally, a telemaintenance test center was built in the Zentrum für Telematik. It was used to run different online tests with the real equipment at the facility to decide on the final software architecture. As depicted in Figure 6.13, the test center consists of a computer with several monitors and a network connection. Here, the VideoLAN Client (VLC) was used for video transmission, in the final prototype ffmpeg / ffserver was used for that task.

The offline testbed and the test center ran through several cycles of iterative improvements, as findings from the first one were implemented in the second and vice versa. This matches the rapid prototype approach with several small iterations that

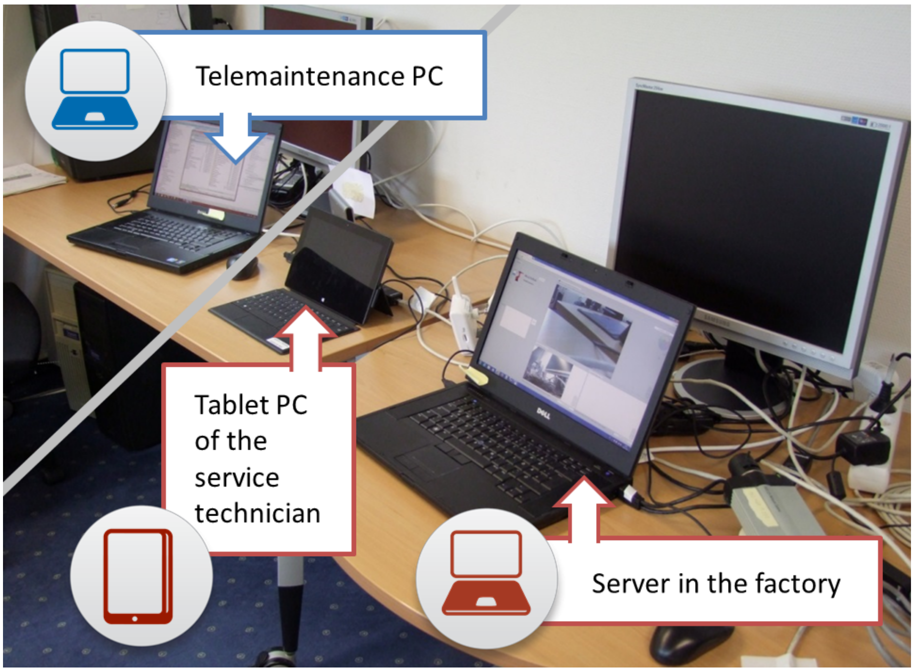


Figure 6.11: *Prototype 1: Testbed structure with network emulation (photo)*

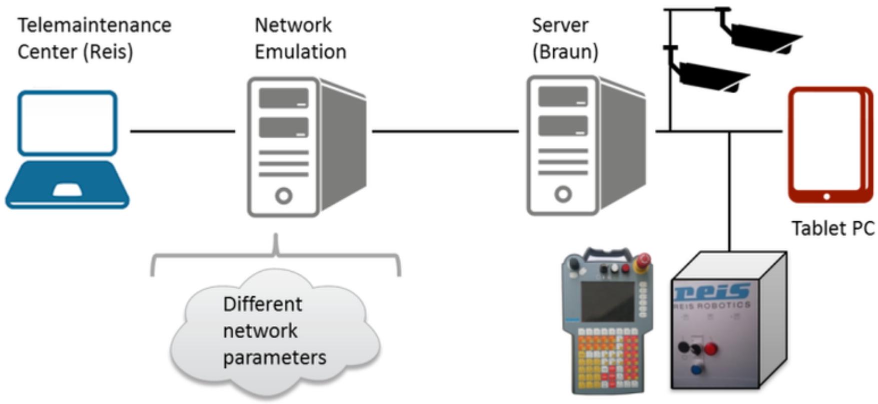


Figure 6.12: *Prototype 1: Testbed structure with network emulation*

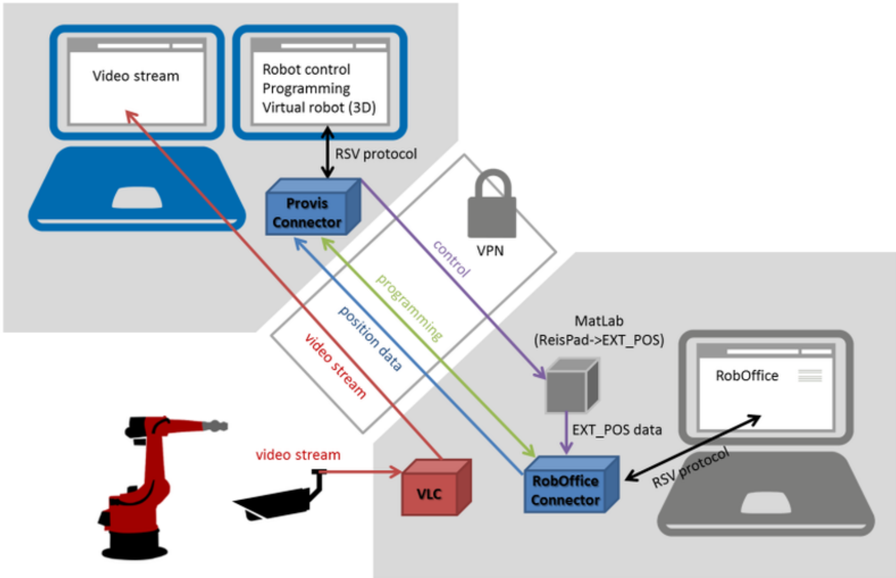


Figure 6.13: *Prototype 2: Teleoperation testbed with different data streams*

had proven useful for research projects covering the integration of several hardware components and the development of new software.

6.2.2.4 First test in an industrial environment

Based on the test maintenance center which has been installed at the Zentrum für Telematik, similar equipment was installed at KUKA Industries after its final development. The goal was to perform a teleoperation scenario, where the expert at KUKA Industries helps the service technician at Braun P&G to solve a problem with the robot.

A modified robot firmware version, provided by KUKA Industries, was used for both the test center and the final prototype. This firmware enables robot control by a device which is not physically connected to the robot. We needed to clarify legal issues about possible damage caused by teleoperation and the use of a modified firmware, before any tests could take place. Since May 2014, remote access to the robot has been possible on a fixed date every week. The plant must be taken out of production for this period and a service technician had to prepare the robot and needed to be near the robot for safety reasons and for interaction tests.

In order to validate the usability of our framework and the software built upon it, we carried out several tests: We evaluated different features in experimental testbeds and tested the applications' functionality and usability in weekly experiments with service technicians at the production site. In addition, short tests of the teleoperation components were carried out biweekly during the last year. The current prototype implementation of the AMS software was evaluated in a milestone test during October 2014 including all project partners. For this test, the service center was set up at KUKA Industries in Obernburg, Germany. It consisted of a computer

Figure 6.14: *Test person with mobile device following the expert's instructions in front of the switch cabinet*

(Dell Latitude E6410) with an additional monitor and the AMS software, a mobile phone and a robot programming device. The facility to be serviced was the environment of an injection molding machine producing parts for electric toothbrushes, in the factory of P&G in Marktheidenfeld, Germany. The test facility consists of the injection molding machine, an assembly in which membranes are added to the molded parts, and a KUKA Industries Cartesian robot handling the parts. In addition, the facility was equipped with a computer, two high definition IP cameras and as well as a mobile device (Microsoft Surface Pro tablet) with built-in camera to be used by the service technician. As the facility's network is located inside the worldwide P&G virtual private network (VPN), the center was connected to the P&G VPN by a host-to-site VPN via the P&G VPN gateway in Brussels. The goal was to test whether the implemented software is adequate for use in a teleoperation scenario with the expert performing several tasks remotely to guide the service technician in order to identify and solve a problem concerning the robot. In addition, the performance of the system to adapt to bandwidth changes was evaluated.

In the initial setup, three video streams were transferred to the expert: The first provided an overview of the facility, the second stream was capturing the details of the assembly machine, while the third stream was captured by the camera of the service technician's mobile device. After connecting, the expert was able to control and program the robot, like normally done over a local area network. The expert was able to steer the robot remotely using video and Provis' AR feedback, while the local technician stood by, pressing the dead man's switch. Subsequently, the local service technician, as well as one participating scientist performed multiple tasks under expert supervision. E.g., "Go to the control cabinet and take a picture of the voltage information in the display of a certain component", as shown in Figure 6.14.

To facilitate communication between the participants, both a chat functionality and a screen shot and paint functionality (covered in detail in Section 6.3.3) were available in the GUI. After these first tests, we triggered the bandwidth adaption routines with false inputs, to emulate a deterioration of the end-to-end connection quality. Subsequently, the system adapted the video quality several times during the maintenance session, to prevent impairment of the higher priority services. While the adaption was noticed by the users, in most cases it was tolerable and work could continue.

The experts and technicians were content with the features offered by the prototype. Both chat and paint features were considered very helpful, due to the fact, that the service technician works in a very noisy environment. Also, guiding the technician based on the video from the mobile device ("not that display!"), proved helpful. They expect the completed system to be very helpful tool in future telemaintenance use-cases.

6.2.3 Timing

6.2.3.1 Network delay

The setup of a stable and secure communication link between the telemaintenance center at KUKA Industries (or for tests at the ZfT) and the plant at P&G is essential for the teleoperation scenario. It requires several organizational and technical steps to reach this goal.

The first step was the constitution of the Virtual Private Network (VPN) access to the company intranet and the implementation of the security measures requested by the cyber security team. Now the computer at the telemaintenance center on the left side of Figure 6.4 can establish a tunnel to the P&G VPN Gateway (middle). Within this tunnel, a second VPN tunnel is embedded to enable the secure end-to-end access to the test computer at the production site (on the right side). This device was provided by P&G and equipped with the ZfT software. It is connected to the robot control, the cameras and other local hardware.

The second step was the characterization of the connection which leads through several subnets. For measurement purposes we use a server outside the ZfT research facility network to omit internal traffic influences.

Figure 6.15 depicts the simplified network structure together with the average base delay, which was determined by multiple ping requests during a longtime measurement of 3.5 million requests over a period of 4.5 days. These simple measurements helped to define boundaries for the sampling time of an autonomous control process, which only requires little bandwidth.

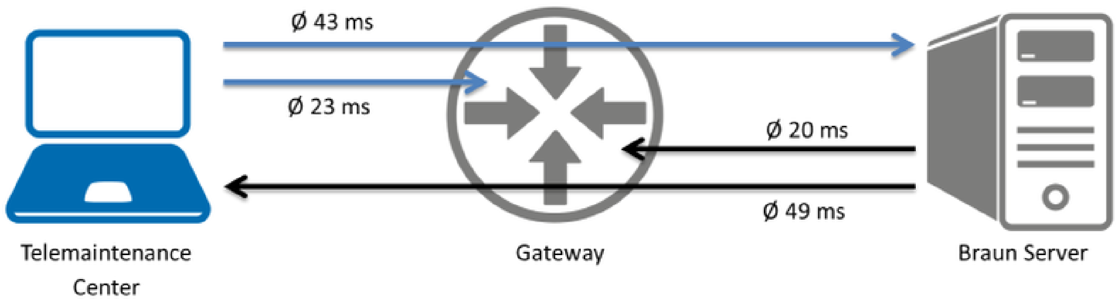


Figure 6.15: *Abstract network structure with ping delay*

When it comes to larger data sets, especially for the video streams, the overall bandwidth available during transmission is important. As all communication to the P&G Europe network is routed through the same gateway, the quality of service within our VPN tunnels varies with the combined network load and time of day, but almost irrespectively from our test traffic.

To obtain information about the Quality of Service (QoS) within our telemaintenance VPN tunnels, we used the Distributed Internet Traffic Generator (D-ITG) to generate the traffic for sets of measurements with different packet sizes, packet rates and transport protocols. While using a standard $16MBit/s$ downstream, $1MBit/s$ upstream asymmetric digital subscriber line (ADSL) connection at the service center, the available up- and downstream bandwidths are highly asymmetric between the facility and center. Hence, we created different test scripts for "upstream" and "downstream" connections in accordance with the known constraints.

As we traverse multiple VPNs, the most important actuator to optimize throughput is to packetize data according to the current maximum transfer units (MTUs) of the VPN components. Multiple tests were directed at automatic discovery of optimal packet sizes. In addition, sending all data over UDP and implementing reliable packet transmission functionalities were needed on the application layer prove useful. Furthermore, we created test streams that resemble the data streams generated by our telemaintenance applications. Their behavior is shown in Figures 6.16 to 6.18 and discussed in the following paragraph.

Figure 6.16 depicts the best-case one-way facility-to-telemaintenance-center delay of TCP and UDP test streams sharing the VPN connection. As one can see, the TCP traffic (light blue) suffers from up to six times higher delay than the average of the UDP streams. This is on the one hand due to occasionally occurring packet losses causing TCP retransmissions and on the other hand due to TCPs down-regulation on shared connections.

Figure 6.17 and Figure 6.18 depict the worst case within our downstream measurements, when bursts of packets are delayed during transmission.

While the packets of the sparse TCP stream (light blue line) were delayed but retransmitted, the UDP streams (all other flows) suffered packet losses in proportion to the number of transmitted packets. These losses are depicted in the upper part of Figure 6.18. The periods of increased loss and delay occurred on all UDP streams

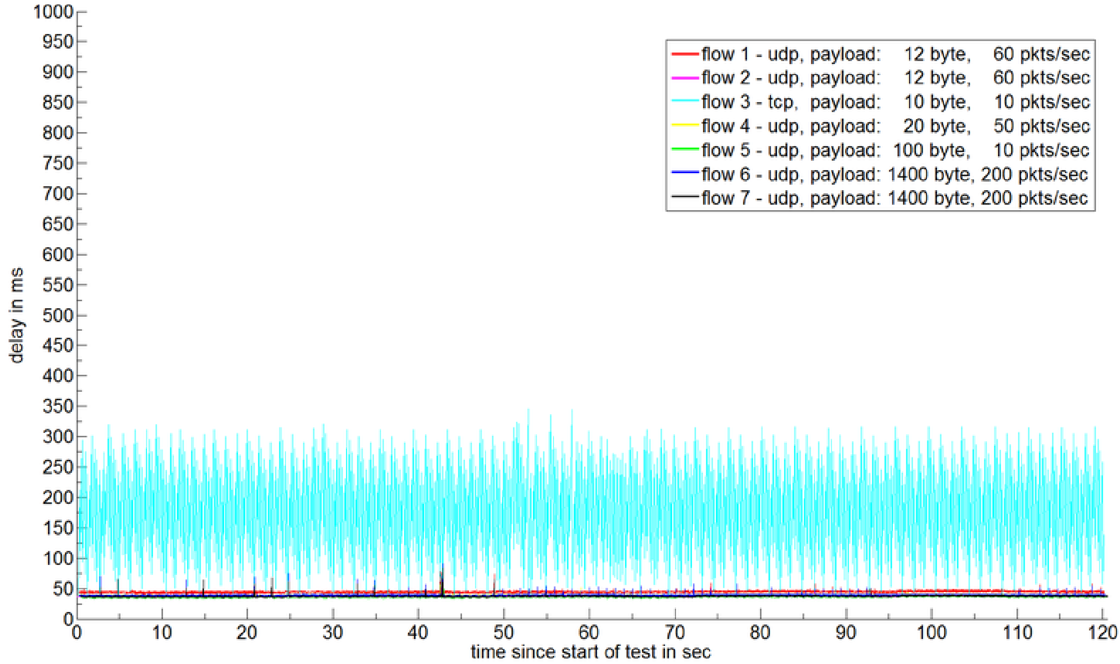


Figure 6.16: *UDP and TCP delay (best case)*

in bursts, which lasted up to 2 seconds.

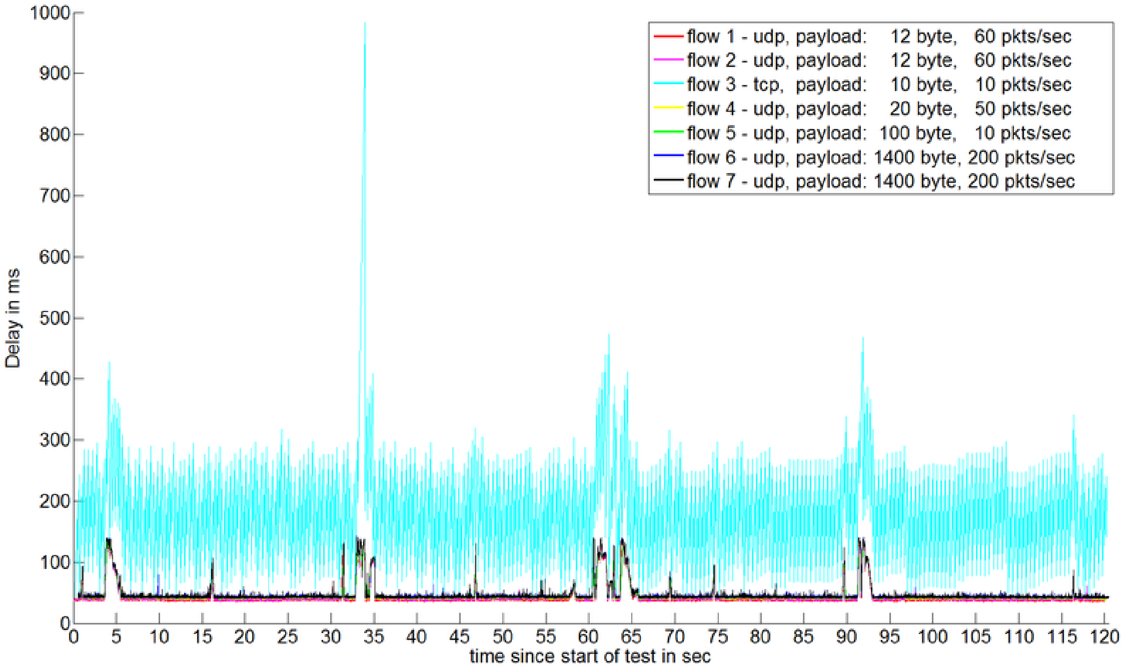


Figure 6.17: *Delay of TCP and UDP test streams (worst case)*

If the measurement system meters values like these, no teleoperation of the robot

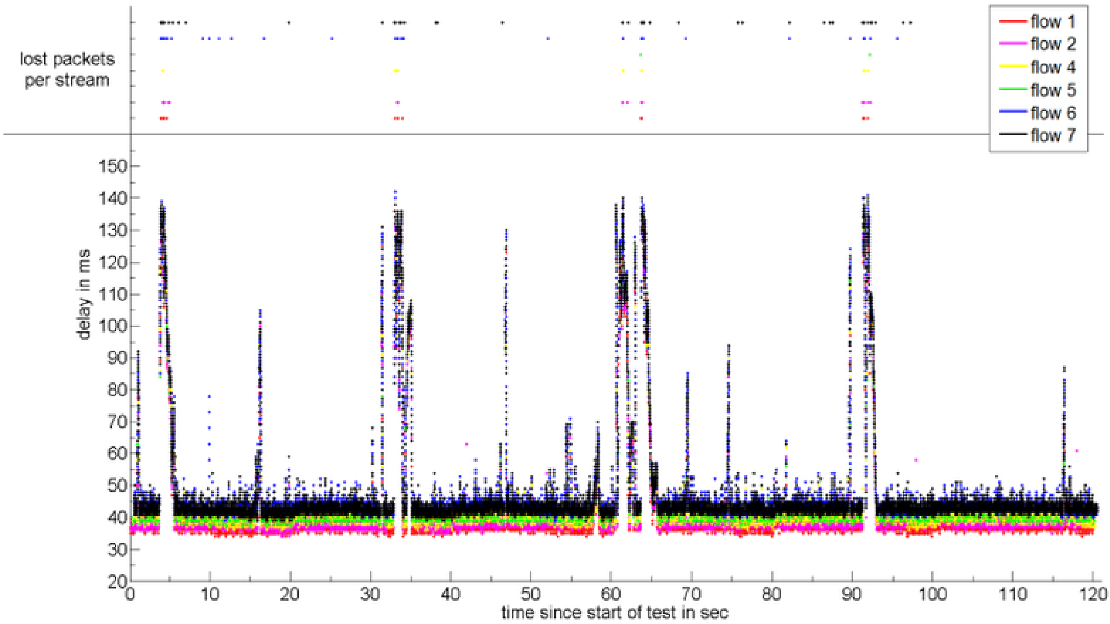


Figure 6.18: *UDP delay (worst case) with losses*

is permitted over this connection, as neither the secure transmission of steering commands nor continuous video feedback can be provided.

6.2.3.2 Human delay

To be able to estimate the time domains later, the "human delay" needs to be considered. Humans are known to be much slower than machines. The "simple" reaction time on a button click task is around $215ms$ [193], but the mean of a human response delay in supervisory control can be much longer ($5s$ to $16s$) and depends on the task [194].

6.2.3.3 Time domains

As a summary of the previous sections, several time domains need to be considered in a telemaintenance scenario, which are depicted in Figure 6.19. As the teleoperation task covers each time domains, they need to be addressed separately in the final architecture.

- Humans can only interfere in any process with the delay of their reaction time, that means with a delay greater than one second.
- When we consider autonomous functions taking place on the telemaintenance center, we need to take into account the measured round trip time, that means the delay of approximately $100ms$.

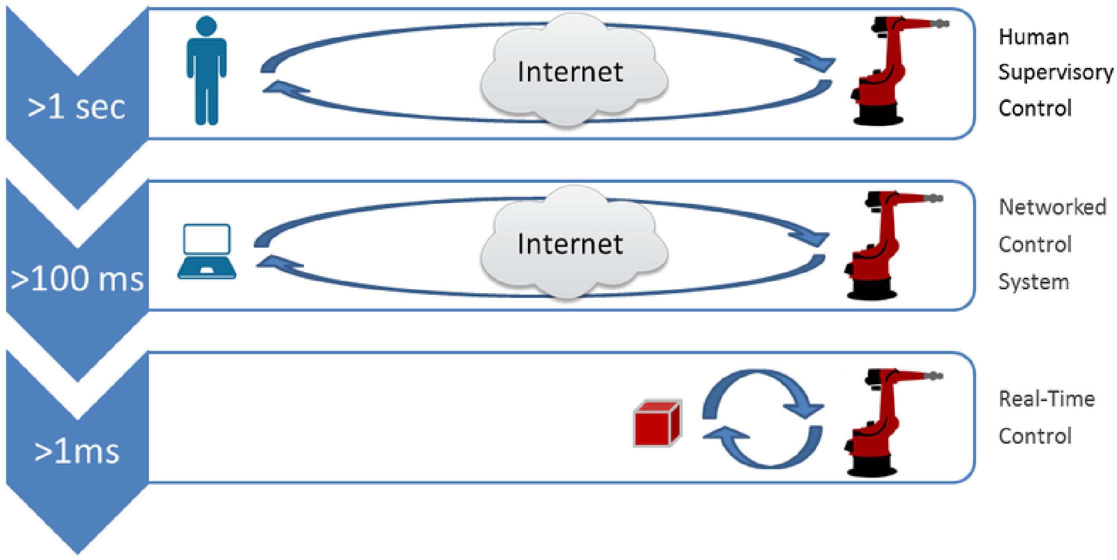


Figure 6.19: *Time domains for control*

- In the robot control environment of the KUKA Industries robot, real-time processes take place that need a delay below $1ms$. The robot cycle time is $10ms$, that means we can get robot values (like speed, position, etc.) e.g. for the condition monitoring in that rate.

As a result of preliminary studies, the time goals depicted in Figure 6.20 have been set for the different parts of the telemaintenance infrastructure. The estimated network delay was used just as human factors and known delays of the hardware. As only the entire delay can be measured properly, the values indicated in Figure 6.20 are estimations based on values presented in the literature and several solitary measurements.

6.2.4 Adaptive Management and Security System (AMS)

As it was described in the previous sections, telemaintenance needs a special combination of services, requiring different bandwidth and sampling rates. As those requirements influence each other and the services need to be included with various security measures, the optimization of the throughput is a research challenge. The services also have different priorities: The transmission of robot commands must not ever be blocked by a bandwidth-consuming video stream.

Therefore, we proposed the "Adaptive Management and Security System" (AMS) in [15], which is depicted in Figure 6.22. Briefly summarized, the AMS permanently measures the quality of the communication link during the telemaintenance scenario and estimates the optimal configuration for the required services. As already described above, there is a chance for a rapid quality decline and bottlenecks in the network. After this has been discovered by measurement units, the AMS decides,

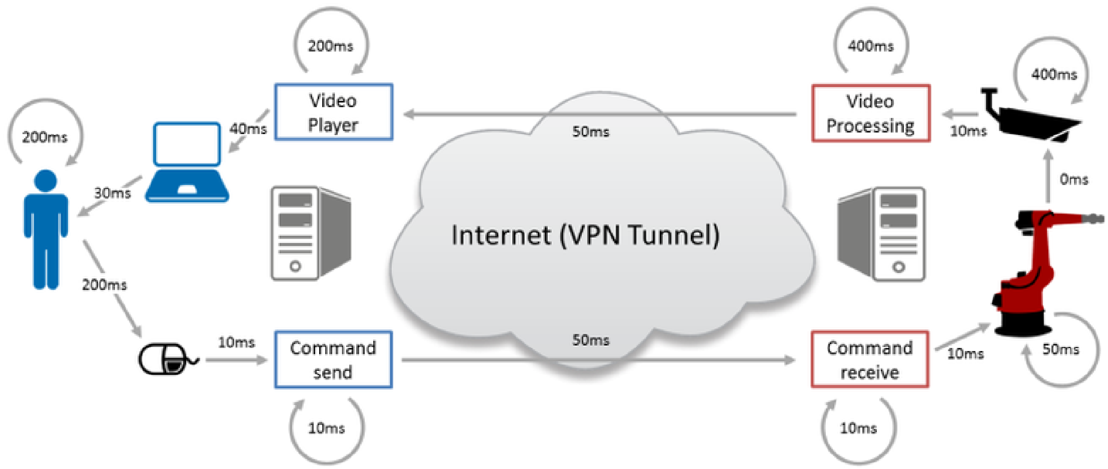


Figure 6.20: *Key timing goals for the prototype*

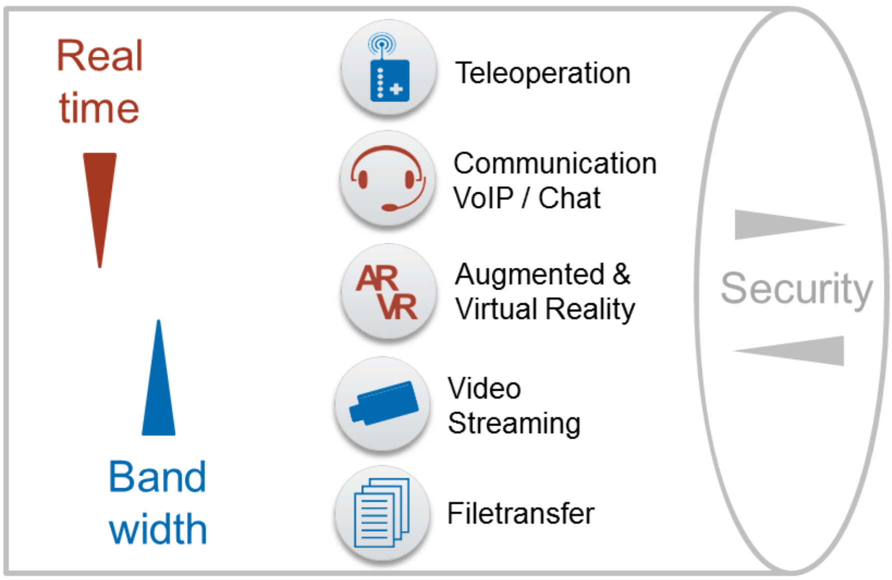


Figure 6.21: *Requirements of the AMS Services in the Security Tunnel*

which service configuration fits best for the altered network state. The AMS can decrease the video stream quality with graceful degradation measures or even stop the transmission of some of the video streams to ensure a safe transmission of the robot commands and the primary video stream which serves as a feedback to the expert. Additionally, semi-autonomous helper functions are triggered dependent on the connection quality, to support the teleoperation. For example the maximum velocity of the drive commands is decreased before dangerous situations arise. In case of an extreme decline of the connection quality, the transmission of external robot commands is interrupted completely and the robot is autonomously able to drive to a safe position.

All described features are performed autonomously by the AMS. As mentioned earlier, there is always a service technician near the robot, who is informed about the connection quality on his mobile device and can interrupt the teleoperation if he or she thinks it is necessary.

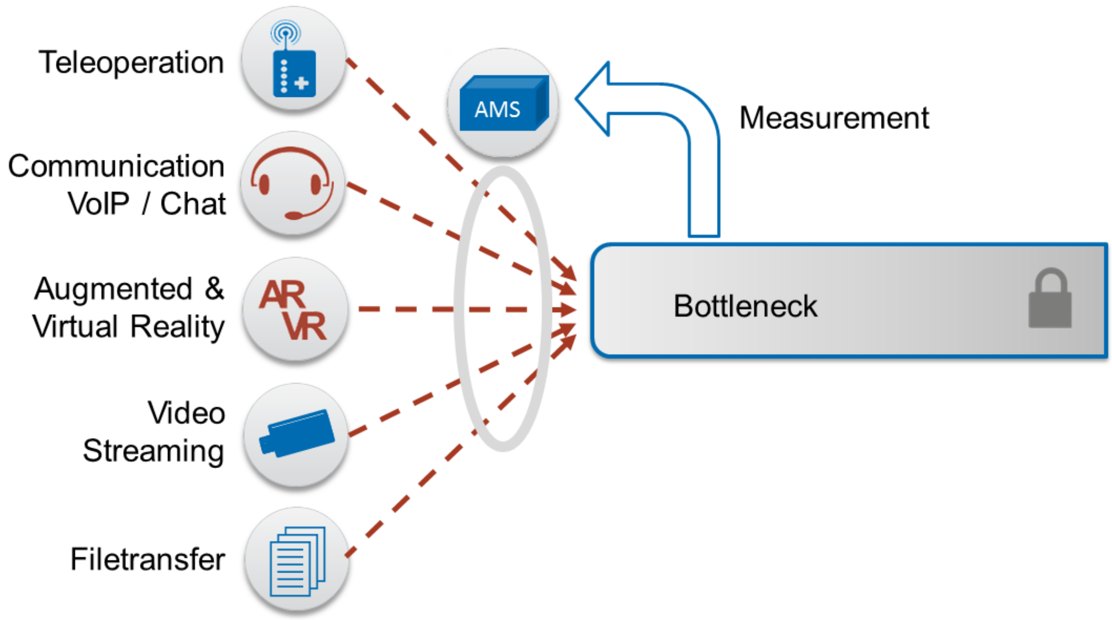


Figure 6.22: *Basic concept of the Adaptive Management and Security System (AMS)*

6.2.4.1 Overview

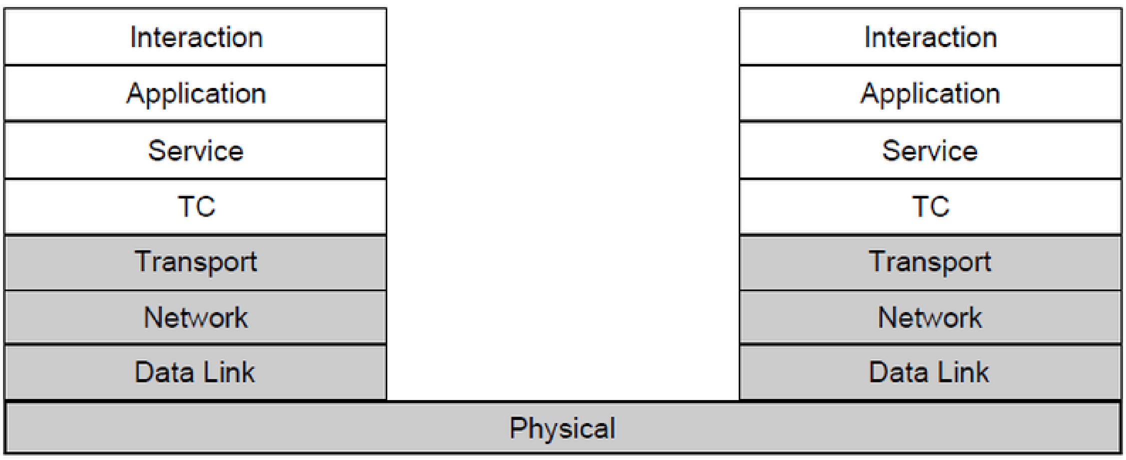


Figure 6.23: *Layers of the AMS architecture on top of OSI layer four*

The Adaptive Management and Security System (AMS) is designed as a modular multi-layer architecture, with each layer providing functions for the layer above. The layers contain a main manager component that exchanges control data with the managers of adjacent layers. In the following sections, we explain these layers top-down, as depicted in Figure 6.23. Interaction with the system is modeled in the InteractionLayer, on which we specify the work flows to be supported. This layer and the ApplicationLayer, which describes the user interface, will be covered later in Section 6.3. Here we cover the infrastructure, consisting of the ServiceLayer, which provides the underlying functionality for the ApplicationLayer, and the TelematicsConnectionsLayer (TCLayer), which resides, as depicted, above the OSI transport layer. It provides a single entry point into the center or factory for the secured data transmission.

Figure 6.24 visualizes the structure of services and the way services interact with components of their own and adjacent layers. The broad arrows represent data flows while the thin arrows represent control connections. The ApplicationLayer component depicted on both sides relies on functionality or data provided by the service shown in the upper middle of the figure. The service itself may, as depicted, rely on other local services, e.g. for common tasks like time synchronization. Services can communicate with remote services belonging to the other AMS instance by using the TCLayer. In addition to these data streams, all services keep a control connection to the ServiceManager depicted at the bottom of the figure.

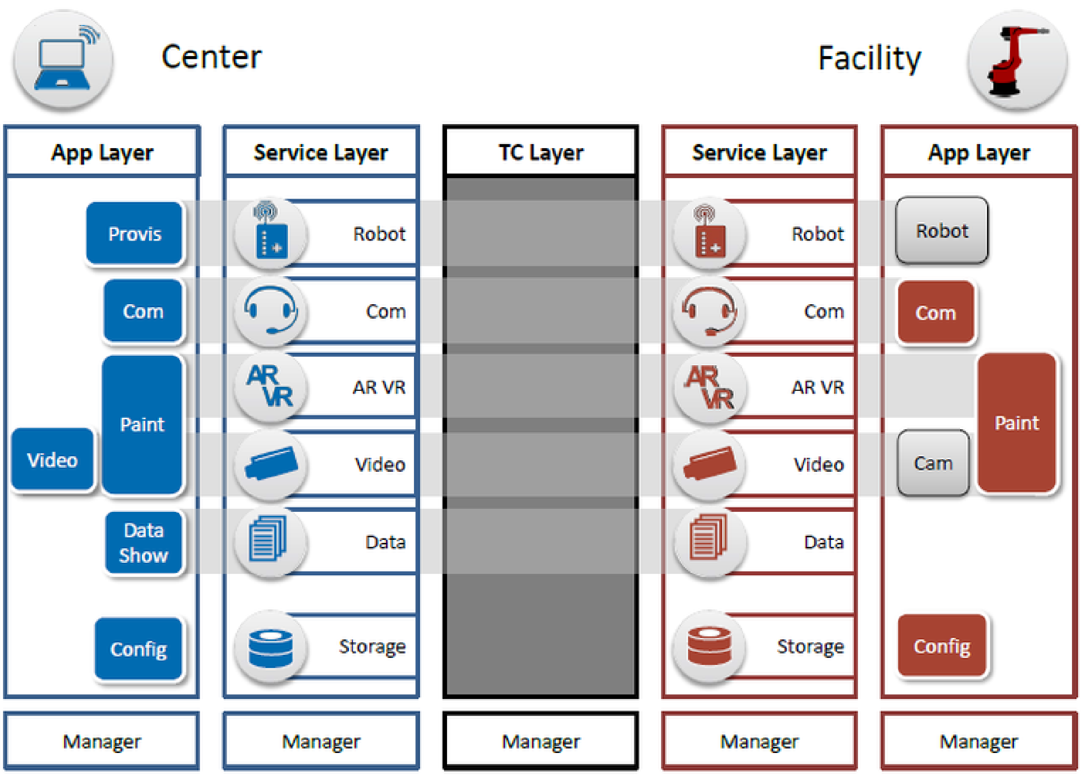


Figure 6.24: *Basic AMS applications and services*

6.2.4.2 Service layer

As depicted in Figure 6.24, each application is facilitated by one or more services: There are for example services for the transfer of video streams, chat functions and synchronized storage of machine data. Another important task fulfilled on the ServiceLayer is to provide access to the hardware, like robots, video cameras and external sensors. Some generic services, like time synchronization or video streaming, are required by multiple other services, while specialized services provide access to vendor-specific software. The ProvisConnector, for example, is a special Robot Service that integrates Reis Robotics' "Provis" robot configuration and programming software [195] into the AMS system. The storage service, combines data from the video service and the robot service, synchronizes this data and saves it. As of now, all services are Java-based. But in general, a service may be any kind of software runnable as process of the underlying operating system.

Services, depicted generally in Figure 6.25, may contain so called routines, in which algorithms or wrappers for external components are implemented, e.g. video streaming software or virtual machines. Both the communication to the ServiceManager and the management of the routines are done by managers within the services. In addition, these manager components are also responsible for adapting the amount of data sent by the services, according to the instructions given by the ServiceManager.

The ServiceManager is the first program to be started on each AMS instance, and the only component which needs to be initialized manually. The ServiceManager is designed to establish a single point of control: One of the ServiceManager instances, typically the one at the telemaintenance service center, is the master, which provides the configuration for all other AMS instances. The services are automatically started by the ServiceManger, and receive basic start up information via the command line. After starting, the services connect to the ServiceManager to get the full configuration and information where they should connect to. For this purpose, each ServiceManager has, at run time, the complete configuration, which is composed of these parts: a list of AMS instances, a list of services on every AMS instance, and configuration information for every service, e.g., the ports used by the services.

6.2.4.3 Telematic connections (TC) layer

The Telematics Connections Layer (TC Layer) is located beneath the Service Layer of the AMS architecture. It provides a secure connection between two or more endpoints, e.g. between a telemaintenance service center and the environment of a machine in a distant facility. This layer works as a stateful firewall and is responsible for admittance and blocking of service connections. In addition, it provides data about the end-to-end QoS of the connection to the higher layers.

Figure 6.26 provides an overview of the data flows: The wide dark grey arrows depict the data originated by and intended for services. This data is sent through the secured tunnel to the AMS instance on the other side. The lower part of the figure visualizes the different kinds of connections in the tunnel: First, there is a

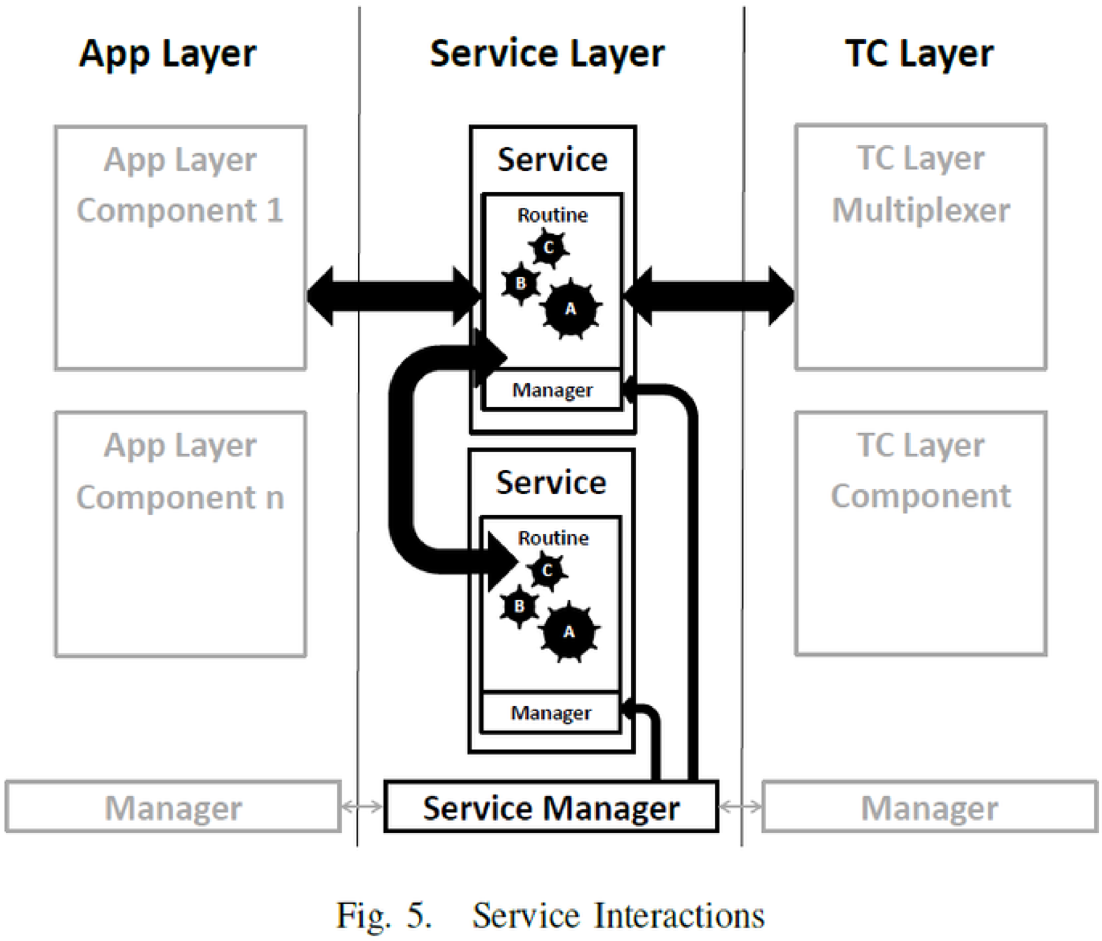


Figure 6.25: *Service Interactions*

control connection, used for information exchange between the managers of the TC Layer instances. Second, there can be one or more ServiceConnections, each of which represents one connection from a local service to a remote one. The third kind of data transmitted is the test traffic generated by end-to-end measurements originated by test routines belonging to the TC Layer. The manager uses data derived from these measurements to dynamically assign each service a fraction of the available network resources. In the following paragraph, we explain the main components of the TC Layer and how they interact in order to provide the main functionalities offered by the TC Layer.

Figure 6.27 provides an overview of the internal structure of the TCLayer, where at the thin arrows represent control data or method calls and the broad arrows service data sent or received through the secured connection. The QoS measurement system consists of a set of components depicted on the left surrounded by the dashed outline. The main component is the "Monitor", a system capable of starting different network test programs, the "Measurements". The Monitor pre-evaluates the outcome of these tests and places the results in the "MeasurementDB". This database contains a

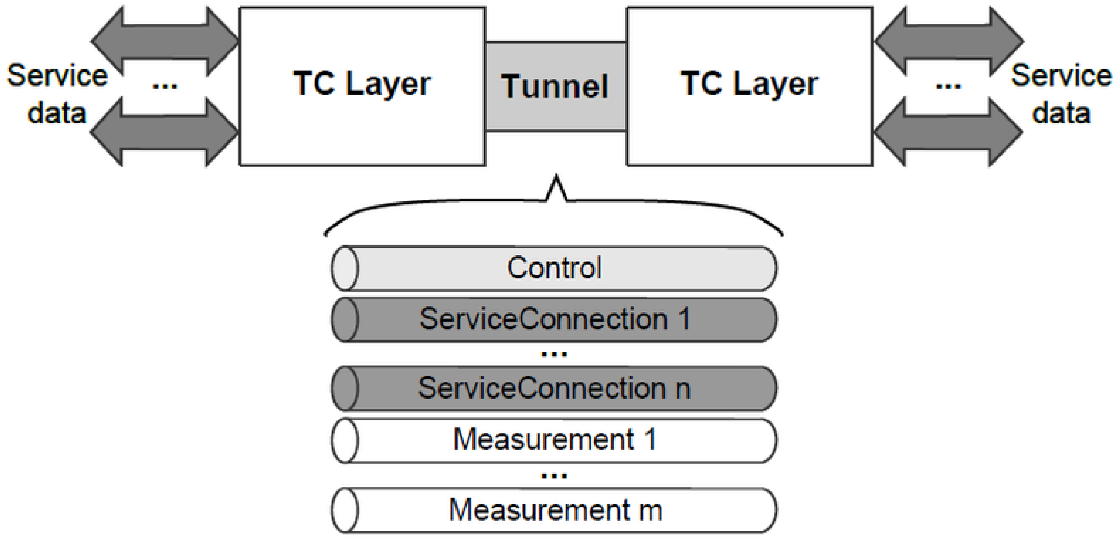


Figure 6.26: *Composition of data streams in the encrypted tunnel*

history of previous measurements for trend analysis and also the data obtained by the Parameter Request module, which regularly collects statistics from the VPN adapter and networking hardware.

The "Manager" is depicted on the bottom of Figure 6.27. This component is responsible for bandwidth assignment and communicates with the Manager of the service layer for configuration purposes. The bandwidth assignment is calculated in the co-located Decider subsystem: The base configuration for the decider is loaded from the "ConfigurationDB", while the decisions are based on data from the MeasurementDB and a set of "ServiceProfiles" containing the constraints and priority of each admissible service. The Manager passes the "Decider"'s results, the decisions to the TC Layer Manager on the other connected AMS system, to the Manger on the Service Layer and to the "Multiplexer". Data about admitted service connections is stored into the "ServicesDB", along with a history of decisions regarding the service.

The Multiplexer is a QoS-aware throughput-limiting stateful firewall implementation guarding the end of the secured connection towards the other AMS instance. The filtering rules, bandwidth assignment decisions and basic instructions regarding priorities and traffic classes are provided by the Manager. These instructions are translated into rules for Linux' Netfilter/Iptables [196], [197] modules and queuing instructions.

The modules – generally comprise a QoS-supporting router and firewall – are depicted within the Multiplexer-box in Figure 6.27. Based on this visualization, we explain how outgoing traffic is processed: The packets sent by the local services are first filtered by the stateful fire-wall, based on Netfilter/Iptables [196], [197]. Here, packets are classified and marked according to their service class. Packets not matching any rule are dropped. The policing is also implemented in the firewall: Traffic flows not matching the policy of their service class, e.g., by exceeding their

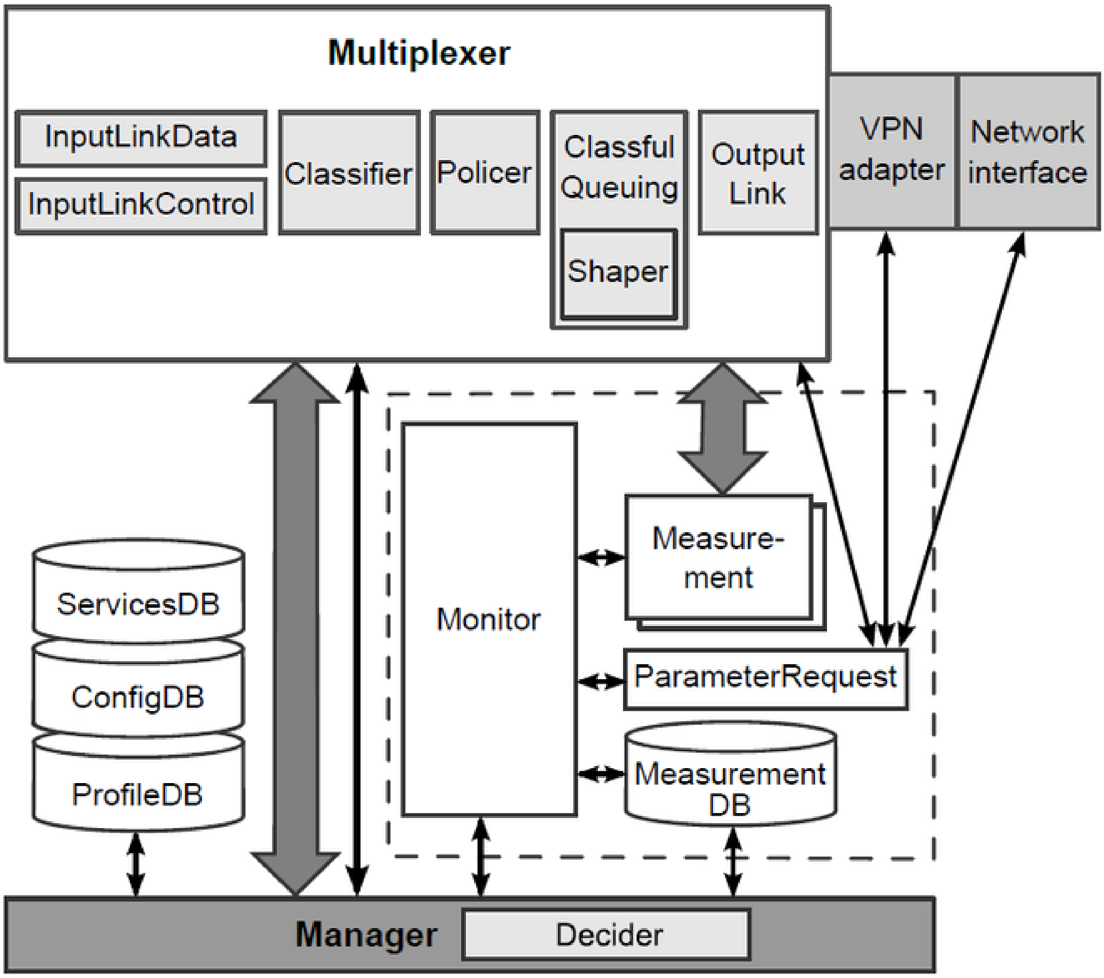


Figure 6.27: *Schematic overview of the Telematics Connections Layer*

bandwidth quota, are marked with an extra "excessive" flag. Depending on the settings for the service, excessive traffic can either be rejected by the firewall or delayed in the next step. The shaping and queuing are implemented by creating a set of queues and queuing disciplines for the traffic control module [198] of the underlying Linux operating system. The priorities of the different traffic classes, are currently realized by priority queuing, based on the marks set by the firewall. In addition, traffic marked as excessive can be artificially delayed (shaping) by sending it through an additional queuing discipline. After traversing the queues, the packets are encapsulated and sent through the secured connection. At the current stage of implementation, the secured connection itself is set up using OpenVPN [199].

6.3 Telemaintenance interfaces and video interaction

6.3.1 Contextual analysis and InteractionLayer

Out of the data from the preliminary interviews (with service employees from KUKA Industries and the service technicians from P&G), the task-interaction perspective of the service technician in Figure 6.28 and of the external expert in Figure 6.29 was created. The demands of the protagonists are depicted in a facilitated decision tree which characterizes the different steps in a telemaintenance process. This enables the developer to derive the required functionalities of the telematics-interface. The interviews clearly showed, that the video-feedback is essential for both sides. It enables the expert to get an overview of the situation and the service technician needs visual guidance to execute so far unknown procedures.

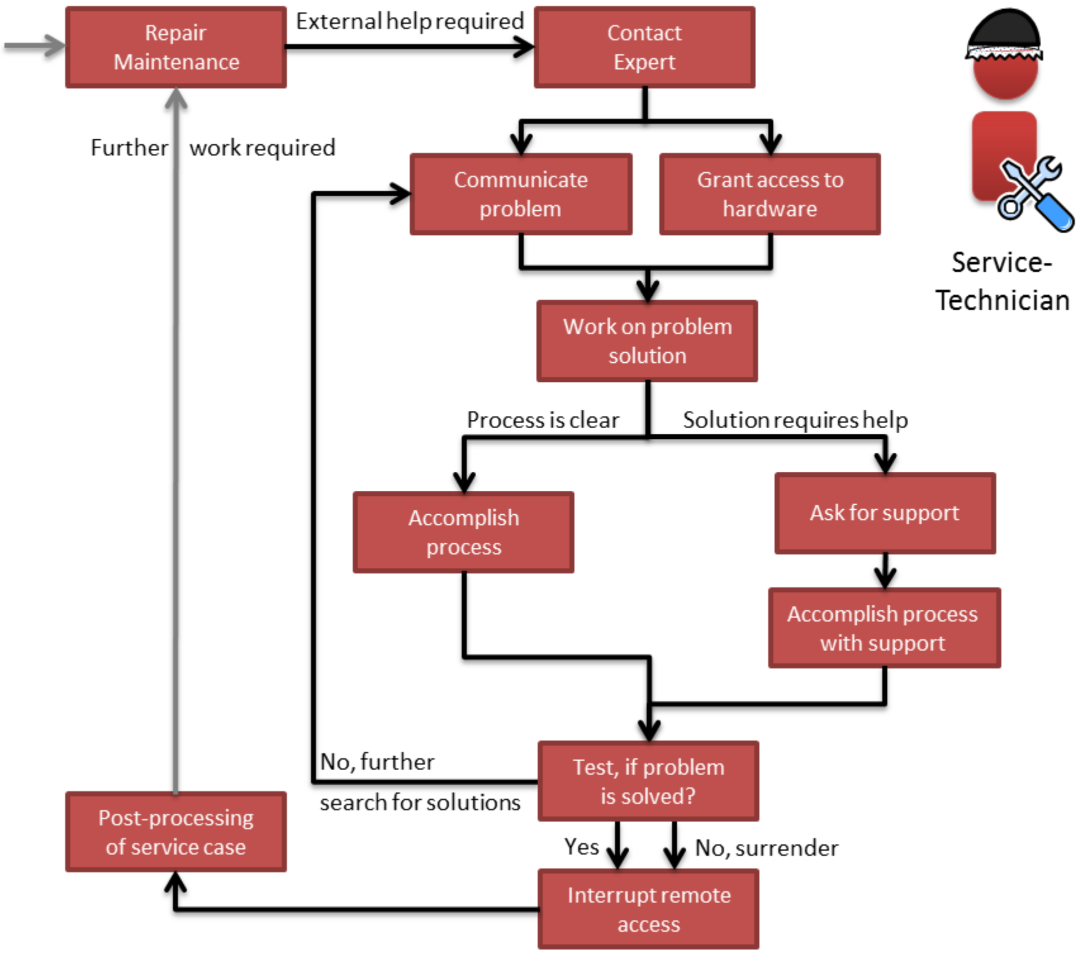


Figure 6.28: *Task interaction diagram out of the perspective of the service technician (left)*

On the InteractionLayer, only the user requirements are considered. The problem

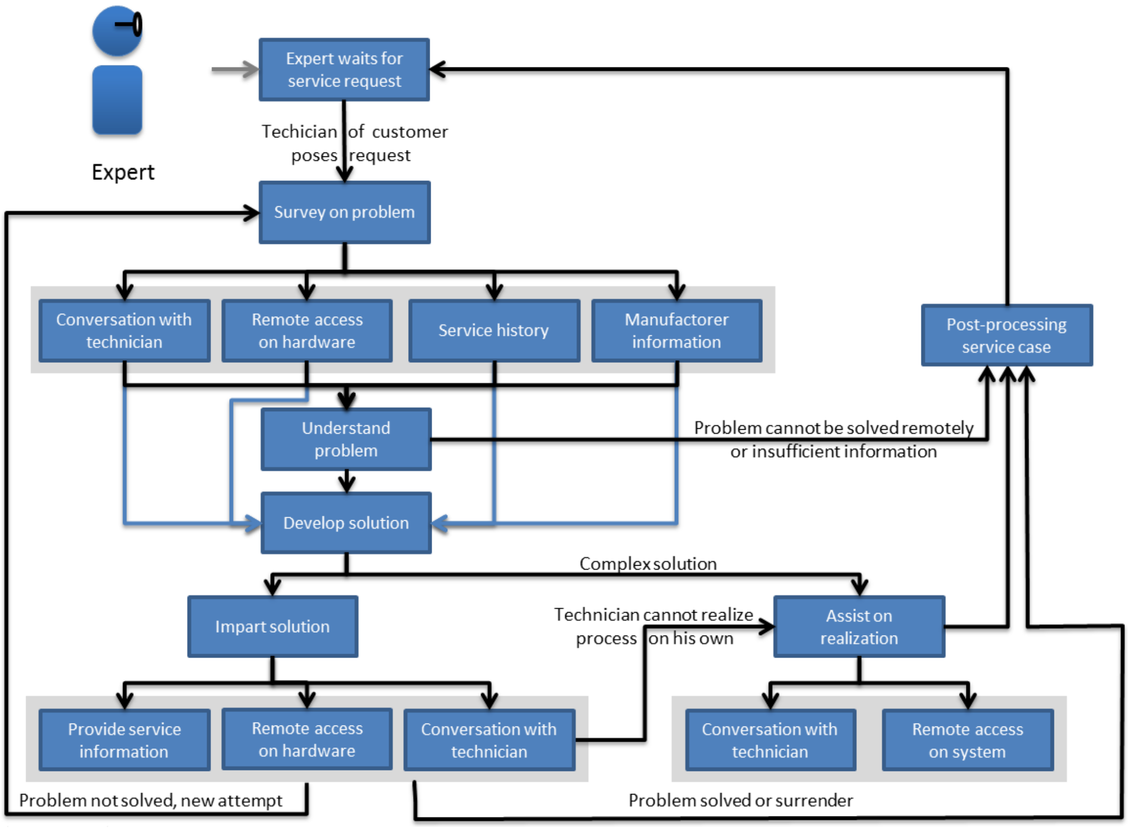


Figure 6.29: *Task interaction diagram out of the perspective of the external expert (right)*

with the plant is very urgent for the service technician. He or she is not able to solve it on his or her own. The whole telematics interface only serves as a tool to help him or her with the problem. The expert on the other side of the connection also has a primary goal: He or she needs to get an overview to the plant and to the current situation. Generally he or she is not informed about the precise configuration of the plant in the facility and tries to understand the problem with several steps. Only then he or she is able to help the service technician to solve the problem.

6.3.2 Implementation and ApplicationLayer

The ApplicationLayer of the telemaintenance scenario is quite complex, as it includes all of the other scenarios. The flow models of the actual and the envisioned processes have been provided in Chapter 3. The interfaces have been developed in an iterative process supported by the various testbed constellations. We also used component models to identify the working environment of the both roles expert and service technician.

6.3.2.1 Expert interface

The final configuration for the expert is shown in Figure 6.30 and consists of three screens, communication equipment, a robot control and a robot programming device. We also assume that the expert might have his or her normal computer desktop on the first screen, to browse manuals, internal specifications etc. and has connection to email and the internet. After the connection to the facility has been established, the configuration of the robot is automatically loaded into the Reis software "Provis" on the second screen. For deeper studies it can also be exported to the hardware robot control. On the third screen, the ZfT software for telemaintenance displays the video streams, as well as information regarding the network quality, a settings dialog, and the core teleoperation functions.

The expert GUI of the ZfT telemaintenance software, as depicted in Figure 6.31, contains a text chat window (on the left). The robot can be controlled with the sliders, on which the current robot coordinates are shown.

Three video views provide insight into the facility. The small traffic light on the left provides information about the current end-to-end Quality of Service between facility and center. It can be changed between tablet camera, the survey of the plant (camera 1 in Figure 6.5) or the montage system (camera 2 in Figure 6.5), while the latter are provided in smaller screens below.

The expert can press the button to switch one of the smaller displayed streams to the main view. The system now tries to accomplish the best quality for this screen. The AMS will automatically regulate the video quality in case of decreasing bandwidth. The expert can talk to the service technician via a telephone connection. The goal of the telemaintenance center is to provide a good situation awareness for the expert.

Figure 6.30: *Component model for remote expert*

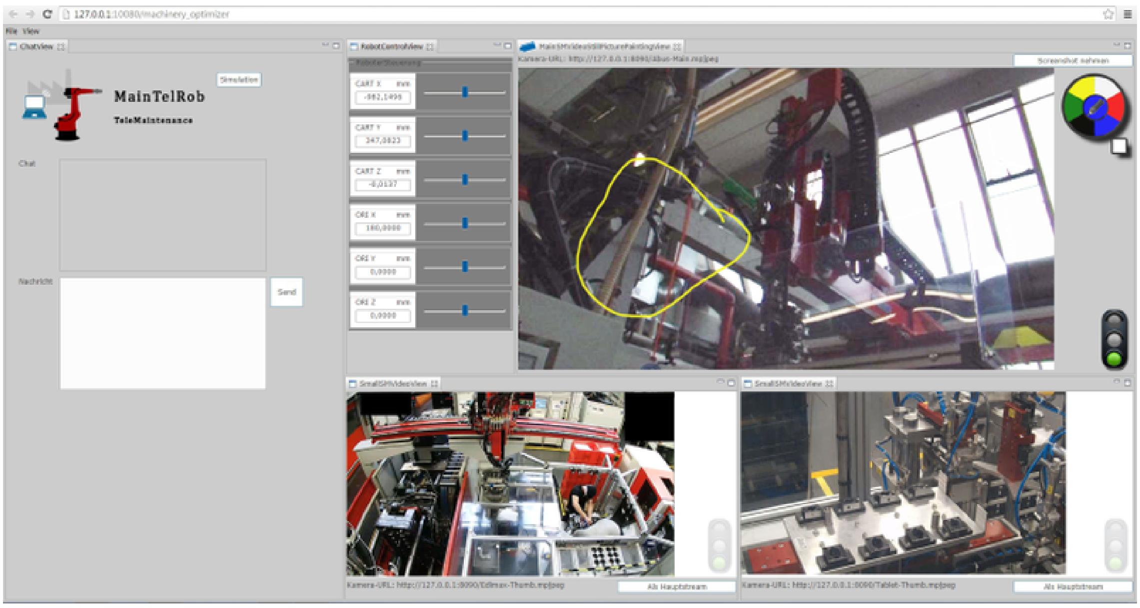


Figure 6.31: *Prototype for remote expert*

6.3.2.2 Service technician interface

The equipment of the service technician needs to be as light as possible. We therefore have integrated as much functionality as possible in the new mobile device depicted in the center of Figure 6.32. In this teleoperation scenario the mobile device serves as communication instance to the robot and provides an additional video stream of a mobile camera.

The service technician at the production site has been provided with a noise-cancelation headset for vocal communication with the expert and a mobile device (shown in Figure 6.32). The user interface provides the video of one camera and live chat functionality. The provided video is selected by the expert and can be used to highlight specific parts in the moving picture or a screenshot (shown in Figure 6.33).

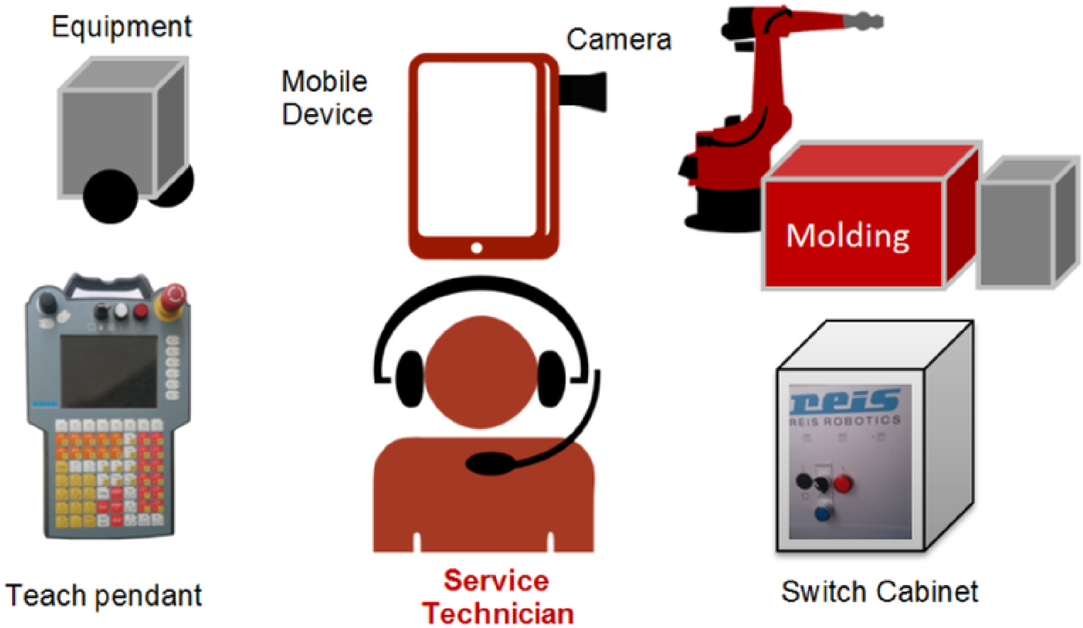


Figure 6.32: *Component model for service technician*

6.3.3 Video interaction

6.3.3.1 Concept

The communication application should provide a possibility for service technicians and experts to build a common base of knowledge and to exchange information about installation. A mobile device is handed to a service technician who has to execute an installation task. This can either be a tablet, a smartphone, or AR glasses. An application is installed on the device for contacting an expert and to transfer information about the installation task at issue to him. The service technician is able to record his working environment with the installed camera in the mobile device.

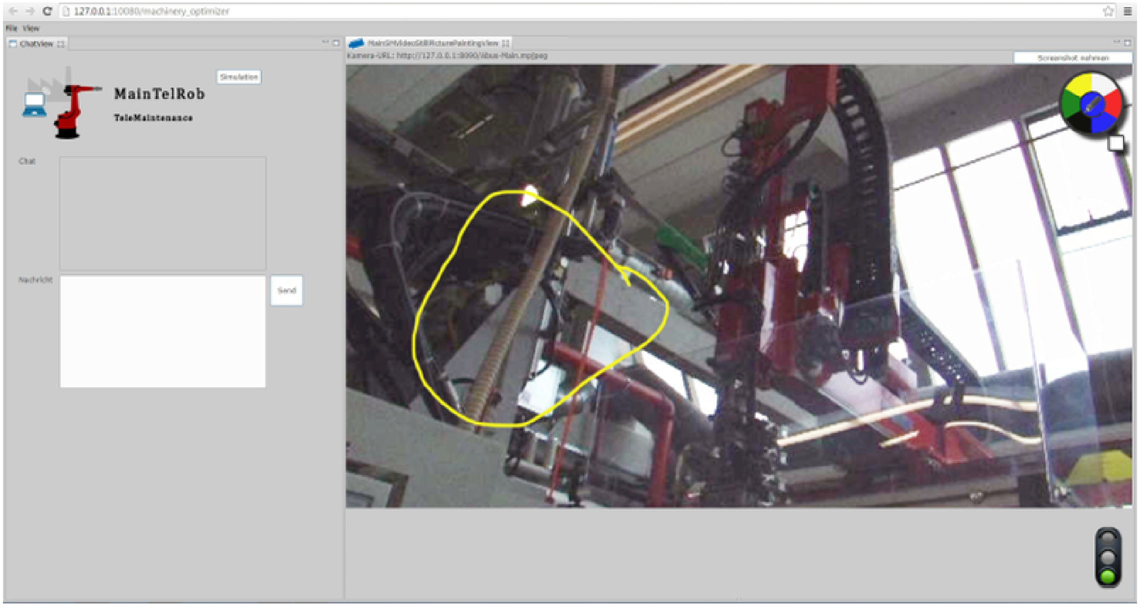


Figure 6.33: *Prototype for service technician*

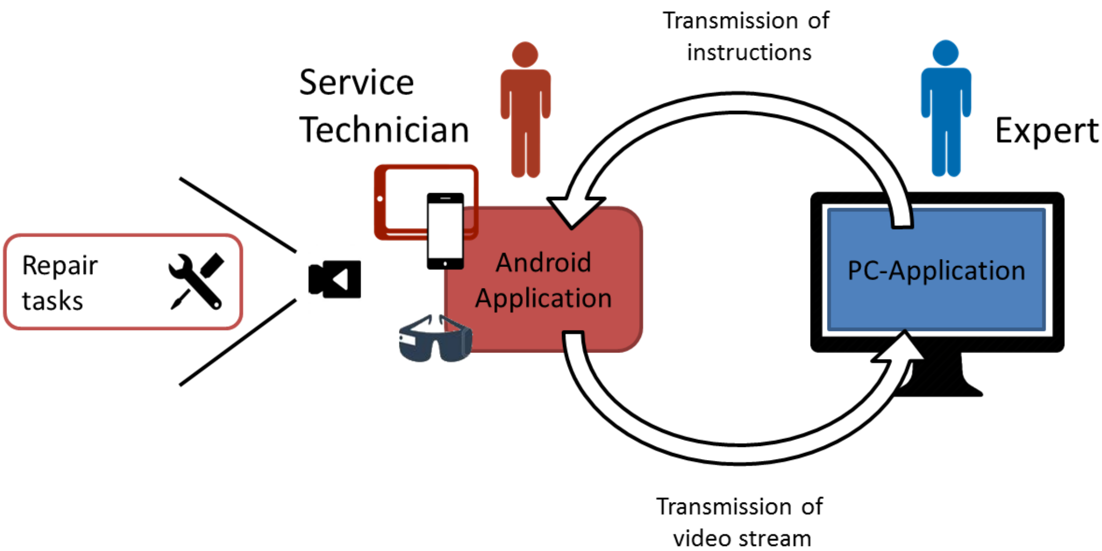


Figure 6.34: *Concept of video interaction*

These recordings then are transferred to the expert's desktop PC via the network. The PC is equipped with a application, which receives and displays the transferred video information. The remote expert can draw installation instructions in the form of lines into the video frame using the mouse. These installation instructions then are sent to the service technician's application via the network and are displayed on his or her mobile device. The display of installation instructions additionally can supported by the use of Computer Vision. Therefore, objects are selected in the video frame and tracked. Tracked objects serve as supporting points for the display of installation instructions. These are shifted relatively to the chosen supporting points and are thereby always displayed at the correct position within the frame.

At first, the implementation of the Android application is specified, which has been asserted in the concept. In the beginning, the basal structure of the Android App is described, that is applied by the service technician on a mobile device. Then a description of the Processing application implememted by the expert on a desktop PC succeeds. In the course of the implementation it is examined, how functionalities of Computer Vision can be integrated best into the system. At last, practical implementations developed for experimental installation operations are listed. The conducted extensions and adjustments in the individual systems are shown in detail for the different experimental conditions. Also the realization in the telemaintenance setup is described

6.3.3.2 Android App

To enable the service technician to send information about the installation task to the expert and to receive installation instructions from the expert, an Android App was developed to be applied on different mobile devices. In Figure 6.35, the basal structure of the application is depicted.

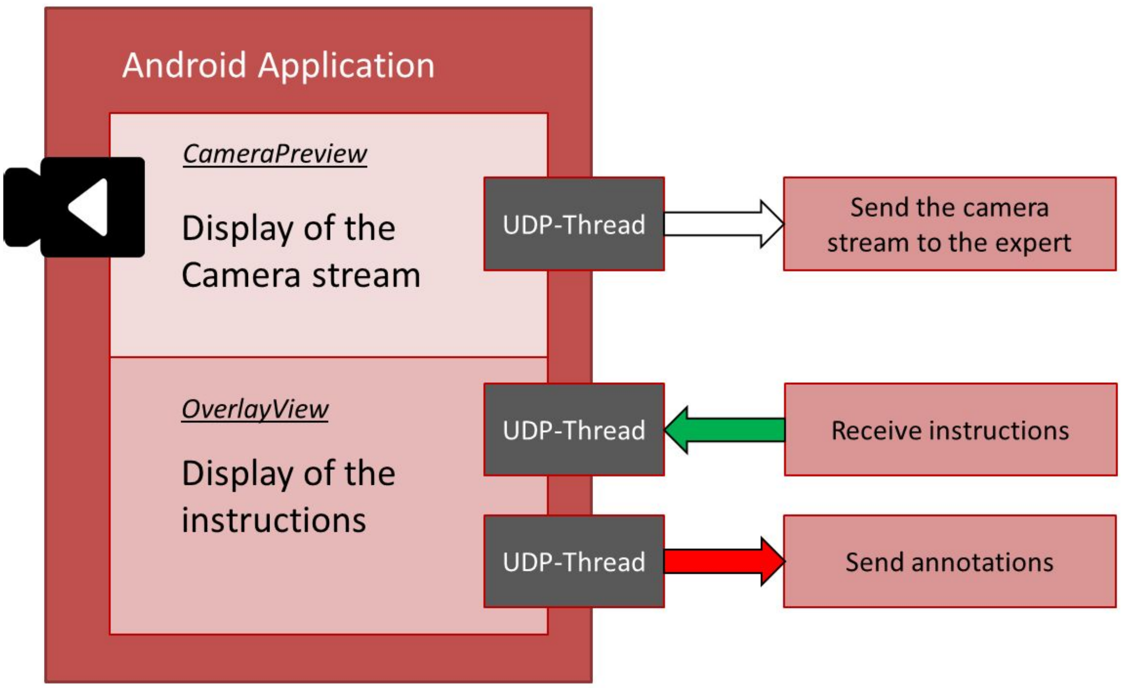


Figure 6.35: *Video interaction realization in the Android application*

In its core the application consists of two Custom Views, which translate the main functionalities of the application. To guarantee the availability of the camera and network functionalities in the application, the Manifest file has to be extended with some entries.

CameraView

`CameraView` is responsible for the display of camera information from the used mobile device. After initializing the View in the `onCreate()`-function of the main class, the `SurfaceHolder` of the class is given a `CameraPreview()`-Object as a callback. The `CameraPreview` Objekt is responsible for picking up camera information and for displaying it on the according View. Further, `CameraPreview` is in charge of transferring the camera frames via the network to the Processing application of the expert. During initialization of `CameraPreview`, parameters for the camera object are transferred to the Constructor, defining which resolution the camera picture should have. When starting the application, a new camera object is initialized with the transferred parameters by the `surfaceChanged()`-function. At last, recording of the camera picture is started via `startPreview()`. In the `surfaceCreated()`-function the camera output is connected to the `SurfaceHolder` and the callback is initiated.

Furthermore, parameters for network communication in the form of `InetAddress remote_addr, int serverport, DatagramSocket ds` are transferred to the constructor of `CameraPreview()`. The `onPreviewFrame()`-function of the class is selected each time, when a new picture is provided by the camera. When a new picture is available, its size is reduced by the method `convertYuvToJpeg()`. The camera picture being provided as `ByteArray`, is first converted into a `YuvImage` converted and then comprized as a jpg and saved as a `ByteArrayOutputStream`. The resulting `ByteArray` is small enough to be sent by UDP over a local network without further division processes. Sending is then executed by `udpSocket.send`.

OverlayView

To display installation information, an `OverlayView` is generated. An `OverlayView` saves all incoming and generated installation information, and is responsible for receiving and sending it via the network.

To fit the plotted and received installation information to the video picture, the `OverlayView` has to lay hierarchical above the camera picture. This is realized in the constructor of `OverlayView` using the order `setZOrderMediaOverlay(true)`. Within the constructor, the parameters for displaying the installation information as well as the threads for network communication are initiated. To ensure visibility of the camera picture through the `OverlayView`, the `SurfaceHolder` of the Views is shown transparently using the order `setFormat(PixelFormat.TRANSLUCENT)`.

Using the extension of `OverlayView` by `SurfaceView`, input via touch can be processed in the class. If a touch on the monitor is registered, an array with the name `pointsDrawing` is set, saving the points of the touch interaction. During the duration of the interaction, touch points are saved in the array. If the touch interaction is finished, the content of the `pointsDrawing` array is added to the `linesDrawFinished` array. `pointsDrawing` thereby saves the points being drawn during the touch interaction , and `linesDrawFinished` saves all lines being finished by the touch interaction. Moreover, `OverlayView` has the same variables for saving incoming installation information. In an array being called `pointsIncomming`, all points of the line being currently drawn by the expert are saved, whereas in `linesIncommingFinished` all finished lines are saved, being already drawn by the expert. All existing installation information exhibiting `OverlayView` is permanently prompted and displayed on the monitor.

The discrimination between the two states of line drawing – if there is a line drawn at the moment, or if its drawing already is concluded – plays a pivotal role for object tracking. To display transferred installation information in an object-related manner, techniques of Computer Vision are employed to fit the position of the lines. This deviation should only concern lines already been drawn. If also the lines in the course of their generation process were concerned, single drawn points could be distorted, and the information about the plotted installation instructions was lost. Two threads are responsible for exchanging installation information for the communication with the Processing application of the expert.

The `DrawingThread` is in charge of receiving and processing installation information coming from the expert. A `String` file is received via a `DatagramPacket`, serving for the display of installation information. The received String files represent a transferred point as well as its state and they have the following structure: `"x-coordinate"+" "+"y-coordinate"+" "+"state"`. Using a `split(" ")`-order, the `String` is divided and the single values of the point are saved. The "state" value of the point is used to determine the state of interaction it was generated in.

- If the "state" value contains a 0, the point was generated at the beginning of a line drawing, and a new `pointsIncomming` array is edified, holding all points of the drawn line.

- If the "state" value contains a 1, the point was generated during of a line drawing and it is added to the `pointsIncomming`.

- If the "state" value contains a 2, it is the last point of the line, and the line drawing is finished. The point will be added to the `pointsIncomming`. Concluding, the content of the `pointsIncomming` array is added to the `linesIncommingFinished`, containing all finished lines.

The `OverlayThread` is the second thread being responsible for information exchange between the app of the service technicians and the application of the expert. Depending on the manner of conception and implementation of the systems, it is in charge of receiving and displaying tracking coordinates from the Processing application, or of calculating the tracking by itself and only receiving values to initiate the tracking process.

When tracking information is made available by the `OverlayThread`, all lines drawn by the service technician himself or received from the expert are shifted according to the tracking information. That means, that all lines in `linesDrawFinished` and `linesIncommingFinished` are fitted according to the tracking information.

Based on the existing tracking information, the line correspondent to the direction of the tracked object's movement in the video picture is calculated. The position of the lines saved in `linesDrawFinished` and `linesIncommingFinished` are then adjusted according to the calculated shift. The installation information thereby moves in correspondence with the tracked object. Thus, installation information can be displayed related to the object and the assignment of installation instructions to specific positions in the picture is sustained.

6.3.3.3 Processing application

For the expert to receive information about the working environment and about the task the service technician is working at, and for giving him information and instructions for task management, a Processing application was developed to be applied on a desktop PC. Figure 6.36 shows the basal structure of this application.

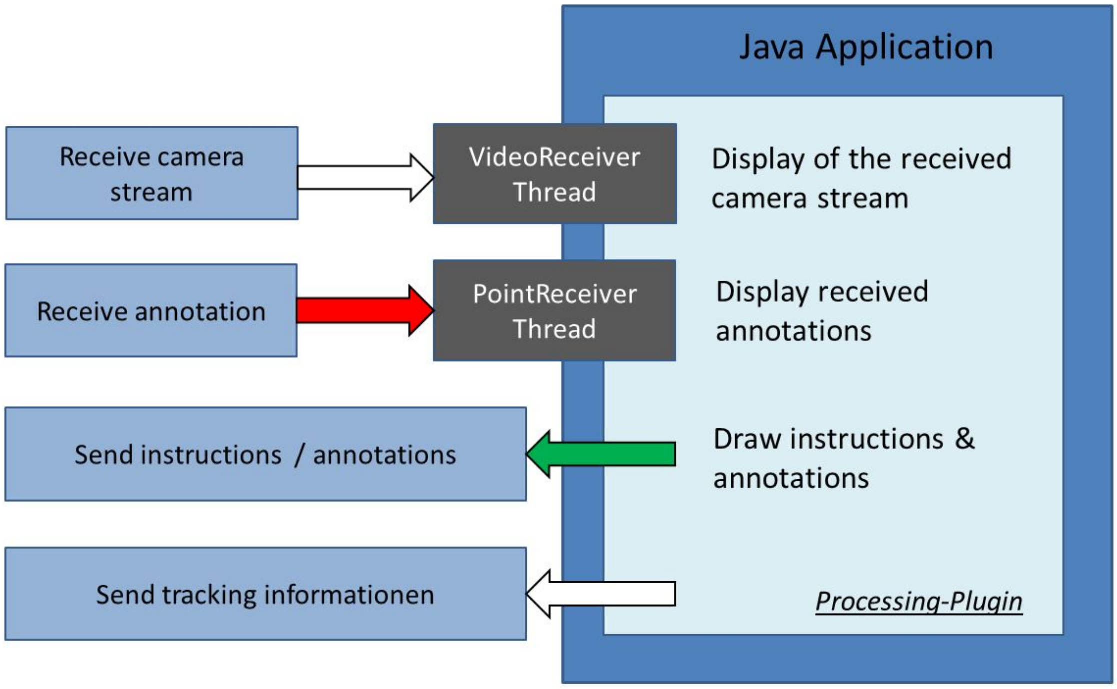


Figure 6.36: *Video interaction realization in the Processing application*

The application for the expert is a Java application made accessible by integrating the Processing Core files and expanding of the class `PApplet` functionalities of Processing. An output window is generated by Processing , that allows the display of received camera data as well as the interaction with the service technician.

The size of display is set by the `settings()`- method. In the `setup()`-method, different variables for the interaction are initialized, and the `VideoReceiverThread` and the `PointReceiverThread` are started. The `VideoReceiverThread` is in charge of receiving video pictures sent from the service technician's mobile device and to make them available for the Processing application. The video pictures are received as `byteArray` by the `DatagramSocket` via the network. With the help of a `ByteArrayInputStream`, the picture information is imported and then converted into a `BufferedImage`. Finally the `BufferedImage` is converted into the picture format `PImage` within the `img.loadPixels()` and `img.updatePixels()`-functions, that can be displayed by the Processing application. The `PointReceiverThread` is responsible for receiving installation information drawn in by the service technician and making them available to the Processing application. Line points are received as `String` by a `DatagramSocket` via the network. The `String` thereby has the same structure that is used in the Android App: `"x-coordinate"+" "+"y-coordinate"+" "+"state"`. The `String` is divided with a `split(" ")`-order and single values of the point are saved.

Visual output data are drawn on the display by the `draw()`-function in Processing. The function is executed as an infinity loop and consistently invoked until termination of the application. The function described below are thereby continuously invoked in the `draw()`-function. Using `vthread.available()`, the availability of new video pictures is checked at the `VideoReceiverThread` to display them. Via `prthread.available()`, the availability of new points for line drawing is prompted at the `PointReceiverThread`. As soon as the `PointReceiverThread` thread receives new points, these are processed in the same manner as already described for the Android App. There are two different arrays for data storage for self-drawn lines and received lines, respectively. According to the state of the drawing, a line is either saved in an array for unfinished lines or in an array for finished lines.

As soon as tracking information is available, either on part of the Android App or on part of the Processing application, the position of all saved and finished lines is adjusted. Based on a previously and a newly received tracking value, the shift of the tracked area is calculated. This shift is transferred to the lines, effecting them to follow the tracked area.

To allow the erasing of already finished lines, a button was placed in the display. This is shown as a rectangle which is drawn at a favored point on the monitor using the `rect()`-order. When the mouse is navigated over the button, it is prompted by means of the `mouseOverButtons()`-function and the button changes color. If the button is pressed, the `clearCanvas()`-function is invoked. It erases the content of all arrays containing line information, as well as all current tracking information.

The functions `public void mousePressed()`, `mouseDragged()` and `mouseReleased()` enable the expert to input into the Processing application. If the left mouse button is pressed on the monitor, the `mousePressed()`-function is activated. It registers the point of event and generates a new array for line drawing. If the mouse button stays pressed and the mouse moved across the monitor, the single points of event are sent to the Android application of the service technicians within the `mouseDragged()`-function. As soon as the left mouse button is released, `mouseReleased()` activates itself. Line drawing is finished and the content of the `sendingPointsProgress` array is transferred to the `sendingPointsFinished` array, where all finished lines are saved. Moreover it is tested in the `mouseReleased()`-function, if an object was selected for tracking.

6.3.3.4 Tracking on PC versus tablet PC

Several possibilities to implement object tracking were taken into account when developing the communication system. We tried JMyron [200], OpenCV [201] (Template Matching, Camshift, ORB-Tracker) and BoofCV [202] (Circulant, TLD, Mean-Shift, Sparce Flow) libraries for the use in Java and Processing on Android applications. It could be shown, that BoofCV provided good tracking performance and was more efficient than the other libraries. This coincides with the results of an experiment [203], where performance and velocity of different Computer Vision libraries including OpenCV and BoofCV on object tracking have been tested.

A further evaluation followed the experimental procedure of [204] and analyzed the algorithms provided by the BoofCV library more deeply. It used video material from two setups depicted in Figure 6.37: The LEGO environment served as base for user experiments and was suitable due to the many similar objects. The environment of the switch cabinet was drawn from scenario 3 described in Section 4.5 in order to bridge the results of the Augmented Reality studies to studies with remote assistance.

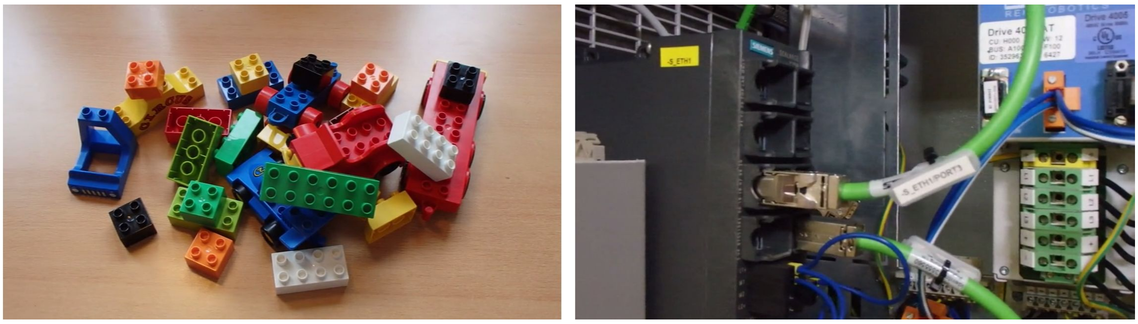


Figure 6.37: *Setup of Tracker evaluation a) LEGO environment and b) Switch cabinet*

Each video frame from the different environments has been manually annotated with the optimal tracking result. The results of the tracking algorithms have been compared and the superposition has been used as a benchmark value depicted in Figure 6.38. The Circulant Tracker provides the best value with 83.32% for the LEGO Scenario and 83.32% for the switch cabinet scenario. The TLD tracker performed only slightly poorer with 81.24% in the LEGO and 63.24% in the switch cabinet scenario. This is due to the fact, that the TLD tracker automatically adjusts the size of the tracked object, so that the superposition values have been corrupted slightly. The experiment also was performed on different scenarios like rotation of the object or recovering the object after it has disappeared from the screen. This feature was only provided by the TLD tracker and that was why it was chosen for further implementation.

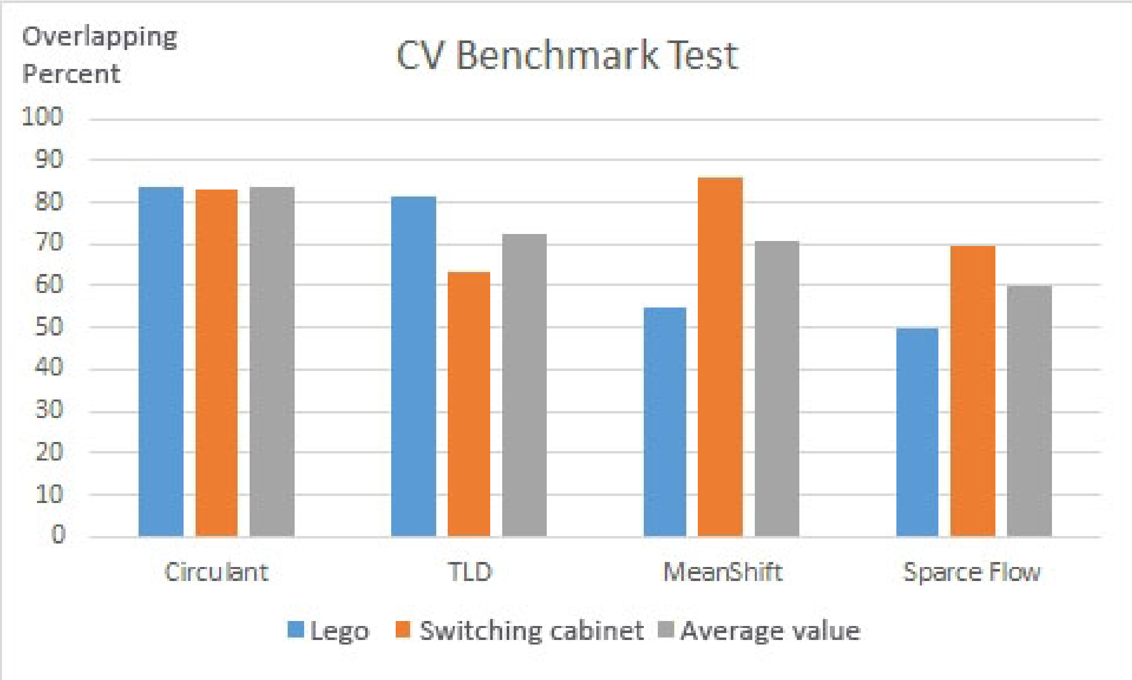


Figure 6.38: *Results of the benchmark for the BoofCV trackers for both environments*

As soon as picture information is available for a system, it is possible to apply the method of Computer Vision for picture analysis. Since visual information is available for both part systems, object tracking is feasible on both the Android App and the Processing application likewise. Figure 6.39 opposes both of the implementation approaches.

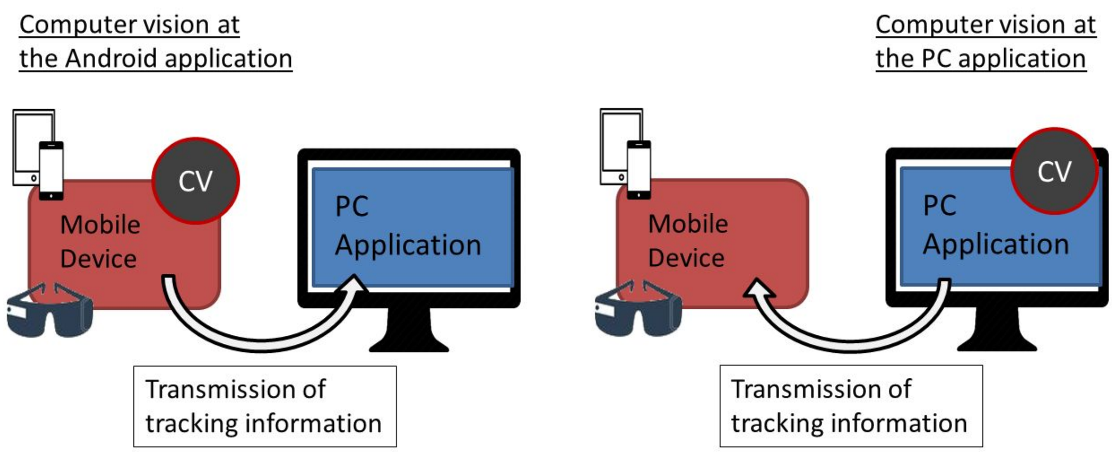


Figure 6.39: *Visualization of the concept*

Depending on the kind of implementation of Computer Vision, different advantages and restrictions arise having considerable effect of the functionality of the application. To make more accurate statements about the different concepts of linking Computer Vision, two distinct versions of the application were implemented and analyzed. A testing system consisted of each an Android application for a mobile device, and a Processing application for a desktop PC. In the first testing system, Computer Vision was linked to the Android application and in the second testing system, Computer Vision was linked to the Processing application. As a mobile device, an ASUS tablet was employed.

Although the functionality of Computer Vision was linked to different applications in the both testing versions, its implementation still follows a consistent structure. In the following section this structure is looked at in detail.

As soon as an application is able to provide picture information, it can be further processed by Computer Vision. To distinguish the object or picture partition to track, an area has to be selected from the picture for the CV to be analyzed. The relevant area can for instance be selected by touch or mouse interaction. After selection of the picture area, its sufficiency for an operation of the tracking software is checked. If for example only a very small area is selected, the user is prompted to repeat the input. If the selected area is sufficient, the tracker is initiated along with it. All newly available pictures are then scanned for the selected area by the tracker. If no match is found, the tracker signals the untraceability of the wanted object. If a match is found between the wanted object and the picture, the position of the located area is given. This position can be used to display object-related installation information.

In the following section, the characteristics of the different implementations of Computer Vision are elucidated in detail.

Computer vision algorithms on Android App

On parts of the Android App, Computer Vision was implemented in the class `ObjectTrackerActivity`. After the camera was started and picture information is available, firstly the `updateTracker`-function is prompted, which transfers the information to the tracker. Then the processed visual information is displayed by the `visualize`-function on the output device and sent to the Processing application via the network. In the `updateTracker` function, the camera picture is transferred as a `MultiSpectral`-file and has to be converted into an `ImageUInt8`-Format, before the picture can be sent to the tracker. A picture area to track can be selected through touch functionality by the service technician or received via the network from the expert. It is prompted with a UDP thread from the `udpServer.hasData()`, if new installation information was delivered by the expert. The selected picture area is then saved in a `QuadrilateralF64`-file format. Using the function `tracker.initialize`, the camera picture and the selected area is transferred to the tracker for initiation.

After successfully starting the tracker, the selected area is compared with each new camera picture. When the tracked object is moving, the position of the picture area is adjusted within the function `tracker.process()`.

Computer vision algorithms on Processing application

On parts of the Processing application, the functions of Computer Vision are implemented in the `draw()`-function. It is firstly prompted via `vthread.available()`, if video pictures were received from the Processing application via the network. Then it is checked by `tracker.initialize()`, if the tracker can be initiated with the selected picture area. An area can be selected in the Processing application by mouse interaction. Moreover, an area selected by the service technician can be received via the network. If the tracker could be initiated with the selected picture area, all pictures received by the service technician are scanned for the selected area using `tracker.process()`. If the wanted object is found in the camera picture, the position of the area is adjusted by `target.set(tracker.getLocation())`. Finally, current tracking information is sent to the Android App of the service technicians via the network.

Measurement of delay

In order to decide, on which side of the communication link the object tracking shall be realized, the latency was measured to estimate the performance of both implementations. The measured values are summarized in Figure 6.40.

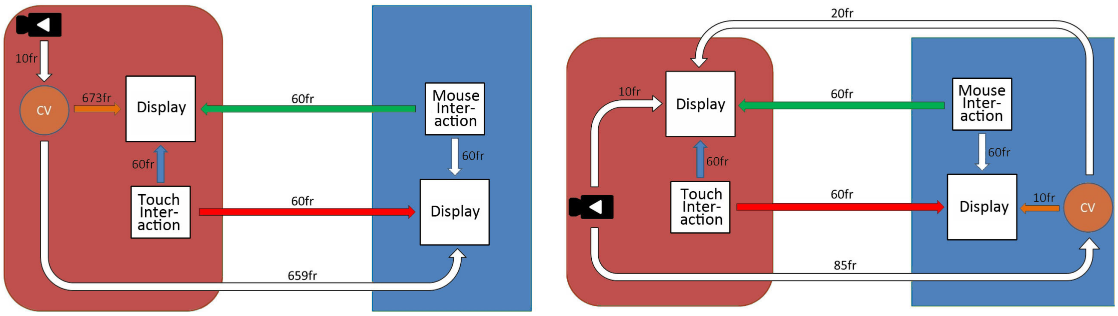


Figure 6.40: *Latency of computer vision algorithms a) in the android application and b) at the Processing application on the PC*

The result of the measurements is, that there is high latency, if the computer vision algorithms are performed in the Android application. After video information was made available by the camera, a delay upon 670 frames has been measured until the image processing had been completed and the information was displayed. Together with the communication link delay and the processing, it took 659 frames until the image information have been presented at the Processing application.

If the computer vision algorithms are performed on the PC, the measured latency has been less. This is due to the higher computational power of the PC. Therefore it has been decided that the Computer Vision algorithms have been linked to the Processing application on the PC.

6.3.3.5 Test application on the tablet PC

After completion of the investigation, different versions of the communication application were implemented for various mobile devices for experimental operation. In this section it is described, which extensions, modifications and characteristics the implemented systems conduct in the basal structure of the communication system.

For evaluating the communication application, an ASUS tablet was chosen as a mobile device. It was a tablet of the model ME302C with an Android operating system Version 4.3 installed. The result is depicted in Figure 6.41. Three different versions of the communication application were implemented, that were meant to be employed on the tablet.

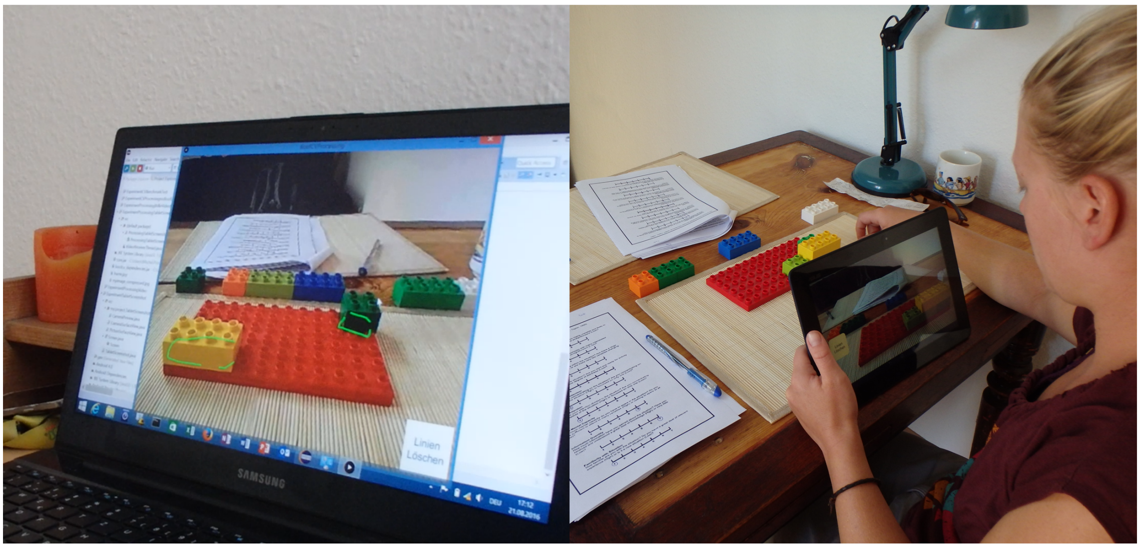


Figure 6.41: *Realization on the tablet PC with LEGO examples*

- **Tablet with video and tracking**
  The first application implemented for employment on the tablet enables the transfer of video information between service technician and expert and supports display of divided installation information by a TLD Object Tracker. With this version being the basis of development in communication applications, there are no noteworthy deviations or modifications of the described basal structure.

- **Tablet with video without tracking**
  The first application implemented for employment on the tablet enables the transfer of video information between service technician and expert and supports display of divided installation information by a TLD Object Tracker. With this version being the basis of development in communication applications, there are no noteworthy deviations or modifications of the described basal structure.

- **Tablet with screenshot**
  In the third application for the tablet, the service technician and the expert communicate using screenshots that are being sent back and forth in either

direction. The service technician can take a screenshot of the working environment with the camera and send it to the expert via the network. The expert can directly draw installation instructions into the picture with the mouse and send it back to the expert.

6.3.3.6 Test application on the AR HMD

In another experimental condition, the communication application was to be examined with AR glasses as the mobile output device. For development and experiments, Epson Moverio B-200 AR glasses were used. A screenshot condition could not be implemented for the AR glasses, since this cannot be connected to the operating mode of the glasses. An implementation of a video application without object tracking was also set aside, because it is not practicable for operating on the glasses. Only the version for video transmission and object tracking were implemented.

The application implemented for the AR glasses deviates from the basal structure of the application in such a degree as the transferred installation information having to be displayed on two monitors simultaneously to generate a 3D effect for the user.

The B-200 is able to create a 3D effect showing a different picture of each eye and superimposing them. Therefore the original output picture with the measurements 960x540 has to be converted into two adjoining scaled pictures with the measurements 480x540, respectively.

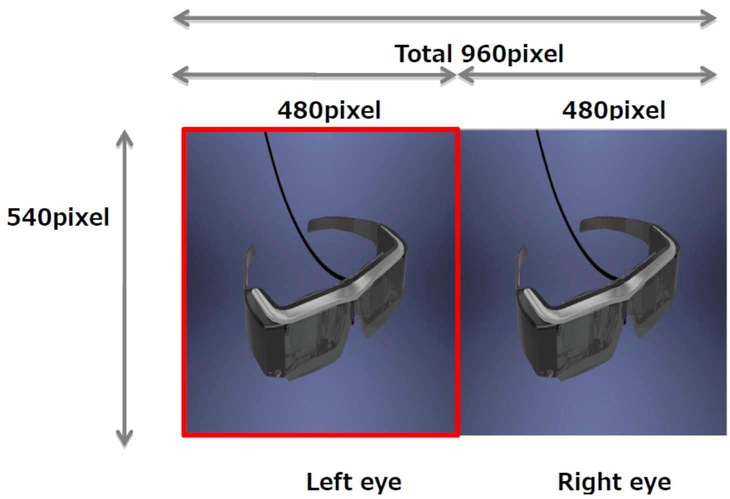


Figure 6.42: *AR screen resolution problem*

To be able to display the installation information side by side, another `View` named `MirrorView` was implemented. The `OverlayView` and the `MirrorView` are configured in the `Layout.xml` to comply with the requirements of the glasses' 3D display for size and positioning.

After both Views and their `SurfaceHolder` are initiated in the main class, the `SurfaceHolder` of the Mirror Views is transferred to the `OverlayView`. During each operation of drawing of installation information on the `OverlayView`, similar information is drawn on the `MirrorView` by the transferred SurfaceHolder.

Since in Android applications the status bar is shown per default, further configurations had to be done to ensure the required 540px picture height. During the initialization of the main class, in the `onCreate()` method the status bar is removed (as can be seen in Figure 6.43).

Figure 6.43: *Status bar on the AR application*

The entire application for the AR HMD device in the application described in Section 4.5 can be seen in Figure 6.44.

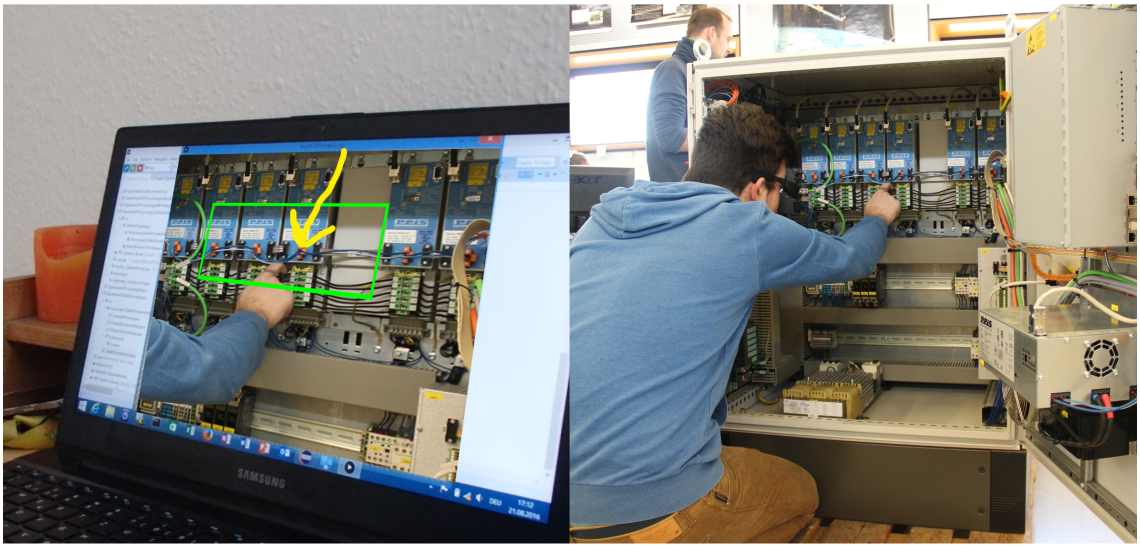


Figure 6.44: *Video interaction on the Switch cabinet with HMD device*

6.3.3.7 Realization for telemaintenance experiment

For the telemaintenance architecture, the tablet with screenshot application has been used in the AMS software architecture. The service technician is carrying the mobile device. By accessing its camera, the expert is able to retrace the area of view of the service technician. With the Paint App depicted in Figure 6.45, both expert and service technician can draw simple outlines to point out areas of interest.

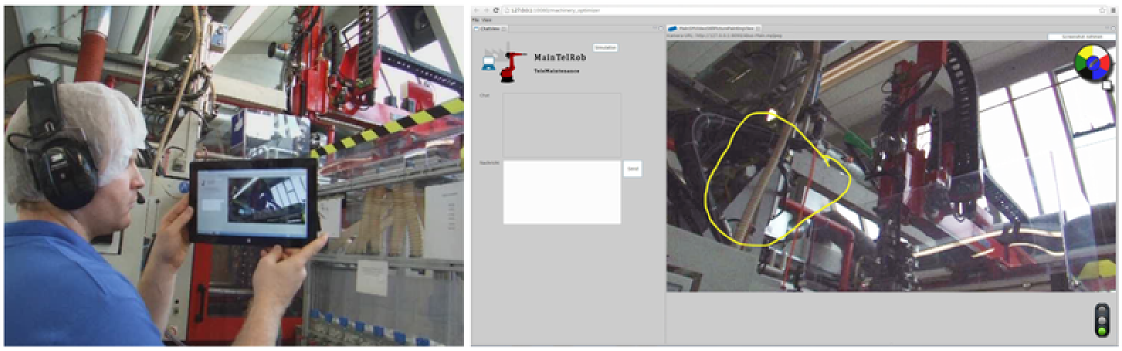


Figure 6.45: *Service technician with tablet PC (left) and screenshot of tablet GUI with Painting App (right)*

6.4 Proof of concept: Motor exchange

In order to evaluate the telemaintenance functionality in a real-world example, a "proof of concept" study has been performed. A quantitative evaluation has not been possible, because the project team was not able to generate the necessary test user quantity.

Based on the qualitative findings of the contextual analysis for telemaintenance, the task of a motor exchange has been chosen in the specifications.

> CITATION FROM THE SPECIFICATION DOCUMENT
>
> SPECIFIED USE CASE: The expert instructs the user to exchange an axis motor. After that the manipulator need to be referenced anew and the safety controller has to be confirmed.
>
> STATUS: Currently the service for manipulator repair is performed via telephone. The remote access is also possible but without teleoperation. If this support does not suffice, a technician has to travel to the facility. The journey time consumes valuable production time, because the manipulator is broken.

Concordantly, all partners declared, that a motor exchange needs to be done by highly qualified personal. In facilities without special trained experts, the KUKA Industries experts need certainly to travel there. The project team wanted to demonstrate, that the journey can be replaced by a telemaintenance software developed in MainTelRob, as the service technician of the facility can perform the difficult repair with remote support.

A motor exchange for a robot axis is very elaborate. The worker has to prepare several steps (like retaining the axis) and needs at least an hour for the repair, if he is trained well. "A motor exchange has never been done remotely. If the local worker does not know what to do, a service technician has to travel abroad." as one manager puts it. A service technician who is familiar with the test plant has mentioned: "If someone has never done that, it can take more than three hours.

In contrast to the other verification scenarios, this experiment has not been performed in the second project year, but at the end of the project. For a long time, it was unsure, if the proof-of-concept can be performed in the active production environment at all.

Performing such a study in an active production line is very challenging: The production process needs to be interrupted, the risk of damaging the robot can never be omitted completely and only few expert personnel is suited for such a difficult task. Thus, there is only one run of the experiment possible under real-world conditions.

For the experiment procedure, the motor of the third axis (z-axis) was chosen. It was agreed, that the same motor should be dismantled, taken out and installed again. The service technician needs to climb onto the molding machine in order to perform those tasks and access the third axis of the Cartesian robot. After the motor has been changed, the safety controller has to be reset and the axis needs to be referenced manually.

6.4.1 Procedure

Figure 6.46 presents the schematic experiment setup with staff assignment and used material. Goal of the documentation was a parallel representation of the operations on both sides in order to perform a qualitative analysis.

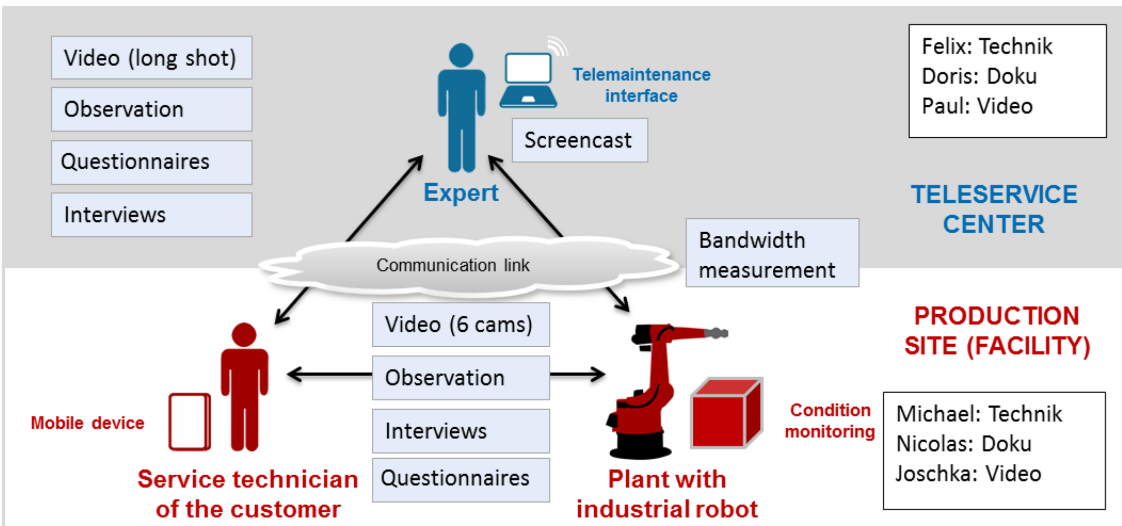


Figure 6.46: *Test setting for motor exchange (July 28th, 2015)*

The service technician supporting this experiment, is a regular trained technician, who has been chosen, because he had not performed the motor exchange during a long time.

Evaluation took place at a plant in an active production line which consists of an industrial robot, an injection molding machine and a montage system which is displayed in Figure 6.47. There are two stationary cameras which have been installed

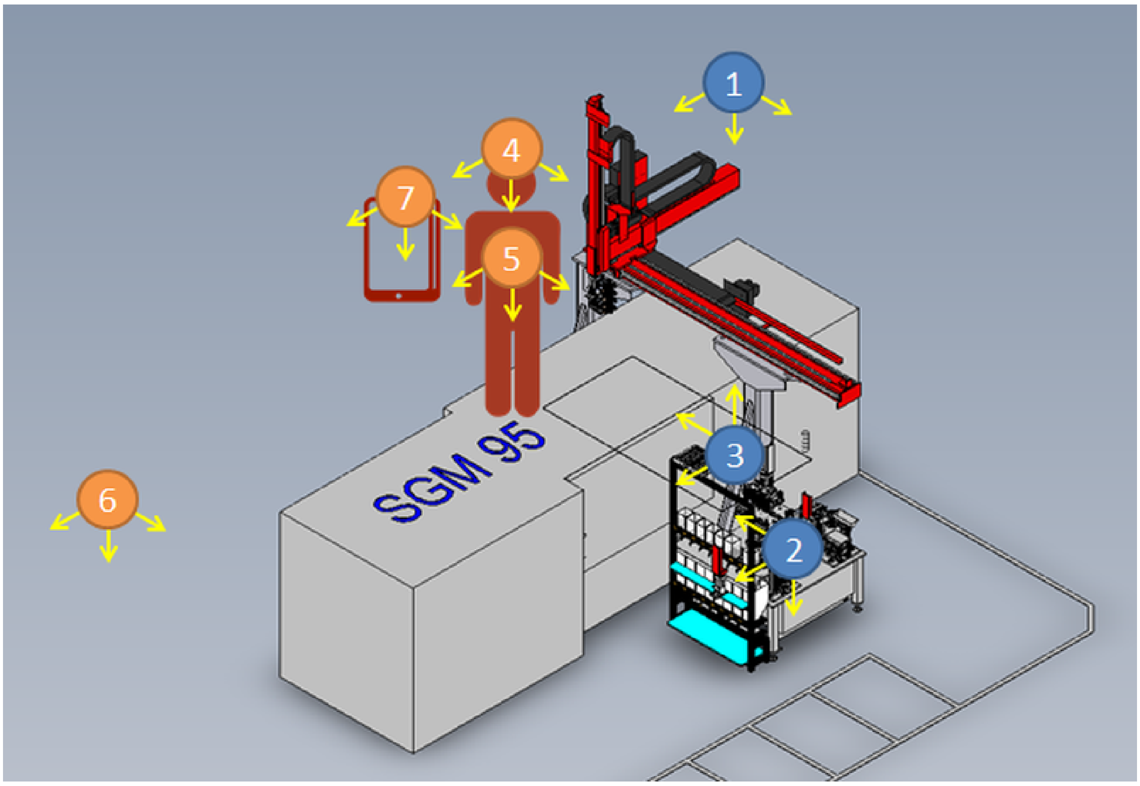


Figure 6.47: *Experimental setting and camera locations*

permanently for telemaintenance (camera 1 and camera 2). Only those two locations have been allowed due to worker privacy. The third camera is also stationary and has been set up especially for the experiment. The service technician, who performs the task, has not performed a motor exchange for a long time and would not have tried to do such a complicated repair on his own. He has an action cam applied to his head (4) and to his chest (5) and he carries a tablet PC which also has a camera included (7). Another camera is provided by a moving person (6), who tries to capture the action as good as possible.

The expert is located on a completely different location and has access to cameras (1), (2) and (7). He also has voice contact to the service technician.

6.4.2 Evaluation

6.4.2.1 Operation Survey

Performing such a study in an active production line is very challenging: The production process needs to be interrupted, the risk of damaging the robot can never be omitted completely and only few expert personnel is suited for such a difficult task. Thus, there is only one run of the experiment possible under real-world conditions.

For the experiment procedure, the motor of the third axis (z-axis) was chosen. It was agreed, that the same motor should be dismantled, taken out and installed again. The service technician needs to climb onto the molding machine in order to perform those tasks and access the third axis of the Cartesian robot. After the motor has been changed, the safety controller has to be reset and the axis needs to be referenced manually. At the beginning of the experiment, both test persons have been introduced to the interfaces, but were left undisturbed by the scientists during the experiment. This is especially important, because the service technician balances on top of the robot in a height of two meters. This was also a reason, why freeze probe techniques were not possible.

On each interaction side a team of three scientists documented the process with video cameras, recorded affect and behaviour of the test person [205] and performed critical decision identification. Table 6.1 shows the chronological steps of the repair task together with the identified critical decisions. The team recorded the critical decisions of the remote expert and the local service technician independently, so that some are mere relevant for one side of the communication.

Table 6.1: *Motor exchange: Chronological working steps and identified critical decisions.*

Time	Chronological working step	Local	Remote	ID
10:14	Beginning		X	1a
10:15	Axis stabilisation	X		2a
10:20	Motor identification	X	X	3
10:35	Tablet PC position during work	X		4
10:45	Required tools	X		5a
10:45	"Old" motor dismantling		X	6
10:50	Tablet interaction for assembly	X	X	7
11:08	Controlling the robot	X	X	8
11:10	Moving to control cabinet	X		9a
11:13	Tablet PC position	X		10
11:15	Backup copy generation		X	11
11:18	Process completion	X	X	12a

6.4.2.2 Video-Interaction Example

The video interaction method with highlighting functionality, which has been described above, was used by the test persons on their own account. They have used it in critical decision 3 (CD3) to identify the motor and to discuss the first production steps (shown in Figure 6.48 where the cover cap of a screw is highlighted in green) and later in CD7 for discussing the assembly steps.

6.4.2.3 Cooperative Use of Teach Pendant

For a correct performance of the motor exchange, the robot axis needs to be moved several times. Further, some tasks need to be performed in the robot control menu,

Figure 6.48: *Tablet PC screenshot of the video interaction*

like, for example, the reset of the safety controller and the referencing of the axis.

Figure 6.49: *Expert with teach pendant at a remote working desk*

For all those tasks, the teach pendant of the robot is needed. The expert has decided spontaneously to fetch a teach pendant on his own for guiding the service technician through the menu, which is shown in Figure 6.49. At the same time, the service technician uses the teach pendant (Figure 6.50). The picture from the head-mounted action camera shows the teach pendant (on the right), which is filmed by the tablet PC (on the left). The video stream of the tablet PC is then submitted to the expert's interface.

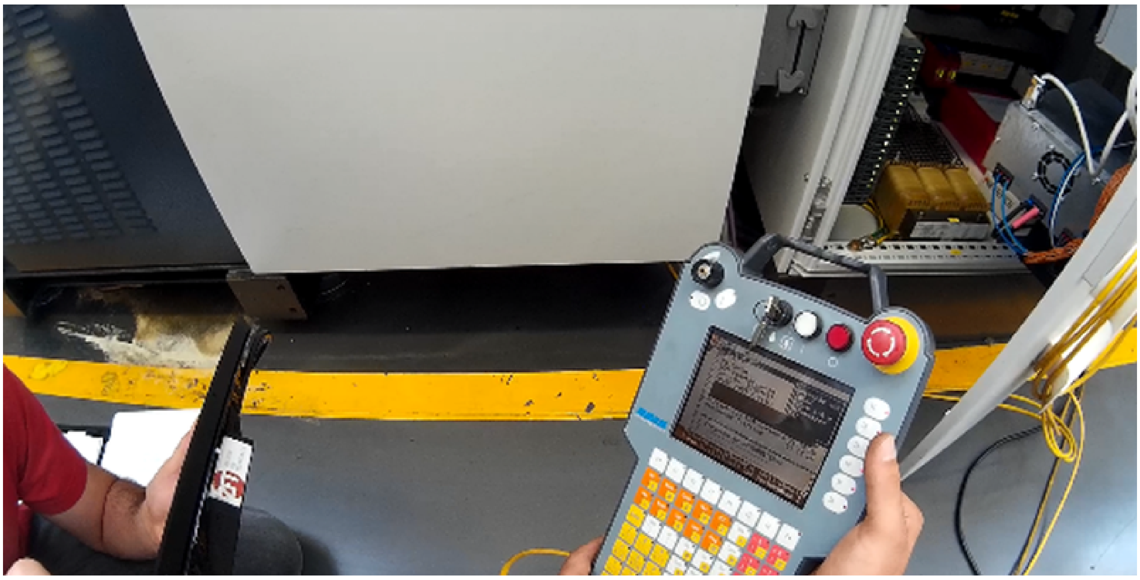


Figure 6.50: *Service technician view (head-mounted camera) on teach pendant*

6.4.2.4 Applied Human Factor Methods

After the experiment, both test persons were asked to fill out several questionnaires: The NASA TLX [60] for workload, the SART [89] for situation awareness, the QUESI [72] for intuitive use, the ISONORM 9241/10 [71] for software ergonomics, and the Keirsey Temperament Sorter [206] for detecting eventual communication barriers.

Afterwards the participants were asked to watch the video sequences of the identified critical decisions (video-guided walkthrough with think-aloud [54]) and answer the CDM probes on those events. This method was chosen, because the experiment could not be interrupted in between. With the video recording, the probands were able to put themselves in their position during the experiment, which aims at enhancing the feedback quality.

As described above, the experiment could only be performed with two test persons. Thus, the measured values only express subjective impressions. The usability feedback was altogether quite positive, both participants confirmed this with the QUESI and ISONORM values. As the service technician's interface is substantially simpler, the values were higher. The task load of the remote expert was higher than of the service technician, which is relatively low after all. The expert also stated that he perceived stress in some of the critical decisions.

The critical decision probes were analyzed according to [207]. An excerpt of the results for the remote expert is provided in Table 6.2. It shows, that the expert is not completely content with the video information provided by the telemaintenance interface and states a lack of situation awareness.

Those findings from the CDM are supported by the SART values displayed in Figure 6.53. While the expert was not able to see all the actions of the service technician due to different camera perspectives, the service technician had perceived

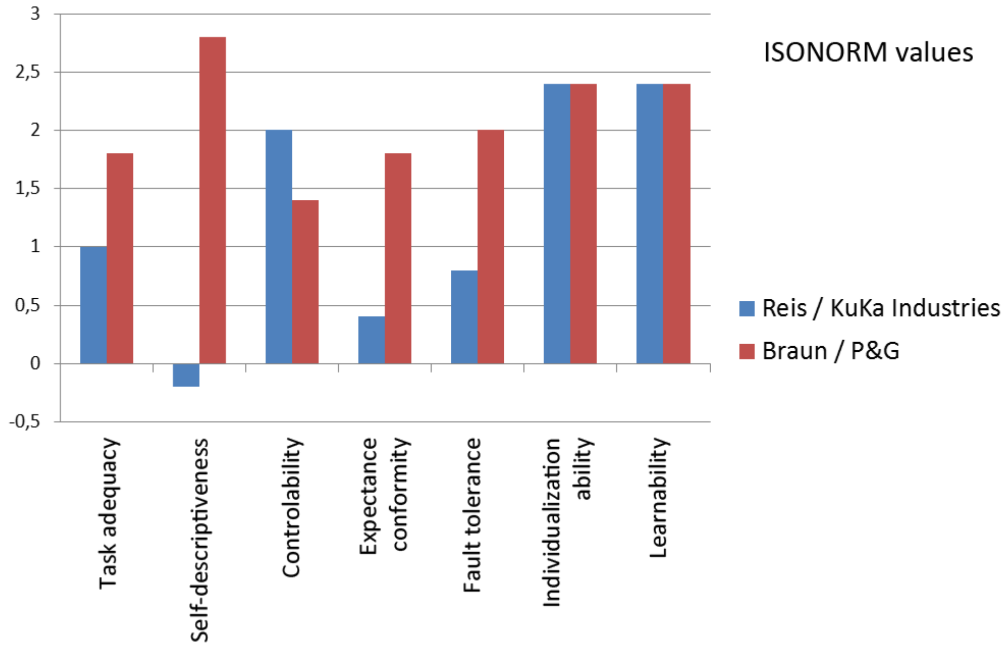


Figure 6.51: *ISONORM values for expert (blue) and service technician (red)*

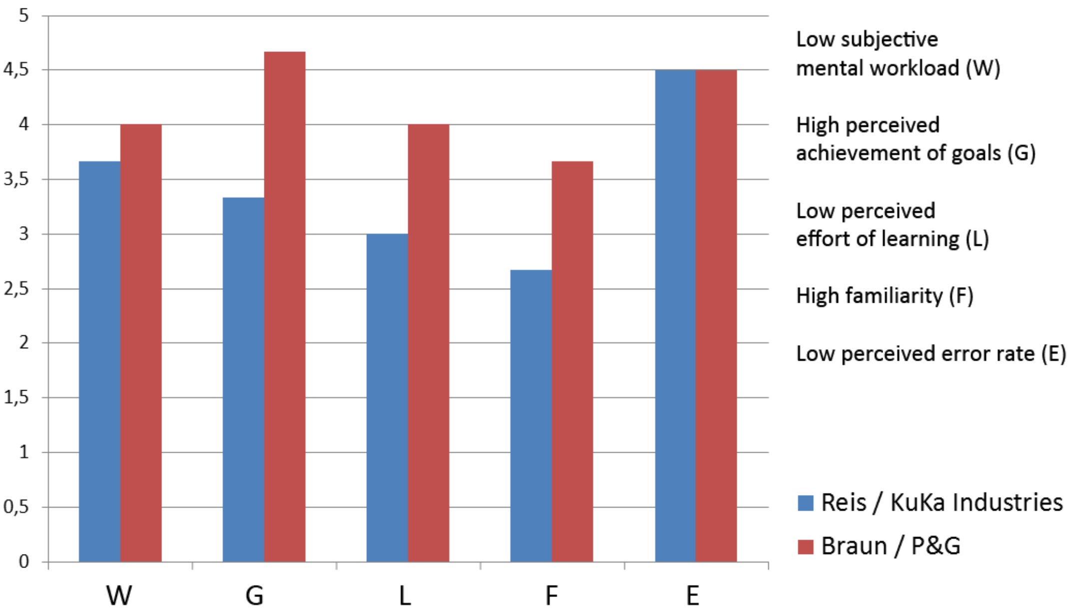


Figure 6.52: *QUESI values for expert (blue) and service technician (red)*

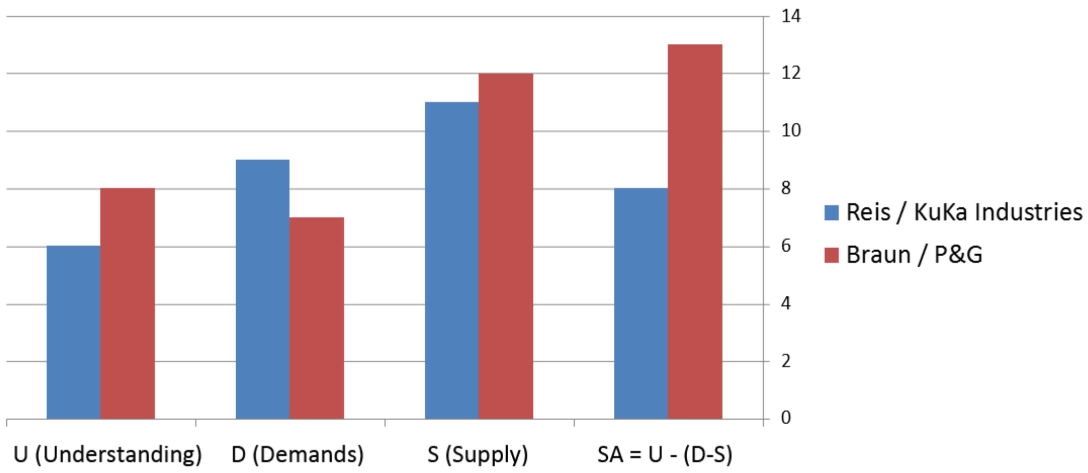


Figure 6.53: *SART values for expert (blue) and service technician (red)*

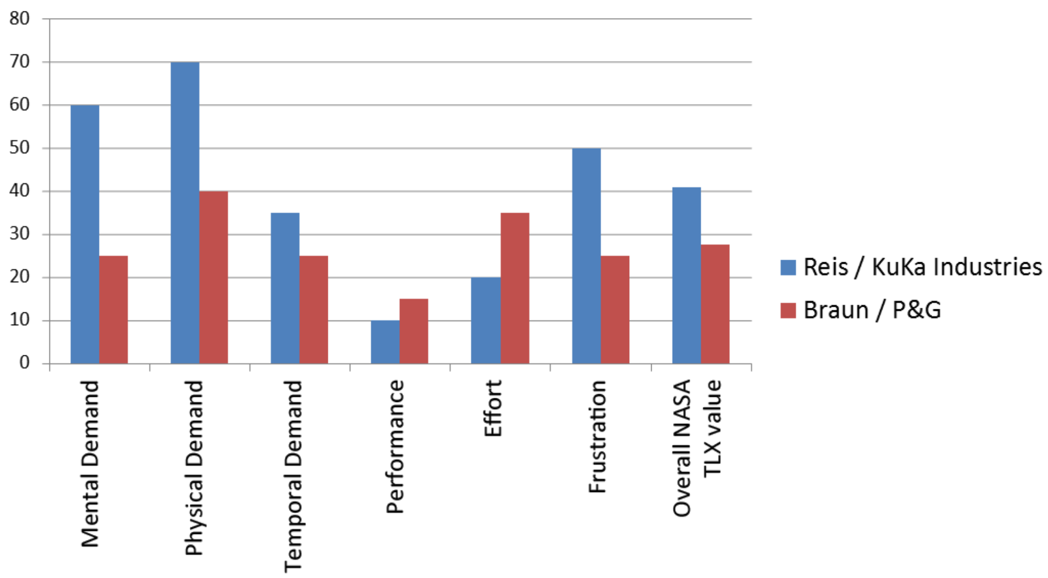


Figure 6.54: *NASA TLX values for expert (blue) and service technician (red)*

that "nothing can go wrong" because "somebody is watching over my actions".

6.4.2.5 Video-based subsequent evaluation

In order to go further than the "proof of concept", more perspectives of the telemaintenance process were recorded during the experiment than those provided to the remote expert (all different cameras are displayed in Figure 6.47). For each camera, the observed critical decision scenes (see Table 6.1) have been clipped separately and summarized in a two-hour movie. Ten scientists with a robotic background were told to put themselves in the position of the expert and identify the situation awareness for the task with the SART questionnaire based on the video information.

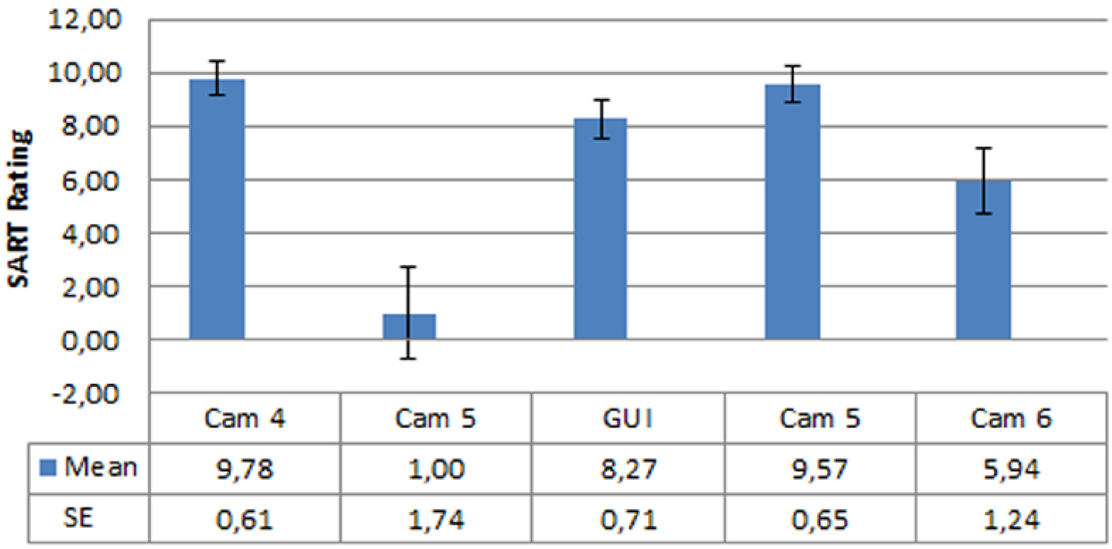


Figure 6.55: *Video-based subsequent evaluation (mean SART values)*

As displayed in Figure 6.55 the mean value for the GUI perspective (consisting of the cameras 1, 2 and 7) has not reached the highest value. This could have been already assumed after the analysis of the expert's feedback. While the camera in the montage system (2) had not monitored the workspace in any of the scenarios, the overview camera (1) did not provide enough resolution to monitor the task of

Table 6.2: *Mean questionnaire values*

Questionnaire	Service technician	Expert
NASA TLX	27.5	40.8
QUESI	4.2	3.4
ISONORM	2.1	1.3
KTS-II	ESTJ	ESTJ

the service technician in detail. The best of the three GUI cameras was the mobile tablet PC camera (7), but during the repair tasks, where the service technician needed both hands, it was not useful, either. The measured situation awareness of the video-based subsequent evaluation matches to the situation awareness value provided by the expert in Figure 6.53.

The static camera on the montage system (camera 5) has achieved an extremely low value. This is due to the fact, that in most cases the service technician hides behind the robot and also the switching cabinet cannot be seen from this perspective (CD 10 and 11). Surprisingly the hand camera (6) was not able to catch enough process information either, because the service technician cannot be seen appropriately when he is operating on top of the molding machine.

The best view on the operation is provided by the two action cameras (head-mounted: camera 4 and chest-mounted: camera 5) whereas the situation awareness is clearly affected by the partially rapid head movements of the service technician.

6.4.3 Summary

Altogether, the experiment has been very successful, because it showed the feasibility of a difficult telemaintenance task, which has not been performed before: The motor exchange on a Cartesian industrial robot. While the service technician at the plant felt reassured by the remote support, the remote expert still lacked situation awareness and was not content with the three camera perspectives provided by the experimental telemaintenance interface.

Video recordings from different other camera perspectives on the experiment scene were analyzed and compared to the experimental setting by scientists with hindsight. The results showed the best values for the two action cameras (head-mounted and mounted around the chest of the worker). While both cameras provided a first-person view on the repair operations (see Figure 6.56), an observer is able to understand what the service technician is doing. Nevertheless, static cameras which

Table 6.3: *Critical decision method (excerpt)*

Goals	Why?	How?	Solution
Observe process better	Service-technician actions are not visible well enough all the time	Sometimes, visual feedback is missing; feeling responsibility for actions	More video information; building mutual trust
Prohibit mistakes	Expert cannot really "do" anything	"Feels" when he needs to give an instruction	More video information; building mutual trust

provide an overview of the scene had been recommended. The new developed video-interaction technique using a mobile device had received a good user feedback.

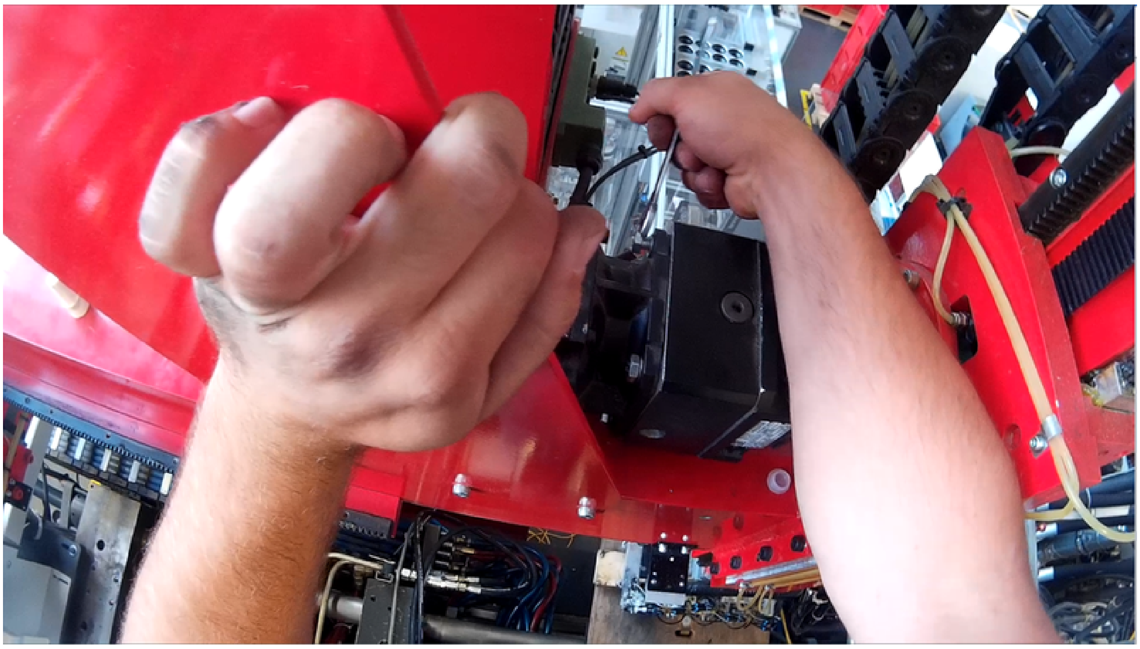


Figure 6.56: *The head-mounted action camera provided the best view on the difficult repair steps, but was very instable due to head movements*

6.5 ARTab

6.5.1 ARTab Concept

The application "ARTab" should enable users to walk around the plant. While they walk in a hall, they perceive a virtual model of the plant through a tablet PC as depicted in Figure 6.57, the position of the tablet PC (red) is tracked by an external tracking system (orange). We use the iSpace System "Metris" by Nikon Motrology [208], a laser-based measurement and tracking system which covers our robotics hall. Rotating transmitters emit laser beams, which are detected by photo sensors on a communication module (PCE). The PCE will transmit the data to a laptop, which calculates and saves the position information. The accuracy of this system is below 0.25 mm with a sampling rate of 40 Hz.

The application concept in Figure 6.57 is divided in two parts: the reality on the right and the virtual scene on the left. The tablet PC has a position in the real world and from this viewpoint the camera takes pictures. We get the real position from the tracking system and put a virtual camera just to the same position in the virtual scene. Then we use the real image of the tablet camera and use it as a background to the scene. When we render both, we get virtual content (the red robot) in the

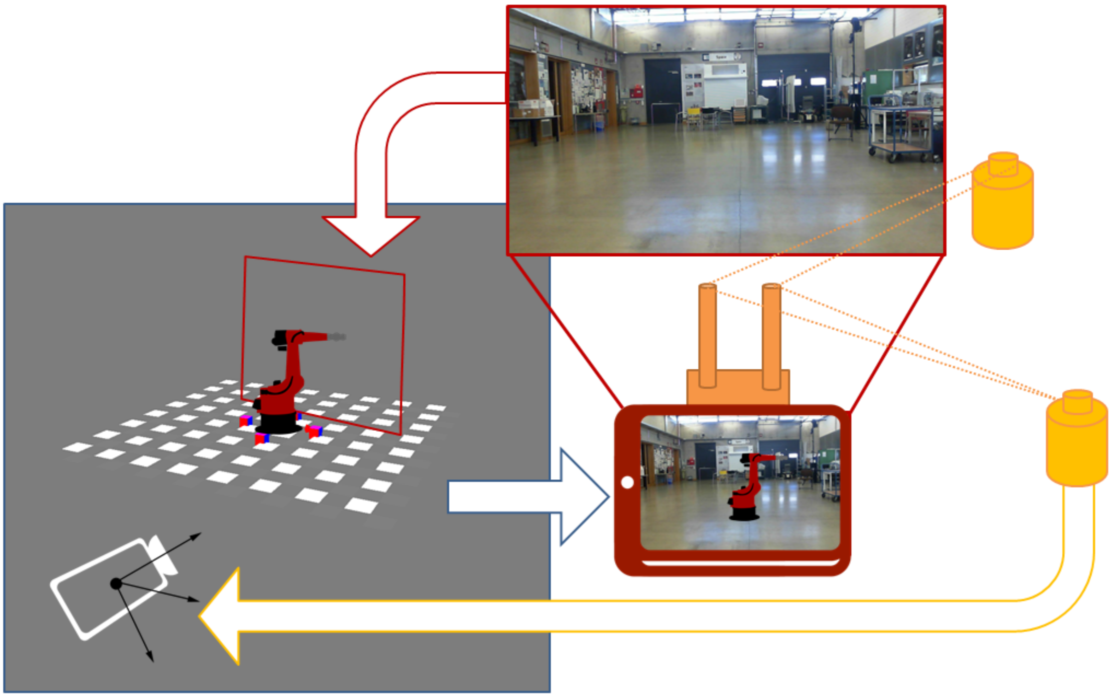


Figure 6.57: *ARTab application concept*

hall. If the tablet PC is moved, the viewpoint of the virtual scene changes, too. The user can walk around the 3D-scene and view it with real context through the tablet PC.

6.5.2 Architecture

We use different coordinate systems displayed in Figure 6.58:

- The tracking frame M (orange) with origin in one of the corners of the hall.
- The tablet PC frame T (red) with origin at the measurement point of the iSpace sensors.
- The camera frame K (cyan).
- The two-dimensional picture frame B (green) of the camera picture and the rendered scene.
- The checkerboard used for calibration has the frame S (yellow).
- The virtual scene has the coordinate system V.

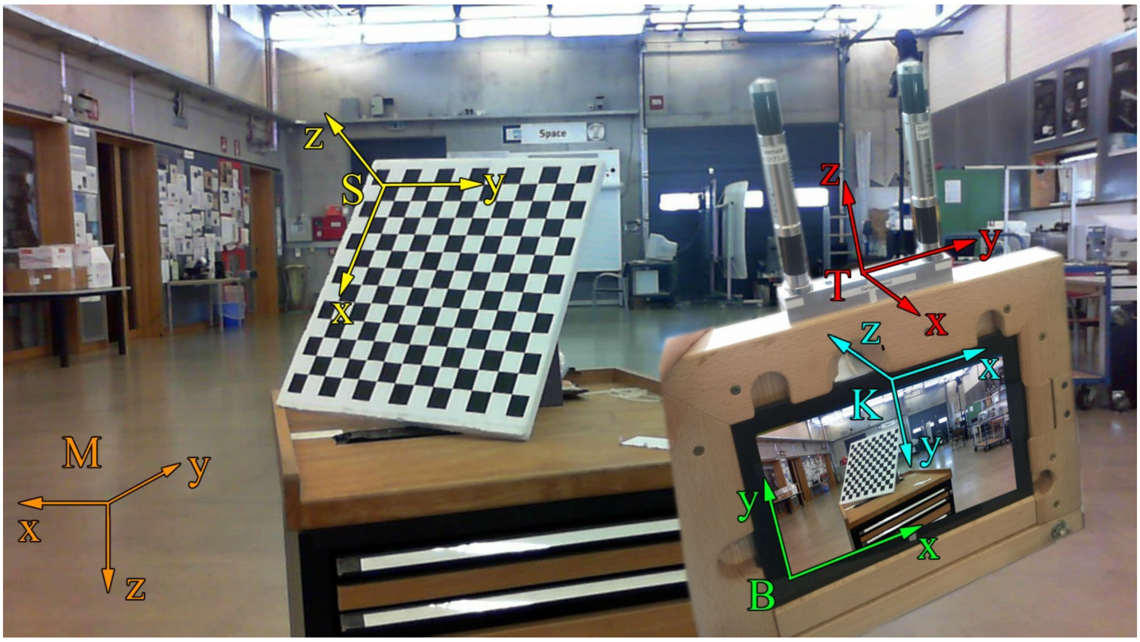


Figure 6.58: *Use of different coordinate systems*

6.5.2.1 Hardware and Network

The system uses a tablet PC, the tracking system iSpace and laptop for calculation of the location information. We use a Microsoft Surface Pro, which is clamped in a wooden frame as it can be seen in Figure 6.58. All hardware components are connected with two networks as depicted in Figure 6.59. The laptop uses an LAN connection and an access point to communicate with the iSpace (Metris) system. Another WLAN network connects the tablet PC to the network.

6.5.2.2 Software

The software architecture is depicted in Figure 6.60. The position data from the iSpace system is transferred via UDP from the laptop to the tablet PC. The absolute camera position with respect to the tracking system frame is calculated and then transferred into the virtual scene frame. The rendering takes place in Unity [209], where the camera location and the camera video are used together with a moving virtual model of a plant. The configuration data like camera calibration and network configuration is stored in an external file.

6.5.3 Frame calculations

6.5.3.1 Definitions

We use a three dimensional real space. A frame or coordinate system is defined by its base out of three orthogonal vectors. Points in the frame are defined by Equation

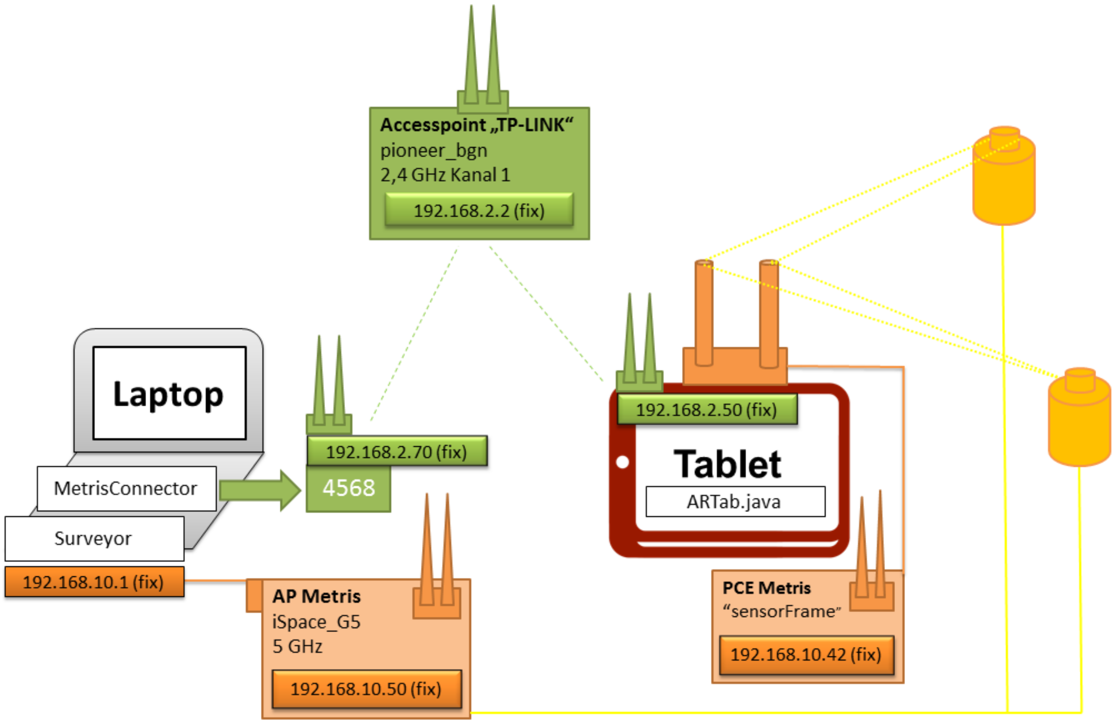


Figure 6.59: *Hardware and network configuration*

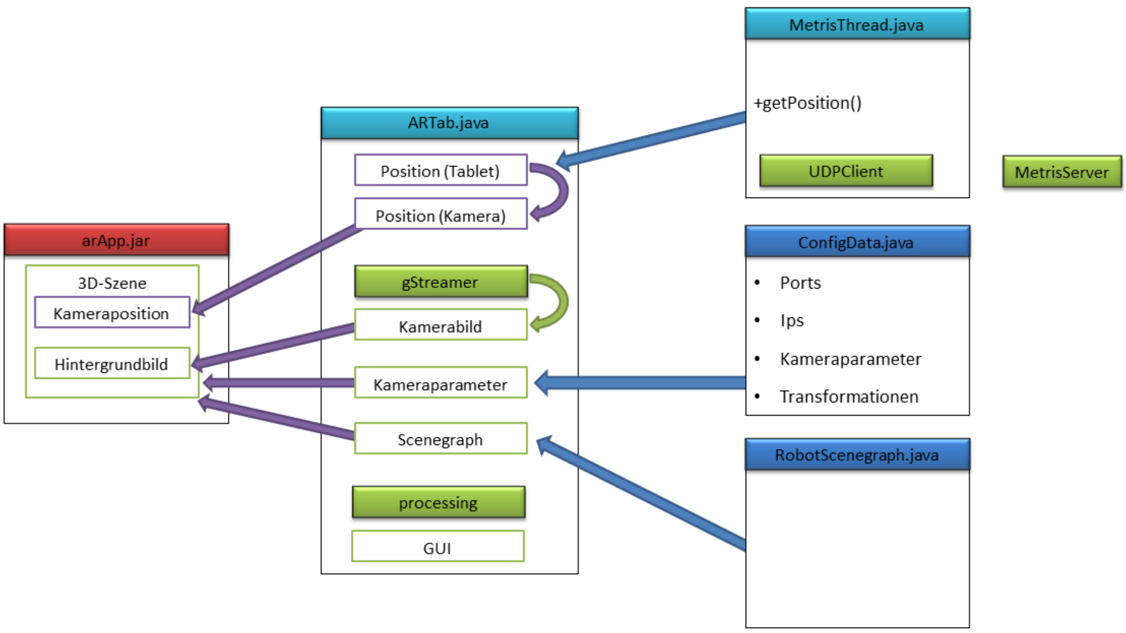


Figure 6.60: *Software architecture*

6.1.

$$p_i = (x_i, y_i, z_i, 1)^T \tag{6.1}$$

$$f : F \mapsto M, p_f \mapsto p_m \tag{6.2}$$

$$p_m =^F T_M \cdot p_f \tag{6.3}$$

Transformations f given in Equation 6.2 map one frame to another and can be displayed with transformation matrices (Equation 6.3).

6.5.3.2 Problem statement

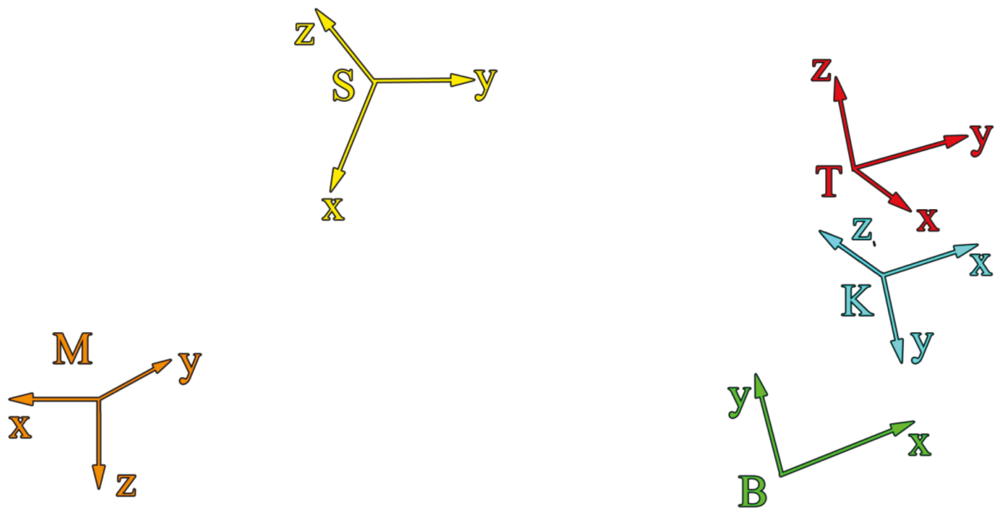


Figure 6.61: *Frames used for ARTab*

The different frames used in this application have been defined above and are depicted in Figure 6.61. At first, we need the position of the camera in the real world. For this the transformation matrix ${}^{M}T_{K}$ needs to be estimated:

- The tracking system gives ${}^{M}T_{T}$.
- Then the matrix ${}^{T}T_{K}$ is needed to calculate the frame K out of the tablet frame T with respect to the tracking systems origin. This cannot be done with the measurement of the distance, because the focus point of the camera lies in the tablet.

6.5.3.3 Camera calibration

A camera maps the three dimensional world on a two dimensional picture. This mapping is done with the camera parameters. In our example of a checkpoint picture with frame S, the extrinsic parameters define the translation matrix ${}^{S}T_{K}$. Another transformation with the intrinsic parameters is used to express the distortions caused by the camera lens. One can estimate those parameters with a camera calibration [117]. A known pattern - in our case a checkerboard - is captured several times in different constellations as depicted in Figure 6.62. We have used the Matlab camera calibration toolbox.

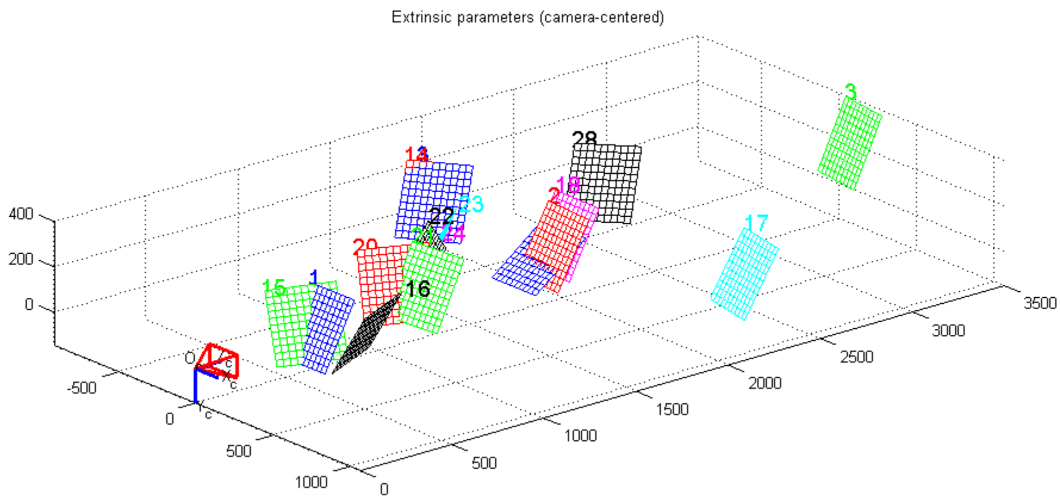


Figure 6.62: *Visualization of camera calibration for ARTab*

6.5.3.4 Camera offset calculation

In order to calculate the camera offset ${}^{T}T_{K}$ we use one picture, and estimated the camera offset out of this picture with

$$ {}^{T}T_{K} = {}^{T}T_{M} \cdot {}^{M}T_{S} \cdot {}^{S}T_{K} \tag{6.4} $$

The position of the checkerboard can be measured with the tracking system, too. This has been done for picture 1 in Figure 6.62. As depicted in Figure 6.63, the checkerboard frame can be calculated with the three measurement points highlighted with yellow. Thus, we know the position of the checkerboard frame with respect to the tracking system, the matrix ${}^{M}T_{S}$.

We now know every matrix of Equation 6.4, so that we can calculate ${}^{T}T_{K}$. For each new tablet position, the matrix ${}^{M}T_{K}$ can be estimated dynamically out of the

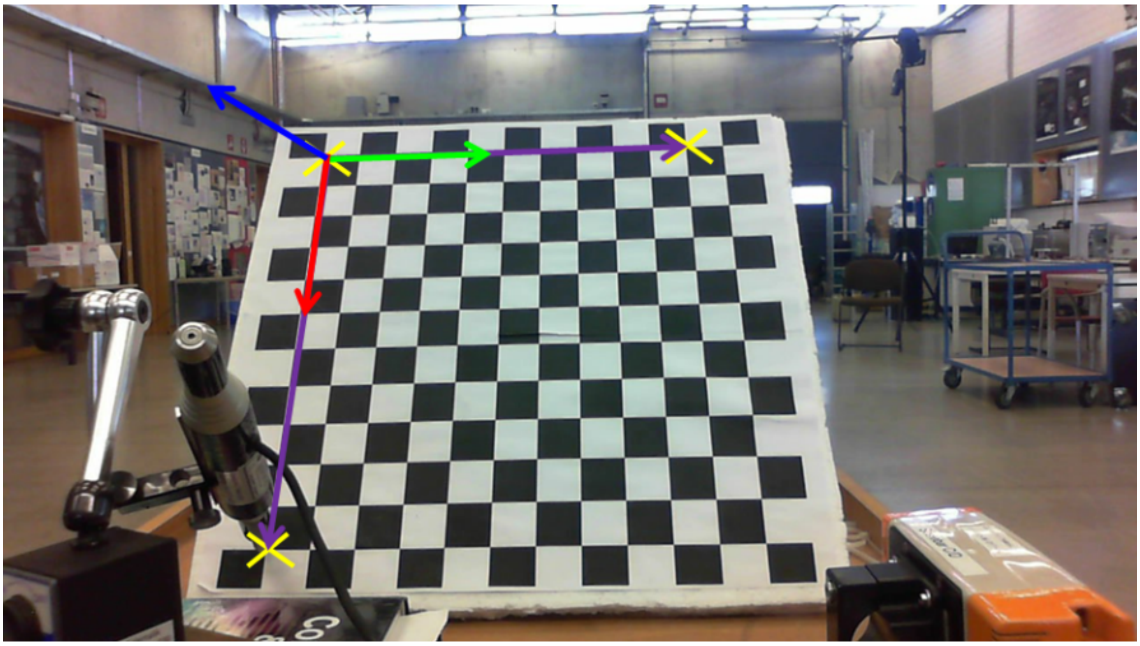


Figure 6.63: *Measuring checkboard frame*

iSpace position and the camera offset according to Equation 6.5.

$$^{M}T_{K} =^{M} T_{T} \cdot^{T} T_{K} \tag{6.5}$$

6.5.3.5 Error estimation

The system has been set up with the architecture described above. The system allows a user to walk around a virtual object perceived "through" a tablet PC. By using the system, we observed, that the virtual frame has an offset to the real world depending on the position relative to the defined origin.

In order to quantify this offset, we used a table in the origin in the real world (Figure 6.64) and a virtual object in the same size slightly transparent. On several positions we measured the offset between the real world picture and the virtual picture by using an additional translation matrix as offset (Figure 6.65). We plotted the error values in the 3D-graph in Figure 6.66.

6.5.4 User study

We performed a user study in order to measure the benefits of the ARTab compared to other possible viewpoint constellations. In this study, a positive influence of highlighting key items on determination and comprehension of complex construction systems should be evaluated. Furthermore, the generation of a positive effect by the possibility of virtually visiting the construction is determined. Besides that main objective, the eligibility of Augmented Reality versus Virtual Reality for a virtual

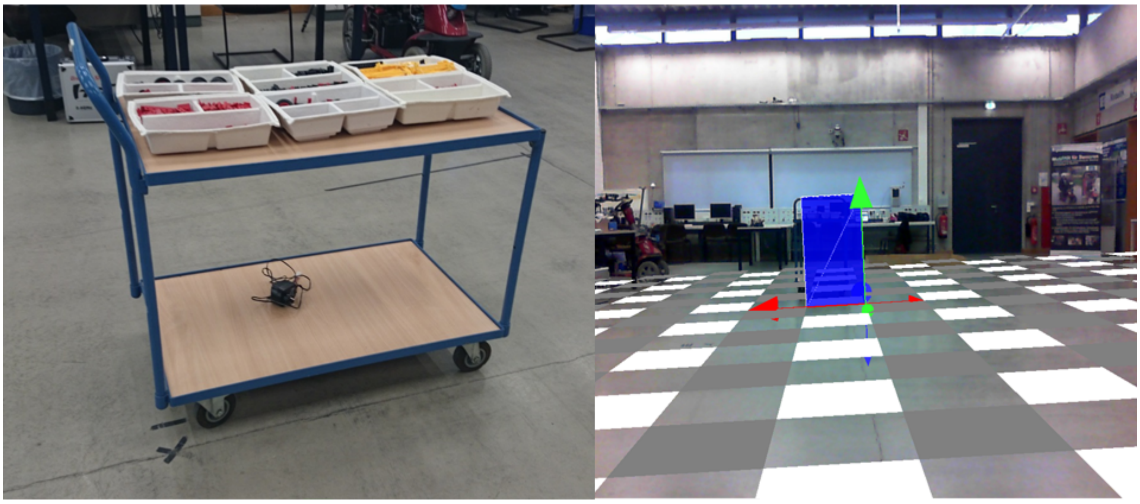


Figure 6.64: *Table in the real world (left) with virtual box (right)*

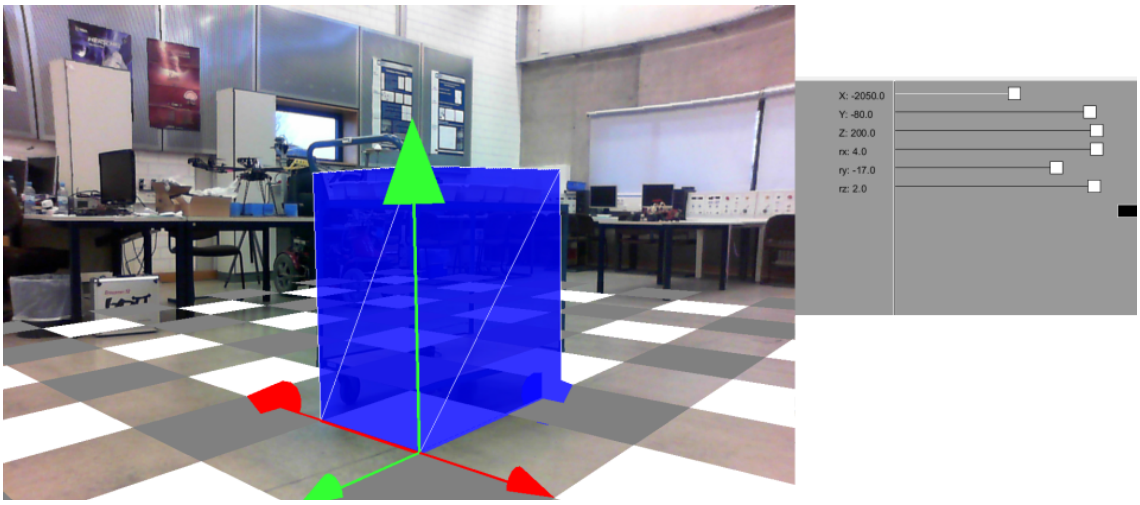


Figure 6.65: *Matrix adjustment tool on the right of the AR picture with virtual blue box on table*

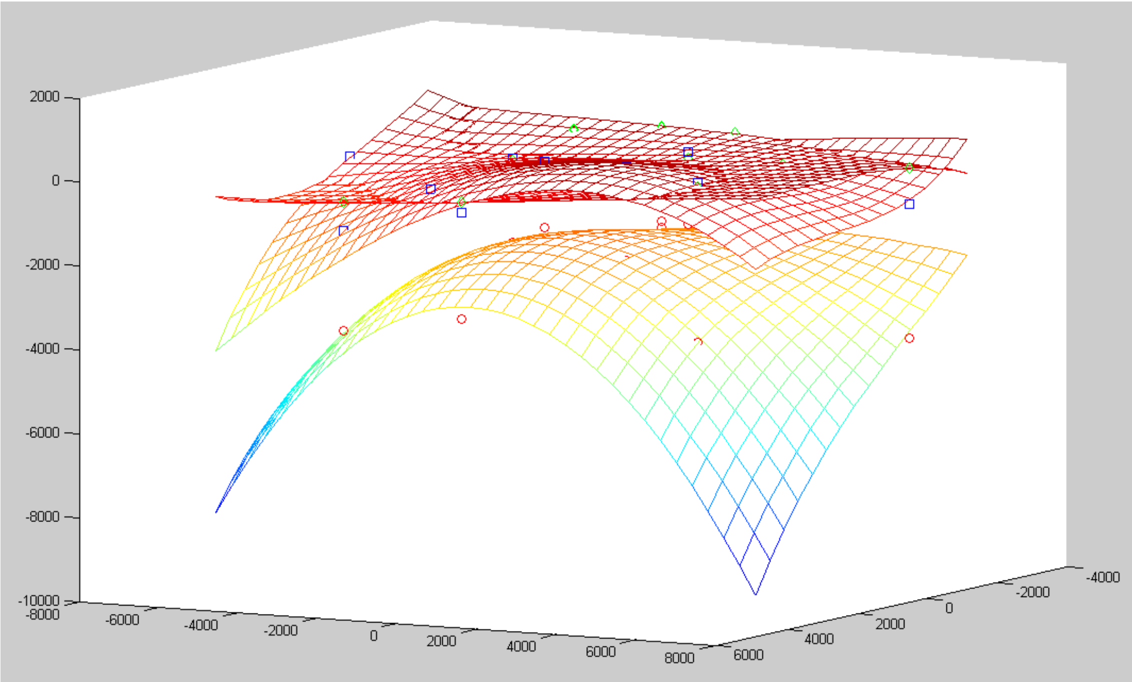


Figure 6.66: *Matrix offset (green: z, blue: y, red: x)*

visit is tested, and if better results can be achieved than by video. Moreover, the difference in results between an immobile view of the VR system and a video without highlighting should be identified.

To evaluate the quality of information, the situation awareness generated by the system with the SART questionnaire [89], individually perceived work load with the NASA TLX [60], and perceived usability with ISONORM [71] and QUESI [72] were measured. In addition, construction comprehension plays a pivotal role for evaluation, determined by the probands' performances in three exercises about construction organization and production process.

6.5.4.1 Method

Participants

60 individuals participated in this study, recruited from the institute's acquisition system or personally by the evaluator. Test persons were between 18 and 33 years old, average age was 22 years. The majority of probands was female (40). All participants were either enlisted over the "proband system" of the University of Würzburg or personally by the evaluator. Also, some of the test persons (11) already had had any kind contact with industry robots. Almost all of the participants regularly use computers (58), and their preferred operating system was Windows (47). Participants recruited via the acquisition system were credited one hour of experimentation time as a reward.

Experimental plan

According to the employer's directive, the experimental procedure began early February 2016 and lasted until end of June 2016. The time required was owed to some participants' absence and the systems' developing time. The majority of systems (VR and AR) were developed just during the conduction. Therefore, there was no previous analysis of the systems and a possible adaptation of tasks to the system was not accomplishable.

Due to technical reasons, the experiment was executed in Building M3 (Robotics hall) of the University. Devices mounted there were required to conduct experiments in the AR-group.

Experimental design

The experiment was drafted for a between-subject-design. For each of the systems there were two experimentation groups. These show different characteristics for the experimental conditions System with highlight and "mobile view" depicted in Figure 6.67:

- Group 1: Video ("Highlight": no, mobile view: no)
- Group 2: Video ("Highlight": yes, mobile view: no)
- Group 3: VR ("Highlight": no, mobile view: no)
- Group 4: VR ("Highlight": no, mobile view: yes)
- Group 5: AR ("Highlight": no, mobile view: yes)
- Group 6: AR ("Highlight": yes, mobile view: yes)

The trial was structured into three parts. In the first part, the test persons were given an introduction to the experiment and demographic data was surveyed. In the second part the participants were meant to solve tasks with the help of their system. Finally the participants finished further questionnaires.

The study's part with exercises comprised a 5 minute exploration phase (shown in Figure 6.68) as well as three exercises with a follow-up questionnaire each (NASA TLX). Since the participants did not have any knowledge about the construction, additionally to testing their comprehension a learning effect was ought to be achieved. Thus, exercises were not randomized.

1. Introduction
2. Exercise "Axis"
3. Exercise "Lego"
4. Exercise "Process"
5. Final questionnaires

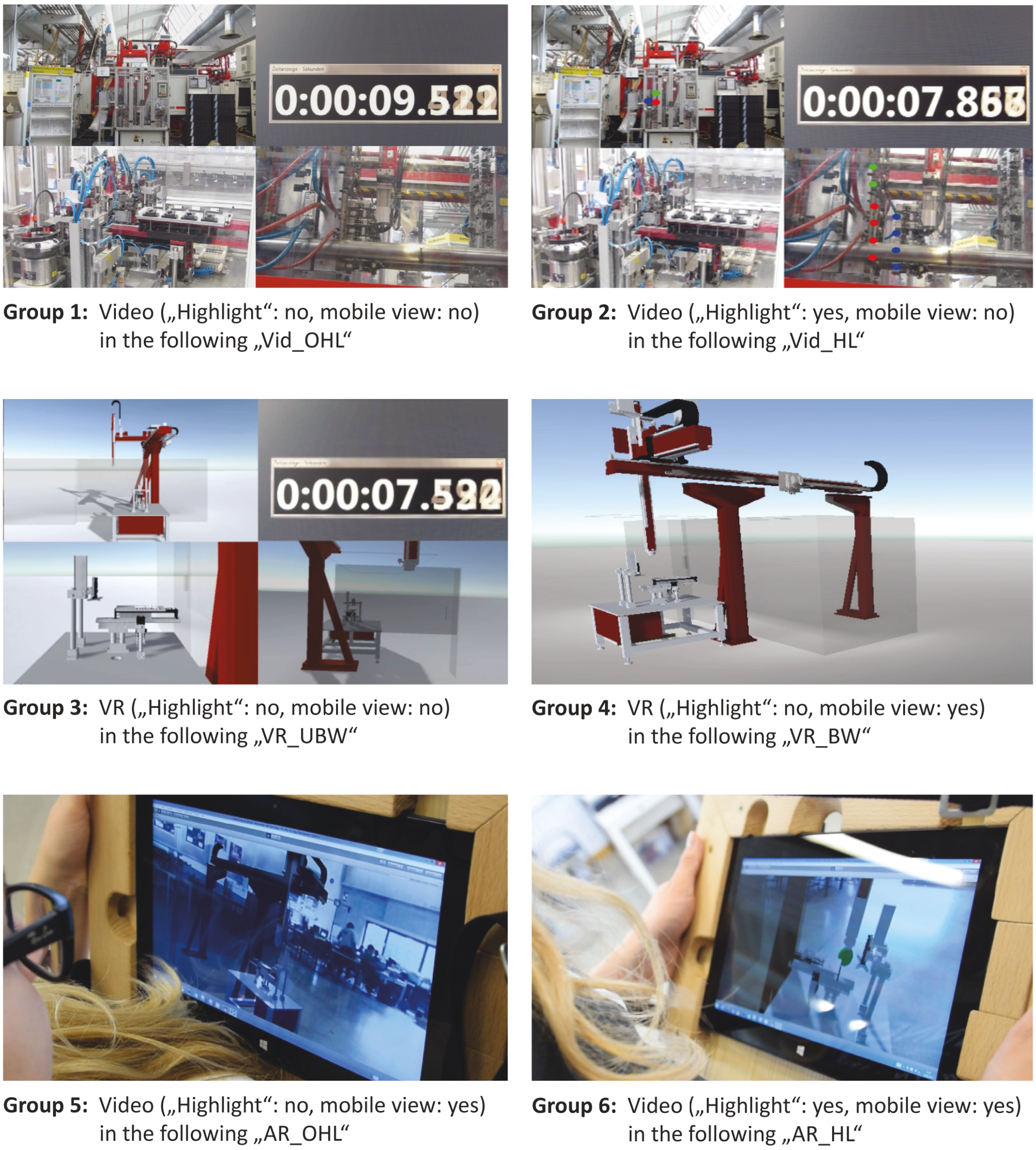


Figure 6.67: *Six experiment conditions for ARTab user study*

Figure 6.68: *Participant in exploration phase (freeplay) for ARTab user study*

During some experiments the probands recorded video tapes. These were not intended for data survey, but for advertisement and demonstration.

Material

Experiments largely were conducted on a All-In-One PC with Windows as operating system. Only the test person groups for the AR-system used a tablet with an attached tracking device instead of the computer. For the exercise "Lego" a model of the construction (not true to scale) was designed. A scaled display was not possible due to reasons of time and costs. Furthermore, the questionnaires NASA-TLX, Quesi and ISONORM 9241/110 were used.

Systems

Different shapes of three visual display forms were employed for the study: Video, Virtual Reality (below: VR), and Augmented Reality (below: AR). The systems allow observation of an industry machine producing toothbrush bases. The construction essentially consists of an injection molding machine, an assembly station and a robot. The systems VR and AR use virtual models of the construction, the video shows the real construction in operation.

Experimental groups were presented different forms of systems regarding the experimental conditions "system with highlight" and "mobile view" (depicted in Figure 6.67):

- Group 1: Video ("Highlight": no, mobile view: no) in the following "Vid_OHL"

- Group 2: Video ("Highlight": yes, mobile view: no) in the following "Vid_HL"
- Group 3: VR ("Highlight": no, mobile view: no) in the following "VR_UBW"
- Group 4: VR ("Highlight": no, mobile view: yes) in the following "VR_BW"
- Group 5: AR ("Highlight": no, mobile view: yes) in the following "AR_OHL"
- Group 6: AR ("Highlight": yes, mobile view: yes) in the following "AR_HL"

The condition "highlight" characterizes the marking of the produced parts' position by colored dots. The color represented the current production stage of the part. The condition "mobile view" describes if the participant is able to move within the system in any way.

- **System description video:**
  This system shows a video of the construction, recorded at the project partner of the ZfT. The participant was not able to actively move the camera in any of the configurations. The video was replayed using the VLC-Media Player.
- **System description VR:**
  The VR-system shows the construction, just the injection molding machine was only represented by clear grey box. Identification of the injection molding machine and related processes by the participants was thereby impaired. The participants were able to move with help of a PC keyboard in a Unity-based program, for the immobile configuration a video was generated in the program.
- **System description AR:**
  Participants were presented a specially prepared Tablet-PC. They were able to move actively around the construction, which was projected into the room. The tablet served as a "viewing window" into the virtual part of the AR. Organization was similar to that of the VR system.

6.5.4.2 Experimental procedure

The experiment was divided into three parts. In the first part, the participants were illuminated about their role in the following test, moreover, demographic data were surveyed. The subsequent part began with a five minute free-roaming-period and included three tasks, accompanied by a questionnaire each (NASA-TLX). Final data were recorded using the questionnaires QUESI and ISONORM.

Testing exercises

Within the frame of this study the test persons had to manage three tasks. The exercises were meant to display the process of construction analysis in a simplified way. Therefore, the tasks covered both the construction's organization and the production process.

The exercises were given in a fixed sequence with a required learning effect:

1. Task: "Axis" (Task group construction organization)
2. Task: "Lego" (Task group construction organization)
3. Task "Process" (Task group production process)

Task sequence was determined by complexity and the degree of necessary knowledge about the construction.

Figure 6.69: *Picture of the plant provided for the "Axis" task for ARTab user study*

- **Task "Axis"**
  This task was part of the task group "construction organization". The participant was shown a picture of the construction (depicted in Figure 6.69). Using this picture, the test person should identify the movement axis of the robot and their orientation. Correctly assigned main axis (X, Y, Z) and orientations were assessed. One point was given for each correctly chosen naming and orientation. The relation of achieved and achievable score was measured.

- **Task "Lego"**
  Several sectioned constructions made from Lego were presented to the participant. These had to be assembled to build a model of the construction. The participant's job was it to identify the correct one from various versions of the same part, and to insert it correctly into the full model. The correct choice of the part and the assembly of parts to a sample model designed by the author were evaluated. A participant performing this task is shown in Figure 6.70.

Figure 6.70: *Participant at the "LEGO" task for ARTab user study*

- **Task "Process"**
  The test person was presented his or her assigned system for a period of five minutes. Afterwards, the test person should retrace the assembly process in two steps with the help of a plan view. Initially, process steps were to be roughly marked. Then the participant should take notes on the interaction of the system in and between the process steps. The amount of correctly assigned post-its was counted (within a tolerance area, shown in Figure 6.71), as well as the correctly described actions. There was no need for technical terminology. For each correctly assigned action and placement, one point was given in the analysis respectively. The relation of achieved and achievable score was measured. This was the problem as in the domain expert study in Section 5.3.3.

  The content of this task should draw a connection to a previous study, where the task was given in a similar way. The former test persons were experts who had access to the collective system of the project MainTelRob. Without having background knowledge about the construction, the experts performed poorly in evaluation, in spite of having considerable expertise: Only one was able to describe the process correctly without previous knowledge.

Measured data
To assess the systems, three evaluation categories were introduced:

- construction comprehension
- situation awareness

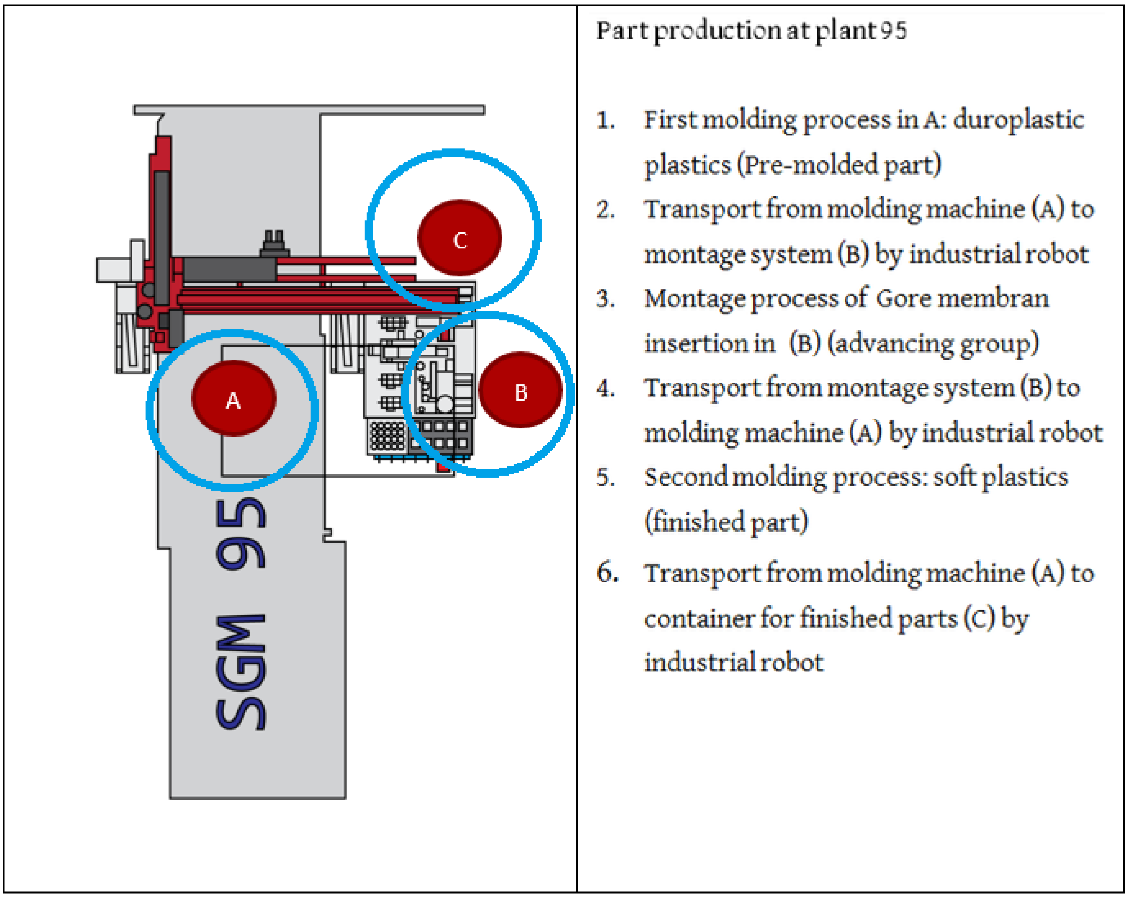


Figure 6.71: *Sample solution for task "Process" for ARTab user study*

- usability in general

To determine comprehension of the construction, solutions from the exercise part were used. To evaluate situation awareness, the questionnaires SART [89] and NASA-TLX [60] were employed. Using the QUESI-questionnaire [72] and criteria from the ISONORM 9241/110 [71], the general usability of the systems was rated.

6.5.4.3 Results

First of all, the measured results are plotted in various diagrams:

- **Category Situation Awareness:** The results on the SART questionnaire are presented in Figure 6.72. It can clearly be seen, that the Attentional Demand for the AR applications are considerably higher than the other experimental conditions, resulting in a lesser situation awareness value. The NASA TLX values for the different experiments are presented in Figure 6.73. The Lego experiment was the most demanding experiment, while the Video experimental conditions seem to be the most demanding experiment conditions.

- **Category Usability:** The results of the questionnaires ISONORM are depicted in Figure 6.75. Both VR conditions seem to receive substantial lesser values than the other experimental conditions, while the AR conditions provide throughout high values. The results of the QUESI questionnaire are shown in Figure 6.74. Altogether the static VR condition is considerably lower than the other conditions, through every QUESI category.

- **Category Construction comprehension:** The results of the experiment "Axis" can be seen in Figure 6.76, where the AR without highlighting seems to provide the best values. The results of the "Lego" experiment are shown in Figure 6.77. The VR conditions seem to provide the lowest average deviation of the components. The results of the experiment "Process" are depicted in Figure 6.78. The first three steps have been identified correctly by most of the experimental conditions. Similar to the user study in Section 5.3.3, the second injection molding process has not been identified by a lot of participants. While the highlighting on the video seems to show no consequences, the AR with highlighting allows some of the participants to identify the whole process correctly.

Secondly, the systems (Video and VR, VR and AR, VR immobile and Video without highlight) were compared by a one-factorial MANOVA (Multivariate analysis of variance [210]), comparison of the systems on base of the conditions "highlight" and "mobility" was conducted using a two-factorial MANOVA.

Comparison highlighting and mobility
A two-factorial MANOVA was employed to analyze, if highlighting items or a mobile camera have an effect on the three evaluation criteria, respectively. In the

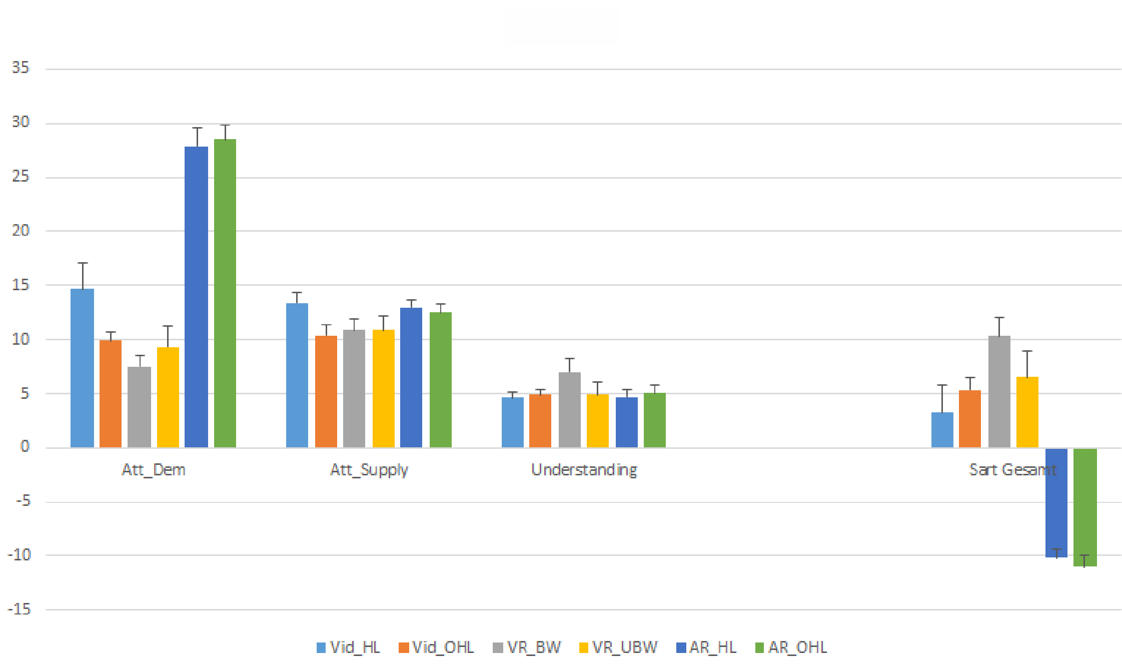


Figure 6.72: *Results of the SART questionnaire*

following, both factor's results as well as those of the combination of both, will be addressed.

- n: the number of test samples used
- M: the median of the values
- SD: the standard deviation of the values
- F: the calculation of significance dependent on the degrees of freedom
- p: the probability used for significance testing
- d: the effect size

Remarks to "Highlighting" and "Mobility"
For the statistical analysis on the question, if highlighting or a moving camera has an effect on the three evaluation criteria, a two-factorial MANOVA has been conducted. In the following, the results on those two factors individually and their combination is described. Because the VR systems cause an imbalance of the data (4 experimental conditions without highlighting compared to two conditions with highlighting), only the video systems and the AR system have been compared.

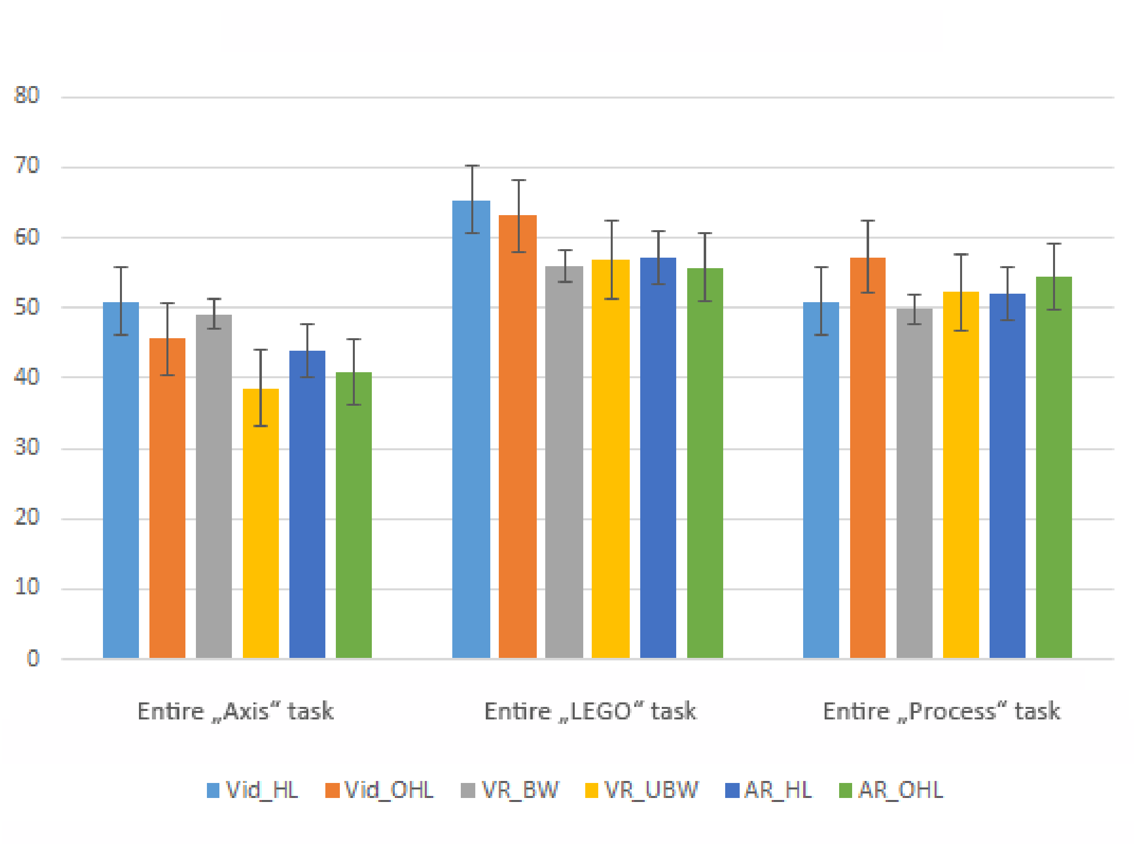


Figure 6.73: *Results of the TLX questionnaires for each task*

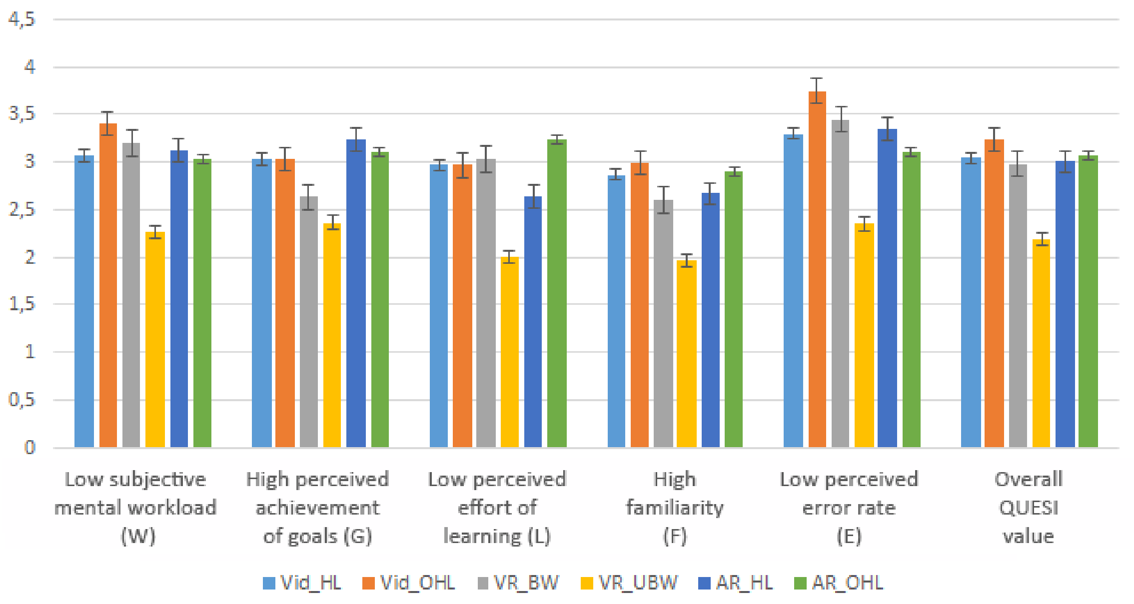


Figure 6.74: *Results of the QUESI questionnaire*

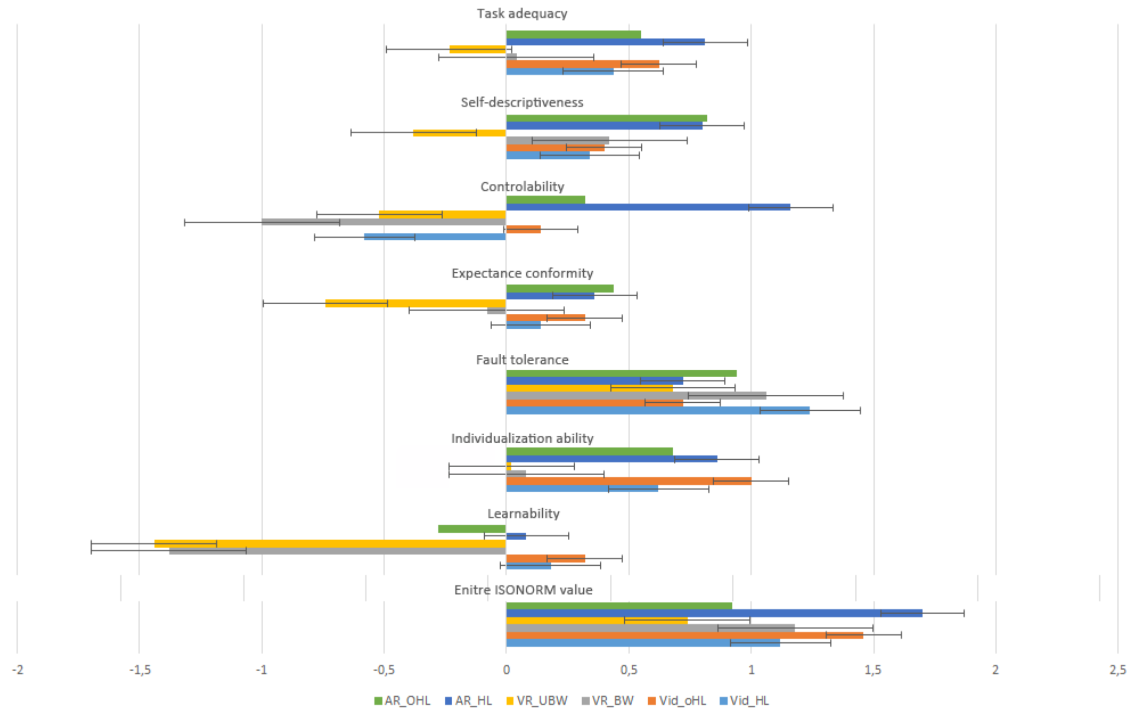


Figure 6.75: *Results of the ISONORM questionnaire*

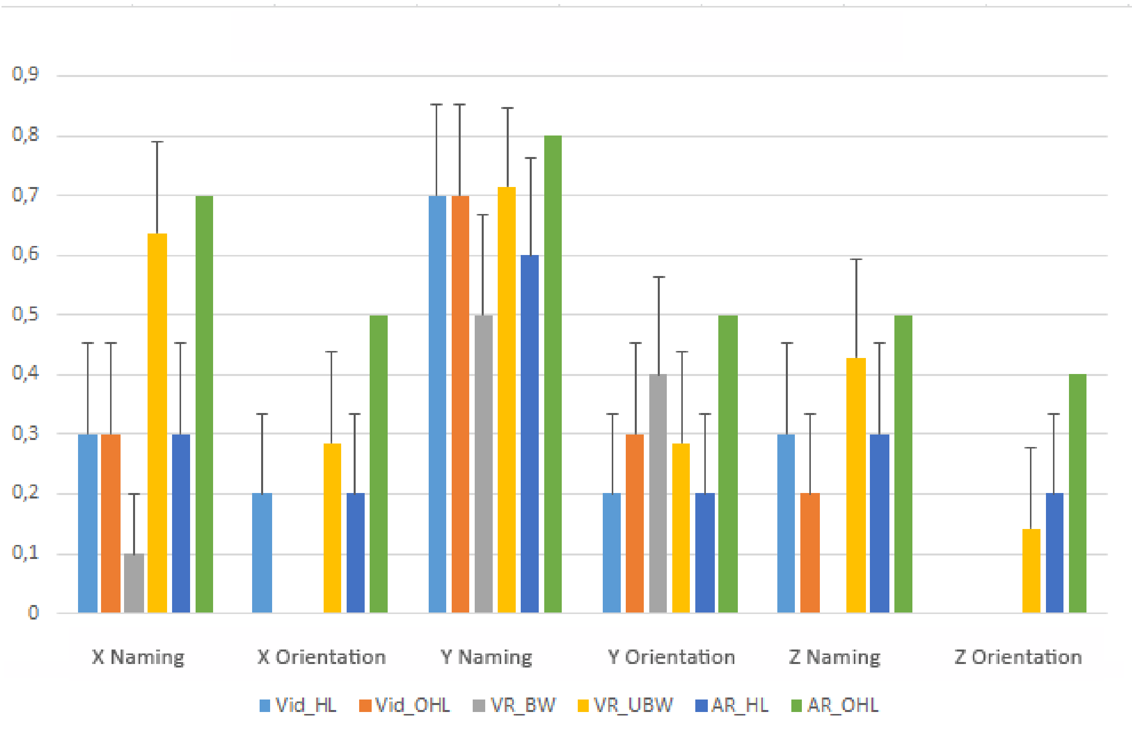


Figure 6.76: *Results of the axis identification task*

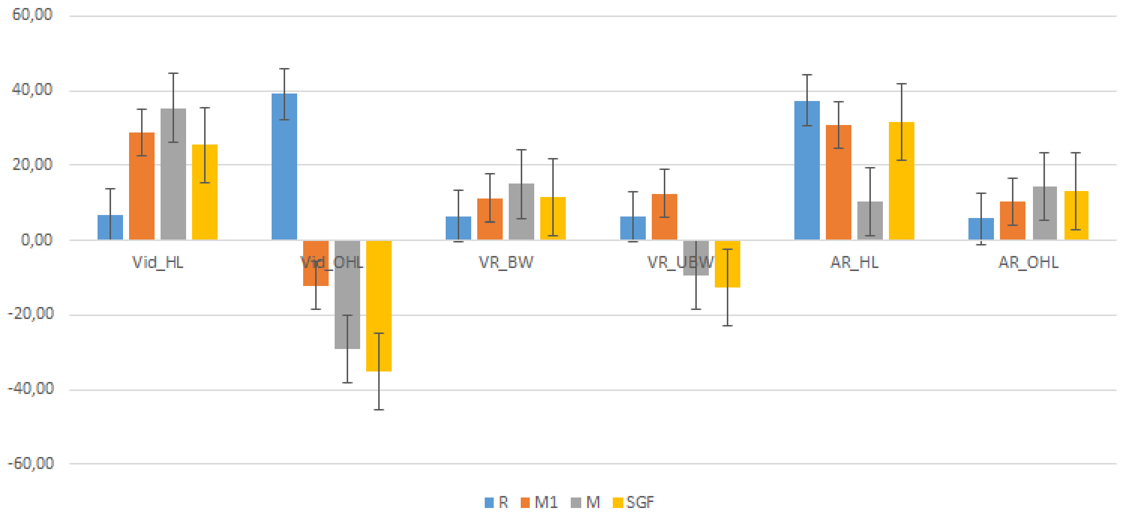


Figure 6.77: *Results of the lego task*

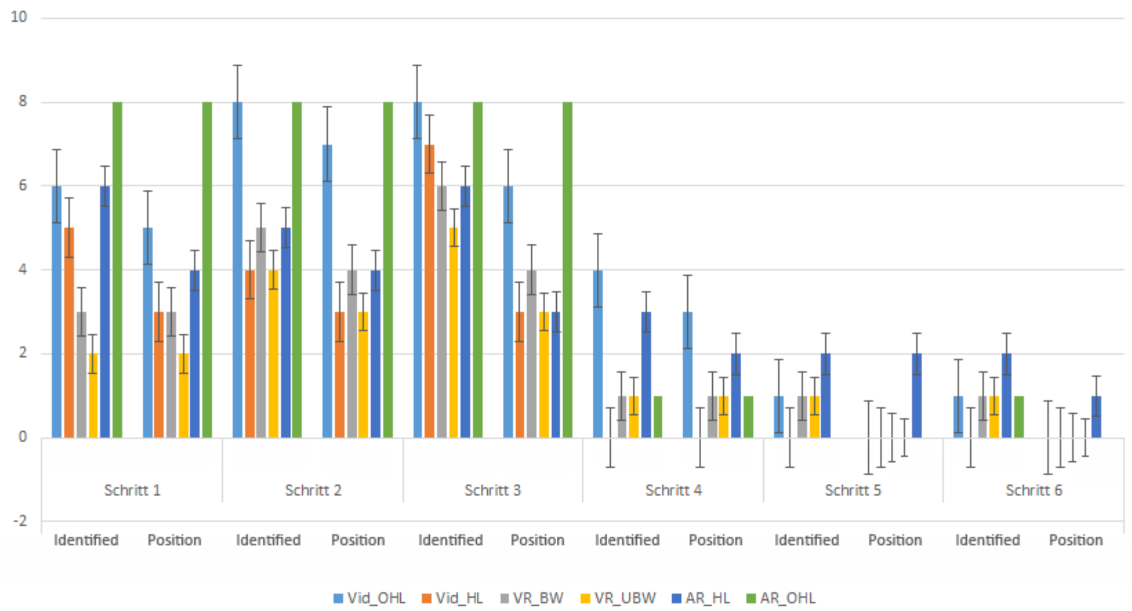


Figure 6.78: *Results of the process identification task*

1. **Factor "Highlighting".**
   - In the category situation awareness, the following results were returned: In the questionnaire SART, on average, systems with highlighting ($n = 20$, $M = -3.45$, $SD = 8.92$) were rated less than those without ($n = 20$, $M = -2.8$, $SD = 9.05$). In the questionnaire NASA-TLX, probands rated systems with highlighting ($n = 20$, $M = 51.46$, $SD = 15.63$) on average better than those without ($n = 20$, $M = 51.46$, $SD = 15.63$). No stage of the factor "Highlight" has an advantage for the category situation awareness. There was a significant difference shown by MANOVA ($F(2, 35) = 0.63$, $p < 0.05$,$d = 0.12$). Nonetheless, this is not to be understood as a significance, since a Box's M test was significant with a value of 2.256 ($p = 0.003$).
   - The following results were gained in the category usability: The QUESI questionnaire showed systems with highlighting ($n = 20$, $M = 3.15$, $SD = 0.82$) on average being rated better than those without ($n = 20$, $M = 3.03$, $SD = 0.89$). Likewise, using the ISONORM 9241/110 questionnaire, systems with highlighting ($n = 20$, $M = 0.62$, $SD = 0.76$) were evaluated better than those without ($n = 20$, $M = 0.59$, $SD = 0.65$). In this category therefore systems with highlighting are advantageous. However, no significant difference was found ($F(2, 35) = 0.76$, $p < 0.05$,$d = 0.009$).
   - In the category construction comprehension, due to missing data the group size of the group was reduced to 18 participants. The following results were sampled: During the task "axis", participants could reach averagely 38% of the achievable score ($M = 0.38$, $SD = 0.29$) in the group "highlighting", whereas the test persons of the group without highlighting performed poorer, with 34% of the achievable score on average ($M = 0.34$, $SD = 0.24$). In the task "Lego" and the group with highlighting, the components' position averagely deviated approximately 8.42 units from example position ($M = 8.4202$, $SD = 5.57$), while the mean deviation in the other systems was a bit more with approximately 9.8 units ($M = 9.80$, $SD = 5.65$). During the task "process", systems with highlighting achieved a mean of 28% of the possible score ($M = 0.28$, $SD = 0.30250$), and those without gained approx. 43% ($M = 0.43$, $SD = 0.37$). Thereby, participants using systems with highlighting performed better overall. No significant difference was found, however ($F(3, 30) = 0.53$, $p < 0.05$, $d = 0.19$).

   In two of three categories, system with highlighting achieved better results than those without highlighting. A MANOVA employing all evaluation categories did not yield a significant difference ($F(7, 26) = 0.71$, $p < 0.05$, $d = 0.23$).

2. **Factor "Mobility" (which is equal to the comparison of AR and video systems).**

- In the category situation awareness, the following results were produced: In the questionnaire SART, systems using mobile perspectives n ($n = 20$, $M = -10.60$, $SD = 2.85$) were rated on average poorer than those not using mobile views ($n = 20$, $M = 4.35$, $SD = 6.00$). Test persons evaluated systems of the category "mobile" ($n = 20$, $M = 53.25$, $SD = 15.77$) in the NASA-TLX questionnaire on average better than systems without movement ($n = 20$, $M = 54.08$, $SD = 17.96$). No stage of the factor "mobility" has an advantage for the category situation awareness. MANOVA showed a significant difference ($F(2, 35) < 0.01$, $p < 0.05$, $d = 1.00$). Nonetheless, since the Box's M test was significant with a value of 2.26 ($p = 0.02$), this means that this is not to be rated as significant.
- In the category usability, the following results were obtained: In the QUESI questionnaire, systems with mobile perspective ($n = 20$, $M = 3.04$, $SD = 0.81$) were rated on average better than those without ($n = 20$, $M = 3.14$, $SD = 0.90$). Likewise evaluation of the ISONORM 9241/110 questionnaire gained a better rating for systems with ($n = 20$, $M = 0.68$, $SD = 0.55$) than those without mobile perspective ($n = 20$, $M = 0.53$, $SD = 0.82$). In this category, therefore systems with mobile perspective perform stronger. Nevertheless, no significant difference could be shown ($F(2, 35) = 0.49$, $p < 0.05$, $d = 0.16$).
- In the category construction comprehension, for the task "Lego" the group size of the group "mobile" was reduced to 19 participants due to missing data, group size of "immobile" was reduced to 17. These results were obtained: During the task "axis", participants of the group "mobile" were able to achieve on average 40% of the possible score ($n = 19$, $M = 0.40$, $SD = 0.28$), while participants of the group "immobile" performed poorer, with averagely 32% of possible points ($n = 17$, $M = 0.32$, $SD = 0.24$). In the task "Lego", the groups with mobile perspective showed deviated positions of components of on average approximately 10 units ($n = 19$, $M = 10.42$, $SD = 6.05$) from the example positions, mean deviation in other systems was a bit smaller with approximately 7.6 units ($n = 17$, $M = 7.65$, $SD = 4.74$). In the task "process", systems with mobile perspective gained on average 46% of achievable points ($n = 19$, $M = 0.46$, $SD = 0.40$) and those without mobile perspective reached 24% of possible points ($M = 0.24$, $SD = 0.21$). Overall evaluation showed the participants using systems with mobile perspective performing better in this category. No significant difference could be seen ($F(3, 30) = 0.19$, $p < 0.05$, $d = 0.400$).

In two of the three categories, systems with the condition "mobile" came out on top. A MANOVA employing all evaluation categories did obtain a significant difference ($F(7, 26) = 0.01$, $p < 0.05$, $d = 1.00$).

3. **Factor combination "highlighting" and "mobility".**

- In the category situation awareness the following results were found: In the SART questionnaire, systems with the conditions "without highlighting" and "immobile" were rated best ($n = 10$, $M = 5.40$, $SD = 3.66$). In the NASA-TLX questionnaire, systems with the conditions "with highlighting" and "immobile" were assessed best ($n = 10$, $M = 50.92$, $SD = 19.23$).

  In this rating category, systems with the conditions "immobile" and "with highlighting" or "without highlighting", respectively, were rated best. Still, no significant differences were found ($F(2, 35) = 0.57$, $p < 0.05$, $d = 0.14$).

- In the category usability, the following results were seen: In the QUESI questionnaire, systems with the conditions "with highlighting" and "immobile" were rated highest ($n = 10$, $M = 3.32$, $SD = 0.78$). In the ISONORM 9241/110 questionnaire, systems with the conditions "with highlighting" and "mobile" were assessed best ($n = 10$, $M = 0.81$, $SD = 0.48$). In this category, no factors have performed clearly best. However, no significant differences were to be found ($F(2, 35) = 0.53$, $p < 0.05$, $d = 0.15$).

- In the category construction comprehension, the following results were seen: In the task "axis", participants using the conditions "without highlighting" and "mobile" performed best ($n = 10$, $M = 0.43$, $SD = 0.23$). In the task "Lego", test persons using a system with the conditions "with highlighting" and "immobile" ($n = 9$, $M = 6.92$, $SD = 3.99$) gave the best performance. In the task "process", test persons using a system with the conditions "without highlighting" and "mobile" ($n = 10$, $M = 0.57$, $SD = 0.42$) showed the best performance. System with the conditions "with highlighting" and "immobile" performed best in this category (two out of three cases). No significant differences were seen ($F(3, 30) = 0.57$, $p < 0.05$,$d = 0.18$).

None of the combinations could be rated clearly better than the others. A MANOVA over all rating categories showed no significant difference ($F(7, 26) = 0.80$, $p < 0.05$, $d = 0.19$).

4. **Comparison Video and VR systems**

- In the category situation awareness, the following results were returned: In the SART questionnaire, the video groups ($n = 20$, $M = 4.35$, $SD = 6.00$) rated worse than the VR groups ($n = 20$, $M = 8.15$, $SD = 6.75$). Also in the NASA-TLX questionnaire, the video groups ($n = 20$, $M = 54.08$, $SD = 17.96$) assessed ordinarily poorer than the VR groups ($n = 20$, $M = 51.04$, $SD = 16.66$). In overall rating of this category, the VR system therefore performed better. Still, there was no significant difference shown by MANOVA ($F(2, 35) = 0.143$, $p < 0.05$).

- In the category usability, the following results were obtained: In the QUESI questionnaire, videos was on average rated better ($n = 20$,$M = 3.14$, $SD = 0.90$) than the VR systems ($n = 20$, $M = 2.59$, $SD = 0.96$). Also the ISONORM questionnaire showed better values on average for the video systems ($n = 20$, $M = 0.53$, $SD = 0.82$) than the VR systems ($n = 20$, $M = -0.097$, $SD = 0.83$). Overall, the video system showed therefore better results in the category usability. A MANOVA in this category did not return significant results ($F(2, 35) = 0.07$, $p < 0.05$).
- In the category construction comprehension, due to missing data the video group size for the task "Lego" was reduced to 17 participants and the group size for VR to 19 participants. The following results were returned: In the task "axis", the participants in the video group were able to reach approx. 31% of possible points on average ($n = 17$, $M = 0.32$, $SD = 0.24$), while the participants of the VR group performed a bit poorer with approximately 22% ($n = 19$, $M = 0.22$, $SD = 0.25$). In the task "Lego", the component positions deviated in the video group only by approximately 7.7 units ($n = 17$, $M = 7.65$, $SD = 4.74$) from the example position, while the average deviation in the VR groups was a bit larger with approximately 7.8 units ($n = 19$, $M = 7.78$, $SD = 4.65$). In the task "process", the video groups reached on average 24% of the possible score ($n = 17$, $M = 0.24$, $SD = 0.21$), and the VR groups gained approx. 28% of achievable points ($n = 19$, $M = 0.28$, $SD = 0.25$). In overall rating of this category, the participants of the video groups performed better. Statistical analysis (MANOVA) in this category did not show any significant difference between both groups' performances ($F(3, 30) = 0.64$, $p < 0.05$).

In general, the video system performed better in two of three categories. An analysis of all categories, however, did not obtain a significant difference ($F(7, 26) = 0.28$, $p < 0.05$, $d = 0.54$).

5. **Comparison of VR and AR systems**

- In the category situation awareness, the following results were returned: In the SART questionnaire, the VR group ($n = 20$, $M = 8.15$, $SD = 6.75$) were rated better than the AR groups ($n = 20$, $M = -10.60$, $SD = 2.85$). Also in the questionnaire NASA-TLX, the VR groups ($n = 20$, $M = 51.04$, $SD = 16.66$) rated better on average than the AR groups ($n = 20$, $M = 53.25$, $SD = 15.77$). In overall rating of this category therefore the VR system performs better. A significant difference was found employing a MANOVA ($F(2, 35) = 0.000$, $p < 0.05$,$d = 1.000$).
- In the category usability the following results were obtained: In the QUESI questionnaire the AR system was on average rated better ($n = 20$,$M = 3.04$, $SD = 0.81$) than the VR system ($n = 20$, $M = 2.59$,

$SD = 0.96$). Likewise, rating by the ISONORM 9241/110 questionnaire showed a better performance of AR systems ($n = 20$, $M = 0.68$, $SD = 0.55$) than of VR systems ($n = 20$, $M = -0.10$, $SD = 0.83$). Thus, the AR system yielded a better result for the category usability. A MANOVA in this category did not return any significant differences, but is close to it ($F(2, 35) = 0.05$, $p < 0.05$, $d = 0.87$).

- In the category construction comprehension, AR and VR group size for the task "Lego" was reduced to 19 participants respectively, due to missing data. The following results were obtained: During the task "axis", participants were able to reach on average app. 39% of the achievable score ($n = 19$, $M = 0.40$, $SD = 0.28$), while participants in the VR groups performed a bit less successful with approx. 22% of possible points on average ($n = 19$,$M = 0.22$,$SD = 0.25$). In the task "Lego", the components' positions deviated approximately 10 units in the AR groups ($n = 19$, $M = 10.42$, $SD = 6.05$) from the example position, while the mean deviation in the VR groups was a bit less with nearly 8 units ($n = 19$, $M = 7.78$,$SD = 4.65$). In the task "process", the AR groups achieved a mean of 45% of possible points ($n = 19$, $M = 0.45$, $SD = 0.40$), and the VR groups had a score of 28% ($n = 19$, $M = 0.28$, $SD = 0.25$). Thus participants of the AR group performed better in overall rating of this category. Statistical analysis (MANOVA) nonetheless showed no significant differences in this category between performance of both groups ($F(3, 30) = 0.11$, $p < 0.05$, $d = 0.50$).

In general, the AR system came out on top in two of three categories. Analysis of all evaluation categories yielded a significant results ($F(7, 26) = 0.000$, $p < 0.05$, $d = 1.00$) with a high effect size.

6. **Comparison VR immobile/ Video without highlighting**

- In the category situation awareness the following results were found: In the SART questionnaire the video groups were rated less ($n = 10$, $M = 5.40$, $SD = 3.66$) than the VR groups ($n = 10$, $M = 6.50$, $SD = 7.91$). Also in the NASA-TLX questionnaire, video groups performed worse ($n = 10$,$M = 57.25$, $SD = 17.00$) than the VR groups ($n = 10$, $M = 52.25$, $SD = 12.35$). In overall rating of this category therefore the VR system showed better results. Still, no significant difference was found by MANOVA ($F(2, 35) = 0.65$, $p < 0.05$,$d = 0.11$).
- In the category usability, these results were returned: In the QUESI questionnaire, video was on average rated better ($n = 10$, $M = 3.23$, $SD = 0.78$) than the VR system ($n = 10$,$M = 2.19$, $SD = 0.75$) . Evaluation by ISONORM 9241/110 questionnaire gave similar results with the video system performing better ($n = 10$, $M = 0.62$, $SD = 0.71$) than the VR system ($n = 10$,$M = -0.23$, $SD = 0.86$). In total, the video system thus gained a better result in the category usability. A MANOVA

in this category showed a significant difference ($F(2,35) = 0.03$, $p < 0.05$,$d = 0.68$).

- In the category construction comprehension, due to missing data the video group size in the task "Lego" was reduced to 8, the size of the VR group was reduced to 9 participants. These results were found: In the task "axis", participants from the video group were able to reach 22% of possible points ($n = 8$,$M = 0.23$, $SD = 0.21$), while participants on the VR groups showed a slightly worse result with 21% of possible points ($n = 9$,$M = 0.21$, $SD = 0.24$).

  In the task "Lego", the components' positions in the video group deviated on average approx. 8.4 units from the example position ($n = 8$,$M = 8.46$, $SD = 5.63$), and the mean deviation in the VR groups was a bit smaller with approximately 8.3 units ($n = 9$,$M = 8.3212$, $SD = 5.36355$). In the task "process", the video groups achieved approximately 25% of the possible score ($n = 8$,$M = 0.25$, $SD = 0.20$) and the VR groups yielded approximately 40% ($n = 9$, $M = 0.41$, $SD = 0.29$). In total rating of this category, the VR group participants therefore performed better. Statistical analysis (MANOVA) of this category did not result in significant difference between both groups ($F(3,30) = 0.574$, $p < 0.05$, $d = 0.157$).

The experimental group "VR, immobile" returned better results in two of three categories. Analysis of all evaluation categories did not show a significant difference between the systems, however ($F(7,26) = 0.30$, $p < 0.05$, $d = 0.33$).

6.5.4.4 Discussion

	Categories						Situation Awareness		Usability		Comprehension				
	Vid_OHL	Vid_HL	VR_BW	VR_UBW	AR_HL	AR_OHL	SART	NASATLX	QUESI	ISONORM	Axis	Lego	Process	Overall	Effect size
1							OHL > HL	OHL < HL	OHL < HL	OHL < HL	OHL < HL	OHL < HL	OHL > HL	OHL < HL	0.23
2							Vid > AR	Vid < AR	Vid < AR	Vid < AR	Vid < AR	Vid > AR	Vid < AR	Vid < AR	1.00
3							OHL + UBW	HL + UBW	HL + UBW	HL + BW	HL + UBW	HL + UBW	OHL + BW	Keine	0.19
4							Vid < VR	Vid < VR	Vid > VR	Vid > VR	Vid > VR	Vid > VR	Vid < VR	Vid > VR	0.54
5							AR < VR	AR < VR	AR > VR	AR > VR	AR > VR	AR > VR	AR < VR	AR > VR	1.00
6							Vid < VR	Vid < VR	Vid > VR	Vid > VR	Vid > VR	Vid > VR	Vid < VR	Vid > VR	0.33

Figure 6.79: *Summary of the results of the AR Tab user study: red and green marks the experimental conditions compared, yellow marked entries show significant values ($p < 0.05$), light green marks ($p = 0.05$), blue marked entries show significant values, where the Box M-Test also has been significant*

The summary of the results is presented in Figure 6.79. Of the six experimental conditions, different combinations have been compared. The comparison (1) of the experiments with highlighting (Vid_HL and AR_HL) to those without (Vid_OHL and AR_OHL) provides no significant result. In the category Situation Awareness a significance has been found, which needed to be rejected. The hypothesis that systems with highlighting would perform better could not be confirmed by the collected data.

If we compare the video systems to the systems using Augmented Reality (comparison 2), a significant superiority of the AR systems has been found. This underlines the hypothesis of AR systems performing better than video systems. Indeed, MANOVA over the total width of evaluation categories returned a significant result, but with the single categories showing no significant differences in the tests, an alpha error can not be excluded. The result on the category "Situation Awareness" needed to be rejected due to the significance of the Box M-Test.

No significant result has been found at the comparison of the video systems to those using VR (comparison 4), but the comparison shows a high effect size. The comparison on "VR immobile" and "Video without highlighting" (comparison 6). In the latter comparison, the usability of the video system seems to have exceeded the VR system.

If we compare the AR systems to the VR systems (comparison 5), the AR system performs significantly better. The VR system seems to provide a significant better situation awareness, whereas the usability of the AR system tends to be rated better. On the different tasks, the superiority of the AR system could not be significantly proved.

The multi-factorial analysis (comparison 3) can answer some of the questions together with the other results: The immobile perspectives seems to provide a better situation awareness because of the lower attentional demand. Concerning usability, the highlighting of the relevant product parts seem to help. Coming to the tasks, the combination of highlighting and static perspectives seems to help on the tasks "Axis" and "Lego", but not for the process understanding. In contrast to the hypothesis, the highlighting seems not to help for process understanding. As none of those results have been significant, this is merely the observation of trends and of the multi-factorial analysis.

Altogether this preliminary study compared the specific implementations, not the interaction methods. It has been an application oriented experimental set-up with systems with huge differences, using non-domain experts. There are a lot of questions for future work like the influence of latency, the specific highlighting methods or suitable video perspectives.

6.6 Summary

This chapter described the telemaintenance architecture that are built upon the innovations of the last two chapters. The mobile device served as communication hardware to the service technician and provided shared visual context to both users.

The expert was able to access the teleanalysis features as well as the newly developed possibility for teleoperation. The applications had been tested sucessfully in a real world remote maintenance scenario.

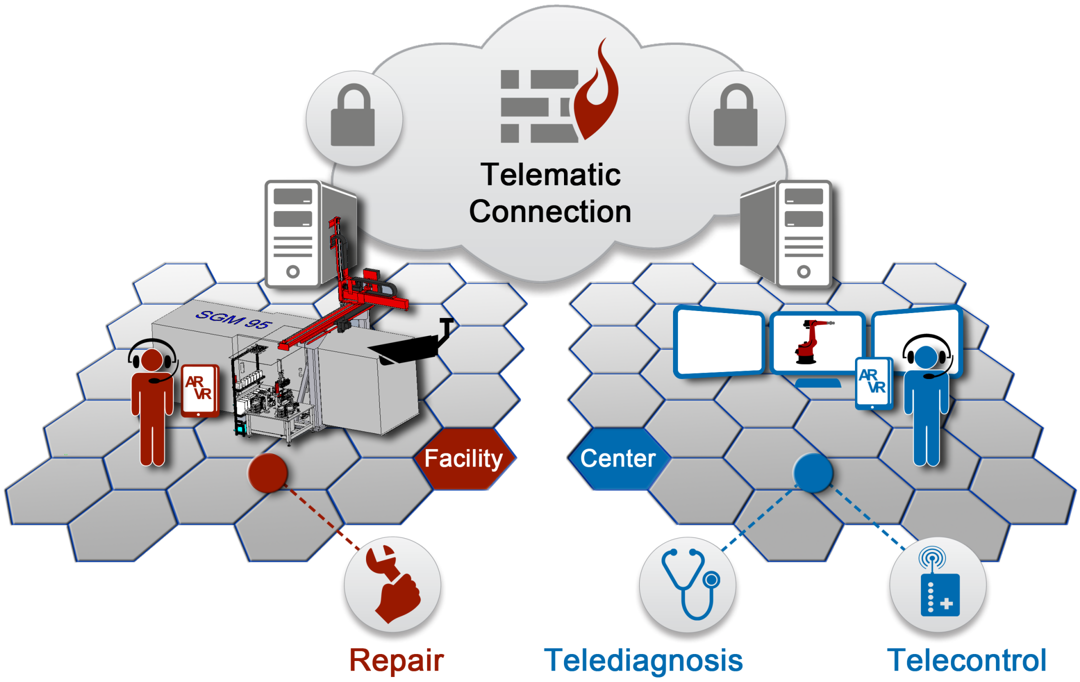


Figure 6.80: *Summary of the innovations: Telemaintenance as Human Supervisory Control*

6.6.1 Telemaintenance Architecture

We developed an architecture for telemaintenance and teleoperation of an industrial manipulator in an active production plant in Section 6.2. It uses a layer based connection approach called "Adaptive Manangement and Security System" (AMS) consisting of InteractionLayer, ApplicationLayer, ServiceLayer and TelematicConnectionLayer. The section described the theoretical approach as well as different steps of infrastructure development. As this framework was also used for networked control, the different delays and timings have been regarded in order to maintain a feasible user interaction.

6.6.2 Interfaces and Video Interaction

The developed interfaces on the ApplicationLayer are presented in Section 6.3. The service technician has been confronted with a machine failure and has problems to identify and correct it. He or she tries to contact the remote expert. That person needs to get a view on the working site. In order to provide a shared visual context for the repair task, a tablet PC with a camera is used by the service technician. A shared video screen can be used by both the service technician and the expert and the application provides the ability to paint over the screen in order to make annotations or clarify the spoken instructions. The application can be supported by computer vision tracking algorithms and therefore can also be used on the HMD device introduced in Chapter 4.

6.6.3 Proof of concept

The final experiment of the "MainTelRob" project, described in Section 6.4, was to perform a motor exchange with the help of the telemaintenance infrastructure. A repair operation was chosen, which has not been performed remote yet. The exchange of a motor of one manipulator axis would normally require the expert to travel to the production site. Both service technician and expert had not worked with the software before, but were able to conduct the repair successfully in cooperate work. The tools developed in the project and described in this thesis proofed successful and helpful for the users. The expert had been provided with three camera perspectives: The top camera, the camera of the tablet PC and one camera in the plant. Additionally, the service technician was provided with two action cams at head and breast and two other cameras were used. Those camera perspectives had not been provided to the expert during the experiment. In an additional experiment based on the video material, the best perspectives with regard to situation awareness were evaluated: The breast camera proofed best, followed by the top perspective and the head camera. This is why we suggest to provide a mobile video perspective with a camera adjusted to the body of the service technician.

6.6.4 AR Tab

The "AR Tab" is an off-site AR-based visualization approach described in Section 6.5 to provide a remote expert with an AR-view of the machine to be maintained. The remote optimization software developed in this project is meant to enable early detection of problems and optimization options of industry plants without the need of the expert's local presence. Since the remote technician has to be enabled to correctly identify the process and to comprehensively understand the construction process in total, the precise display of the machinery is crucial as is the correct view and perspective of the expert to maximize his or her situation awareness.

We conducted a user study for situation awareness as a function of viewpoint and application. We used several static viewpoints with real and virtual content as well as a dynamic virtual environment and the dynamic augmented environment "AR Tab", where a user can walk around the plant with industrial manipulator. The results underline the hypothesis of the specific AR systems performing better than the used VR systems. A similar result between was found by a significant superiority of AR system compared to the video perspectives.

A strong but not significant effect on the evaluation criteria has been found for the superiority of the influence of highlighting of items and a movable perspective. A combined analysis of those two factors did not show a clear result on all of the evaluation categories, but may provide useful insights for future system developments. The unclear results of statistical analysis can maybe be explained by the high system differences.

6.6.5 Economical benefits

In the project "MainTelRob" an economical analysis of the sixth scenarios have been performed on base of the project results in the plant of P&G.

The most important factor for scenario 6 (telemaintenance) are the travel costs. With external help a relative unexperienced service technician can conduct a difficult repair operation. Each service operation can save approximately 1000 Euro with the use of telemaintenance. In the involved facility, another approach has been chosen: The service personnel is very skilled and experienced and can solve most of the occurring incidents on their own. But it can serve as a debugging facility for other production sites, which strengthens the relevance of the topic of telemaintenance in general.

The production plants in the involved facility work 24 hours, 7 days a week, 350 days in a year. The research project has shown the importance of the "Industry 4.0" process. A team has been established which will further discuss those matters in the context of the "Molding" department. This team consists of SPS programmers, computer scientists, maintenance planers and managers.

Overall, the project partner assumes, that by using the proposed systems and concepts, the maintenance costs can be reduced by 15% per year.

6.6.6 Conclusion

This chapter showed the different components needed for telemaintenance acting together in one live scenario. First, the architecture of the system was described. It consists of a theoretical concept, the Adaptive Management and Security System (AMS) and a series of prototypes which implemented the infrastructure in different testbed steps and finally in the production environment. One essential part of the user interaction during a telemaintenance procedure is a shared visual context provided by the "Video Interaction" tool. The mobile device described in chapter 4 is used to enable a common video stream of the tablet camera which can be viewed and annotated by both the service technician and the expert. In the proof of concept of the infrastructure, an experiment of a motor exchange telemaintenance situation, this tool proved very useful. The experiment showed, that the infrastructure was working successfully and that the service technician was feeling very content in this scenario.

The expert nevertheless still lacked situation awareness and refused to teleoperate the robot (although it would have been possible by the software). One reason for this was, that the best camera perspectives provided by head and breast mounted action cameras, were not provided to the expert in this experimental setting. In an additional video based evaluation, the different camera perspectives were compared and a better solution was found. But the expert in the live scenario has also complained, that he does not fully understand the plant context and the production process. We therefore suggest a brief teleanalysis phase with tools explained in Chapter 5 before a telemaintenance operation is started.

The last section tried to improve the situation awareness gained out of a static video perspective, which is usually provided in a teleanalysis scenario. As a comparison, we used the virtual model of the plant (described in Chapter 5) as each static and dynamic ("computer game") perspectives. An additional possibilty was the newly developed "ARTab" application. This is based on a tablet PC used in a room with a tracking system. A virtual model of the plant is rendered on the camera video of the tablet PC, as if the plant was standing in that room. The user was able to walk around the plant. The augmented virtual model proofed significantly better than the mere virtual perspective. Also there has been strong evidence, that a moving perspective served better than a fixed perspective and that a process highlighting led to better results than without highlighting. Overall, this provides interesting starting points for future research.

Chapter 7

Conclusion

7.1 Summary

The research project "MainTelRob – Maintenance and Telematics for Robots" was funded by the Bavarian Economy Ministry and addressed challenges of telemaintenance. The project ran since 2012 for three years with the industrial partners KUKA Industries and P&G. It has been followed by the project "Bayern.digital", which approaches subsequent questions.

Both projects consider maintenance under a telematics view. The core issues of "Industry 4.0" are internet technologies, intelligent objects and an adaptive human machine interaction. They are realized in a specific setting for the maintenance of industrial manipulators. In Germany the term maintenance covers the basic measures inspection ("Inspektion"), maintenance ("Wartung"), restoration or repair ("Instandsetzung"), improvement ("Verbesserung") and weak spot analysis ("Schwachstellenanalyse") (DIN 31051:2003-06).

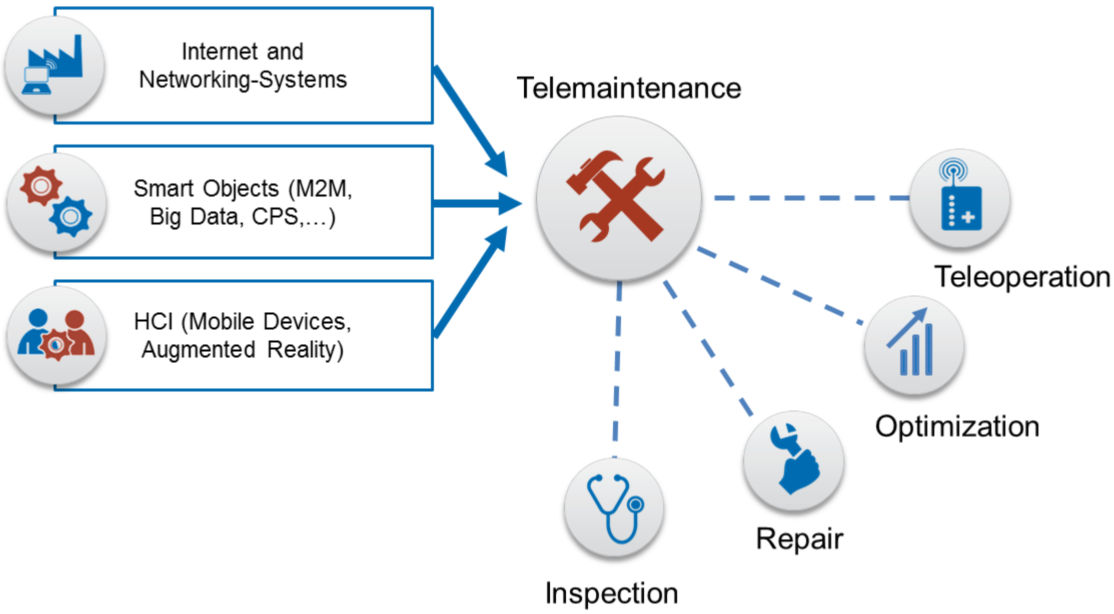


Figure 7.1: *Industry 4.0 and telemaintenace*

The best constellation for maintenance is determined with a cost-efficiency analysis. Breakdowns are extremely costly, as every minute a plant is not producing is very expensive. This applies especially, if the plant is on a critical path of the production. In case of such an unexpected fault, already the diagnosis can take a long time. Thus, they should not happen at all in the best case. The same is true for immediate corrective maintenance measures, because they need to be performed in the case of a breakdown. Otherwise the action can be deferred to a better instant of time (deferred corrective maintenance). Corrective maintenance measures tend to be more costly than preventive maintenance, as something is actually broken and needs to be mended. Hence, the less corrective maintenance actions are undertaken the better. If a lot of preventive maintenance actions are performed, breakdowns can be prevented. But these are also expensive. Hence it is optimal, to drive the plant as near to the threshold level of damage as possible [28]. This section can be summarized as: The goal is to perform as much maintenance as necessary and as little as possible.

In order to detect abrasion continuously and to apply an adequate amount of preventive maintenance, we need to fusion sensor data of different plant parts. Repair actions need to be undertaken with meticulous precision and in an exact amount. If possible, each overhaul process is combined with optimization measures to increase the working supply of the plant. In most cases, external expert knowledge is required for this. Those external experts need to be placed in the position to react in real-time critical situations. Thus, telemaintenace requires the combination of intelligent inspection, repair, optimization and teleoperation. The actual state of telemaintenance (mainly phone support) was improved substantially with telematics methods in the project MainTelRob.

Figure 7.2 visualizes the project context and the different innovations which are described in this publication: On the left side is the customer P&G, depicted in red. The plant consists of an industrial robot, a molding machine and a montage system. The service technicians perform manual inspections with a mobile device or execute repair operations guided by Augmented Reality (see Chapter 4). Data collected by the human, the plant's sensor and video data can be recorded. Sensor data can be checked in real-time by the Condition Monitoring system. The recorded data serves as a base for remote teleanalysis, fault diagnosis and optimization tasks, see Chapter 5.

If external support is needed for maintenance or repair activities, the robot manufacturer (KUKA Industries) depicted blue on the right sight can intervene. The data transfer via the internet is realized in a special networking innovation, the Adaptive Management and Security System (AMS), which takes care of the available bandwidth of the network. The external expert can access the plant remotely for diagnosis tasks, remote control the robot or perform optimization tasks. He can also communicate with the service technician, who uses her mobile device with a video-interaction (see Chapter 6). Both humans are able to solve the problem together in computer supported collaborative work.

With the developed techniques, even complicated repair tasks, which have re-

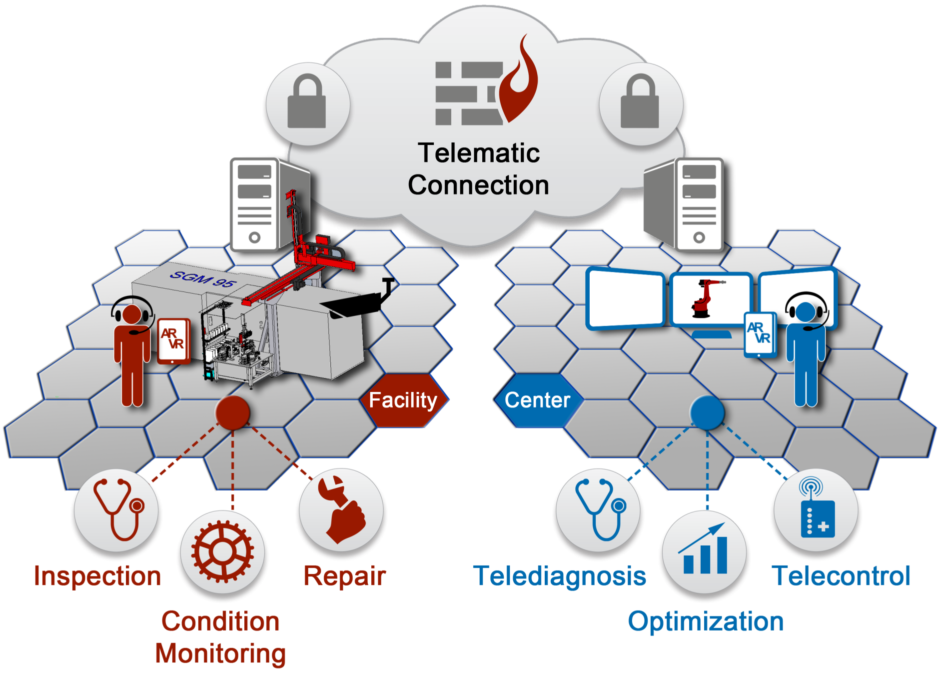


Figure 7.2: *Complete telemaintenance system*

quired the expert to travel to the production site, can now be supervised remotely. In a proof of concept experiment, even the exchange of the motor of an axis of the manipulator was performed with telemaintenance. The key criterion for the external expert is situation awareness. This thesis showed that the situation awareness can be enhanced substantially by using different camera viewpoints. We propose using the teleanalysis phase with Augmented Reality before starting a telemaintenance communication with the service technician.

7.2 Innovation overview

The main research focus of this work lies on the human machine interface for all human tasks in a telemaintenance setup. This thesis provides own work in the use of a mobile device in context of maintenance, describes new tools on asynchronous remote analysis and puts all parts together in an integrated telemaintenance infrastructure. With the help of Augmented Reality, the user performance and satisfaction could be raised. A special regard is put upon the situation awareness of the remote expert realized by different camera viewpoints. In detail the work consists of:

- Support of maintenance tasks with a mobile device (Section 4.2)
- Development and evaluation of a context-aware inspection tool (Section 4.3)

- Comparison of a new touch-based mobile robot programming device to the former teach pendant (Section 4.4)
- Study on Augmented Reality support for repair tasks with a mobile device (Section 4.5)
- Condition monitoring for a specific plant with industrial robot (Section 5.2)
- Human computer interaction for remote analysis of a single plant cycle (Section 5.3)
- A big data analysis tool for a multitude of cycles and similar plants (Section 5.4)
- 3D process visualization for a specific plant cycle with additional virtual information (Section 5.5)
- Establishing a telematics architecture in hardware, software and network infrastructure (Section 6.2)
- Mobile device computer supported collaborative work for telemaintenance (Section 6.3)
- Motor exchange telemaintenance example in running production environment (Section 6.4)
- Augmented reality supported remote plant visualization for better situation awareness (Section 6.5)

Figure 7.3 summarizes the innovations developed in project "MainTelRob" and described in this thesis. In the upper part of the figure, the telemaintenance scenario is visualized, which consists of the live support for the service technician and the telemaintenance environment for the external expert (Chapter 6). Furthermore, the asynchronous teleanalysis (Chapter 5) is visualized and also the use of mobile devices in the production (Chapter 4) can be seen.

The project team out of KUKA Industries, Procter&Gamble and the ZfT was honored with the price "Industriepreis Best of 2015" for this combination of innovations.

7.3 Economical benefits

Next to the scientific evaluation of the developed innovations, a economical analysis has been carried out by P&G throughout the project "MainTelRob".

With the support of a mobile device for inspection tasks, the current working supply of the plant can be estimated more accurately. Further, the realization of individual maintenance plans is facilitated with this flexible software. The users see a huge potential for increasing quality of the documentation with the help of such

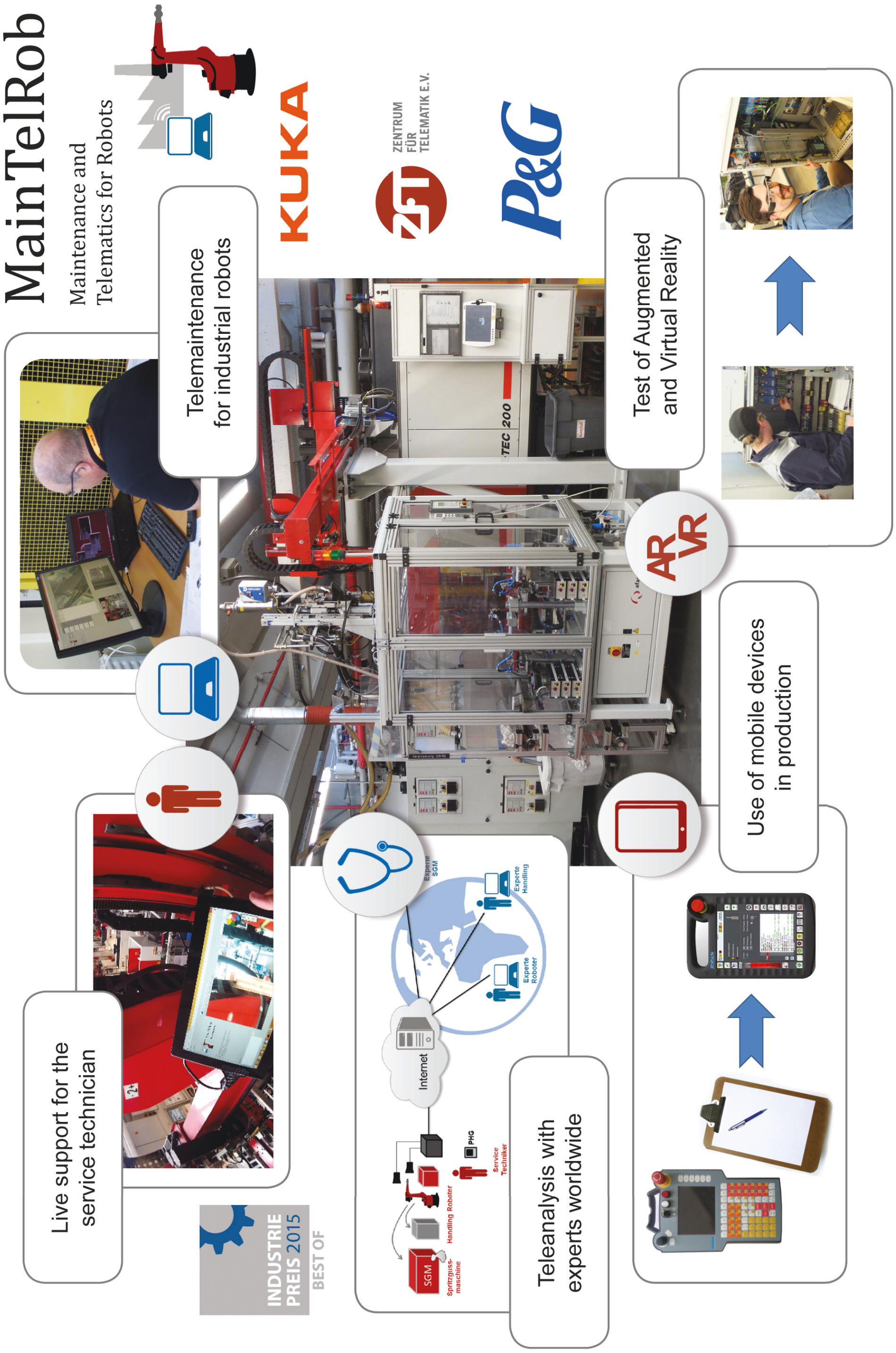


Figure 7.3: *Innovations realized in MainTelRob*

mobile devices, while they hope to decrease the effort for documentation simultaneously. Supervising the relevant parameters with the Condition Monitoring system is also very important for inspection. Asynchronous support through external experts is seen very positively, because the discussion can lead to an optimization of the plant, the production process, and to a higher productivity. The concrete approaches for a reduced cycle duration found in the user study lead to a lower part price, because the plant works 24 hours a day and even small savings in the cycle duration have huge effects. Also the telemaintenance has been evaluated very positively. The direct maintenance costs of external mechanics can be reduced with the software.

In Figure 7.4 the maintenance costs of P&G in Marktheidenfeld for 2015/16 of 2.1 million Euro is shown with its components and the possible savings due to "MainTelRob". The economical analysis resulted in an estimated saving of 15% per year.

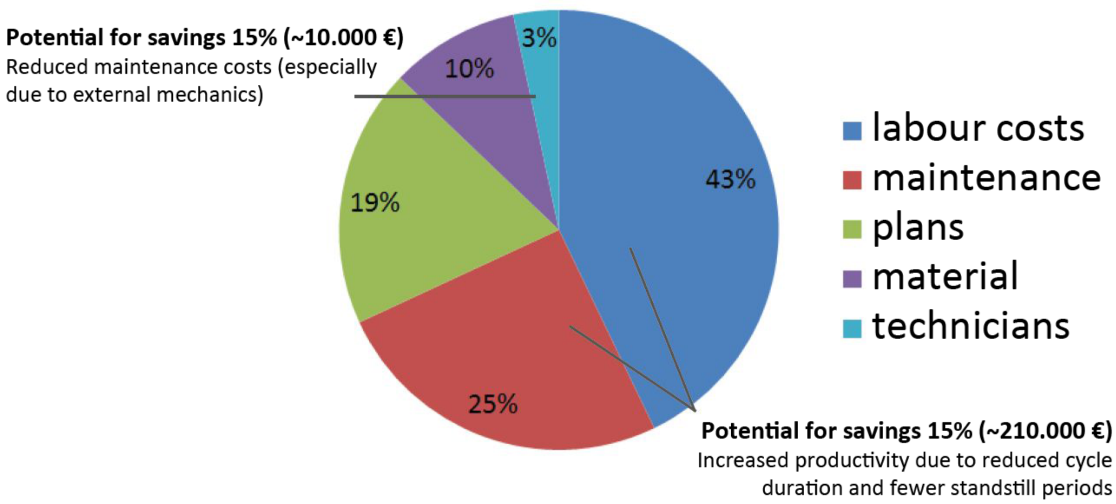


Figure 7.4: *Potential for savings in MainTelRob*

7.4 Suggestions for future work and further applications

The presented work can be applied to any telemaintenance scenario and proves useful for considerations of human factors in industrial projects. As there is an ongoing discussion on mobile devices and Augmented Reality in industrial environments, this work can also provide helpful insights. The results for situation awareness are also useful for teleoperation tasks in general.

Nevertheless, many interesting research questions remain. A further examination on the human-to-human communication with the help of video interaction in repair and maintenance scenario might be interesting. Another insight would be provided by comparing the results with Spatial Augmented Reality solutions or more advanced Virtual Reality methods, i.e. Video-See-Through Glasses or a VR Cave. Of course, the verification of the results of the user studies in practice with domain experts is planned for future projects.

List of Own Publications

[1] D. Aschenbrenner, M. Latoschik, and K. Schilling. **Industrial maintenance with augmented reality, two case studies**. In *22nd ACM Symposium on Virtual Reality Software and Technology*, 2016.

[2] Robert Tscharn, Philipp Schaper Schaper, Doris Aschenbrenner, Diana Löffler, and Jörn Hurtienne. **Benutzerschnittstellen für industrieroboter - eine usability-evaluation**. In *Mensch und Computer*, 2016.

[3] Doris Aschenbrenner, Nicolas Maltry, Johannes Kimmel, and Klaus Schilling. **Artab - using virtual and augmented reality methods for an improved situation awareness for telemaintenance**. In *4th IFAC Symposium on Telematics Applications*, 2016.

[4] Michael Fritscher, Felix Sittner, Doris Aschenbrenner, Markus Krauss, and Klaus Schilling. **The adaptive management and security system for maintenance and teleoperation of industrial robots**. In *4th IFAC Symposium on Telematics Applications*, 2016.

[5] Michael Fritscher, Felix Sittner, Doris Aschenbrenner, Markus Krauss, and Klaus Schilling. **Introducing the facility asynchronous data analysis tool (fadat) to optimize productive machine cycle in industrial plants**. In *4th IFAC Symposium on Telematics Applications*, 2016.

[6] Doris Aschenbrenner, Felix Sittner, Michael Fritscher, Markus Krauss, and Klaus Schilling. **Cooperative remote repair task in an active production line for industrial internet telemaintenance**. In *4th IFAC Symposium on Telematics Applications*, 2016.

[7] Michael Albert, Doris Aschenbrenner, Paul Barth, Michael Fritscher, and Klaus Schilling. **Extension of a telemedicine framework for analysis of industrial machinery data**. In *4th IFAC Symposium on Telematics Applications*, 2016.

[8] Piotr Rygielski, Viliam Simko, Felix Sittner, Doris Aschenbrenner, Samuel Kounev, and Klaus Schilling. **Automated extraction of network traffic models suitable for performance simulation**. In *Proceedings of the 7th ACM/SPEC on International Conference on Performance Engineering*, pages 27–35. ACM, 2016.

[9] Doris Aschenbrenner. **Fernwartung von industrierobotern - intuitive visualisierung des anlagenkontextes mit virtual und augmentedreality**. In *17. Bayreuther 3D-Konstrukteurstag*, 2015.

[10] Doris Aschenbrenner, Michael Fritscher, Felix Sittner, and Klaus Schilling. **Design process for user interaction with robotic manipulators in industrial internet applications**. In *Proceedings of the 33rd Annual International Conference on the Design of Communication*, page 18. ACM, 2015.

[11] Doris Aschenbrenner, Felix Sittner, Michael Fritscher, Markus Krauss, and Klaus Schilling. **Teleoperation of an industrial robot in an active production line**. In *Submitted Paper for: 2nd IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control (CESCIT)*. IFAC, June 2015.

[12] Ali Kheirkhah, Doris Aschenbrenner, Michael Fritscher, Felix Sittner, and Klaus Schilling. **Networked control system with application in the industrial tele-robotics**. In *2nd IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control (CESCIT)*, 2015.

[13] Doris Aschenbrenner, Florian Leutert, Felix Sittner, Michael Fritscher, and Klaus Schilling. **Einsatzmöglichkeiten mobile geräte für wartung von industrierobotern**. In *Proceedings of SPS-IPC-Drives*, 2014.

[14] Felix Sittner, Markus Krauss, Michael Fritscher, Doris Aschenbrenner, and Klaus Schilling. **Entwicklung eines adaptiven bandbreitenmanagement- und sicherheitssystems für die industrielle fernwartung**. In *Proceedings of SPS-IPC-Drives*, 2014.

[15] Felix Sittner, Doris Aschenbrenner, Michael Fritscher, Ali Kheirkhah, Markus Krauss, and Klaus Schilling. **Maintenance and telematics for robots (maintelrob)**. In *3rd IFAC Symposium on Telematics Applications Conference (TA2013), Seoul*. IFAC, 2013.

[16] Oliver Kurz, Manuel Stahl, Doris Aschenbrenner, Michael Fritscher, Robin Hess, and Klaus Schilling. **Safe navigation for autonomous robot systems–sinafar**. *Coordinates*, 2013.

List of Figures

1.1 Schema for industry 4.0 2

2.1 Communication in a general industrial internet setting 8
2.2 Different maintenance types (SS-EN 13306:2001 based on IEC 60050-191 [27] as referenced in [28]) 9
2.3 Visualization of an asset history and its working supply (with changes out of [30]) 10
2.4 Different tasks for telemaintenance 12
2.5 Concept of M2M communication (following [25]) 13
2.6 Simplified scheme of process automation (following [34]) 14
2.7 Automation pyramide 15
2.8 Industrial manipulators 15
2.9 Visualization of quantization 16
2.10 Schema for information transmission [39] 17
2.11 Schematic visualization of packet switching: A message is sent from Europe to China in packets, which will arrive on different transmission paths, in a different order, in different time or not at all (visualized in packet 4) 18
2.12 Development of the events "failure" or "malfunction" from a fault which causes a stepwise or driftwise change of a feature (out of [34]) . 19
2.13 Scheme of a process or product P influenced by faults F: a) process in open loop; b) process in closed loop (out of [34]) 20
2.14 Basic elements of diagnostics and prognostics for Condition Based Maintenance 21
2.15 Knowledge discovery schema from [48] 22
2.16 Statistical regression for prediction of machine state 23
2.17 Rule-based fault detection 23
2.18 Diagnosis with reference cases 24
2.19 Use of decision trees 24
2.20 Integration of different methods 24
2.21 Socio-technical system 25
2.22 Human error sources 26
2.23 Mental workload 27
2.24 Visualization of the concept of mental models and the semiotic triangle of symbol, mental model (thought or reference) and real object (referent) 28
2.25 Visualization of conceptual models 29
2.26 Visualization of usability measuring 29

2.27 The user experience development cycle (following [54]) 31
2.28 The user experience development process (according to [54]) 32
2.29 Time-space matrix for CSCW (following [76]) 33
2.30 Model for H2H communication based on [78] 35
2.31 Visualization of common ground 35
2.32 Human perception adapted from [58] 37
2.33 Regulation layers for human behaviour from [58] 37
2.34 Situation Awareness 39
2.35 Spatial coordinate system 40
2.36 Frames of reference 41
2.37 Possible viewpoints and interfaces adapted from [92] 42
2.38 Cost and Benefits of Different Display Perspectives adapted from [93] 42
2.39 Information spaces adapted from [98] 43
2.40 x-y visualization of data in (a) scatterplot, (b) line graph (c) step chart 44
2.41 x-y data with trend curve (a) scatterplot with fitted LOESS [100] curve, (b) line graph linear trend line, (c) step chart with second grade polynomial regression trend 45
2.42 Circle displays (a) pie chart, (b) "donut" chart, (c) pie chart with too many values 45
2.43 Bar graphs in comparison: (a) bar graph, (b) divided bar graph, (c) same values with line graphs 46
2.44 Reality-Virtuality-Continuum following [114] 49
2.45 General architecture for Augmented Reality applications 50
2.46 Visualization of camera calibration 51
2.47 Pose estimation problem 52
2.48 Sharing and Trading Control after [64] 54
2.49 General scheme of advanced supervision methods with fault management (supervisory loop) from [34] 56
2.50 A telemaintenance system and its block scheme reprentation. Subscripts m and s refer to variables at the master and slave site following [123] 56
2.51 Human Supervisory Control model as introduced in [64] 58
2.52 Easy approach for Human Supervisory Control as a control model according to [64] 59
2.53 Mental models in Human Supervisory Control according to [64] . . . 59
2.54 Combination of Human Supervisory Control: a) remote HSC b) local HSC c) extended HSC 60
2.55 Extended Human Supervisory Control required for Telemaintenance . 60
2.56 Communication processes in extended Human Supervisory Control . . 61

3.1 Word cloud "Project goals" created with http://worditout.com/ . . . 64
3.2 Schema of telemaintenance 65
3.3 Photo of the production plant used for MainTelRob 66
3.4 Lower part of electronic toothbrushes 67
3.5 Project survey 67

3.6 Photos of plant parts 73
3.7 Part production process 74
3.8 Symbols for different major parts of the project 75
3.9 Current flow model 76
3.10 Envisioned flow model 77
3.11 Visualization of the local and remote scenarios 78
3.12 Schematic visualization of scenario 1 of MainTelRob 78
3.13 Schematic visualization of scenario 2 of MainTelRob 79
3.14 Schematic visualization of scenario 3 of MainTelRob 79
3.15 Schematic visualization of scenario 4 of MainTelRob 80
3.16 Schematic visualization of scenario 5 of MainTelRob 80
3.17 Schematic visualization of scenario 6 of MainTelRob 81
3.18 Extended Human Supervisory Control and MainTelRob scenarios . . 82

4.1 Context of this chapter visualized in Extended Human Supervisory Control Model 86
4.2 Teach pendant (left) of a a Reis/KUKA Industries manipulator and a new reisPAD (right) 87
4.3 All in one strategy for the mobile device 88
4.4 Maintenance of industrial robot (visualization part 1) 90
4.5 Maintenance of industrial robot (visualization part 2) 91
4.6 For scenario 1 implemented checklist with checkboxes (representaion on the reisPAD of ROBOTstarVI) 93
4.7 Flow model (left) and stationary BDE Terminal (right) for Scenario 2 95
4.8 Physical model for a part of the CIL (cleaning, inspection, lubrication) tasks for Scenario 2 96
4.9 Scenario 2 paper prototype first version 97
4.10 Scenario 2 paper prototype second version 97
4.11 Scenario 2 first raw version first version 98
4.12 First prototype for manual cyclic control 98
4.13 Test person using tablet PC for CIL for the first time 99
4.14 Final prototype for longtime evaluation 99
4.15 Human Centered Design: Gathering user feedback 100
4.16 Excerpts of the pluralistic walkthrough evaluation 101
4.17 QUESI values for scenario 2 102
4.18 NASA TLX values for scenario 2 102
4.19 Production worker will use mobile communication further in working context 103
4.20 The use of mobile devices increases the quality of documentation . . . 103
4.21 The use of mobile devices reduces the effort for documentation . . . 104
4.22 The switch cabinet with integrated servo amplifies (blue) 113
4.23 Showing a failure in axis 3 controller on reisPAD 114
4.24 Documentation for error code 615 115
4.25 Discussion on the functionalities of scenario 3 115
4.26 Layout concept for the Eyestruction application 117

4.27 The used Samsung Galaxy Tab 2 118
4.28 Device specific data of Samsung Galaxy Tab 2 118
4.29 The Optical See-Through Head-Mounted Display Epson Moverio BT-200 119
4.30 Device specific data for Epson Moverio BT-200 119
4.31 Survey of the used components for the Eyestruction implementation . 120
4.32 Example for maintenance data 121
4.33 Used 3D-models: rotating marker, static marker and arrow 122
4.34 Simplified class diagram of used classes 122
4.35 Layout of MainActivity GUI 123
4.36 Activity diagram for initialization process of Eyestruction 123
4.37 Left: Controller box, Middle: Controller cover, Right: Controller with adjusted cover 124
4.38 Survey of the markers in the switch cabinet 125
4.39 GUI TabletTextActivity 126
4.40 GUI TabletTextActivity with explanation 127
4.41 Function marker for the interaction with the HMD application 128
4.42 HMD GUI with text 128
4.43 HMD GUI with audio 129
4.44 Contrasts used for evaluation 132
4.45 Average amount of errors for each experimental condition 133
4.46 Average required time in minutes for each experimental condition . . 133
4.47 Average NASA TLX value for each experimental condition 134
4.48 Average QUESI-values for each experiment condition 135
4.49 Average comment-value for each experiment condition 135
4.50 Table with all results from AR repair study 136
4.51 Summary of the innovations: Mobile device for local maintenance . . 138

5.1 Context of this chapter visualized in the extended Human Supervisory Control model 142
5.2 Understanding problems in teleanalysis 143
5.3 Relation between Scenario 4 and 5 144
5.4 Relation between Scenario 4 and 5 147
5.5 Screenshot of teach pendant with warning message 148
5.6 Screenshot ProVis with depicted sensor data 149
5.7 Envisioned flow model for optimizations 150
5.8 Paper prototype for optimization 151
5.9 Prototype for optimization 152
5.10 Final prototype for optimization 153
5.11 Average TLX values per task (time optimization, abrasion optimization, second manipulator) 155
5.12 Average TLX values per task 155
5.13 Seperate TLX separate values 156
5.14 Average QUESI values 156
5.15 Average ISONORM values 157

5.16 Concept idea of the Complex Analysis Toolkit (CAT) 159
5.17 Module Least Square Fit 161
5.18 Component M-Data 162
5.19 Rule block V-Data 163
5.20 Position of axis 2 in all test cycles 164
5.21 Result of Least Square Fit on cycle 35 165
5.22 Position of axis 2 after LSF on cycle 35 166
5.23 Time shift of all float point variables of all test cycles with respect to cycle 35 167
5.24 Time shift of all float point variables of cycles 20 to 52 with respect to cycle 35 168
5.25 Scenario 1: Rule editor (left) and threshold comparison (right) 168
5.26 Scenario 2: Verification of a cycle 169
5.27 Module Threshold 169
5.28 Scenario 3: A cycle in reference to many cycles 171
5.29 Module Quantile Function 172
5.30 Scenario 4: Visual Control Histogram 172
5.31 Module "Plot" with selection Histogram 173
5.32 Scenario 6: Complex cycle comparison 174
5.33 Architecture software extensions 175
5.34 Interaction of the rule solver with CAT software 176
5.35 CAT interface in AMS web environment 178
5.36 Wireframes for CAT perspective 178
5.37 Left click interaction with an interaction point in Unity 179
5.38 Interaction model 180
5.39 Interactive Unity environment 180
5.40 Visualization of the UnityConfig_MTR.dtd 182
5.41 I^2T load (axis 3) of cycle 35 183
5.42 Temperature of the motor in cycle 35 184
5.43 Example rule for testing on current peaks displayed in the rule editor 184
5.44 Current trend of motor of axis 3 185
5.45 Motor current of axis 3 with threshold 6 Ampere 185
5.46 Separating warm-up and regular operation in rule editor 186
5.47 Sum of quadratic error for cycle 35 axis 2 position 186
5.48 Validation of acceleration (Axis 4) of cycle 40 in rule editor 187
5.49 Acceleration of axis 4 (50% Quantile) 188
5.50 Acceleration of axis 4 (50% Quantile, zoomed) 188
5.51 Acceleration of axis 4 (100% Quantile) 189
5.52 Temperature area for controller power stage 189
5.53 Rule example for histograms of all VCP data in rule editor 190
5.54 VCP Histogram 191
5.55 First approach for VCP visualization 191
5.56 Correlation between position, velocity, acceleration and motor current of axis 2 192

5.57 Example rule for correlation of velocity, acceleration and motor current in the rule editor . . . 193
5.58 Correlation of velocity derivation, acceleration and current of axis 2 . 193
5.59 From up to down: linear interpolation of the position of axis 2, first derivation of position with respect to time, second derivation of position with respect to time . . . 195
5.60 From up to down: linear interpolation of the position of axis 2, first derivation of position with respect to time, second derivation of position with respect to time . . . 195
5.61 Example of a rule for histogram out of several Binary Streams in rule editor . . . 196
5.62 Validation of cycle 40 with four heuristics . . . 196
5.63 Production process visualization for ProcVis . . . 199
5.64 Production process visualization for ProcVis . . . 200
5.65 UML chart for ProcVis classes in Unity . . . 200
5.66 Unity GUI element for CSVMachineController script . . . 201
5.67 Unity GUI element for PartController script . . . 201
5.68 Unity GUI element for InstantiationTimer script . . . 202
5.69 Begin of process (red part spawns) . . . 203
5.70 Second working step (red part transforms to green part) . . . 203
5.71 All three working status parts visible . . . 204
5.72 Summary of the innovations: Asynchronous remote teleanalysis . . . 205

6.1 Context of this chapter visualized in Extended Human Supervisory Control Model . . . 210
6.2 Flow Model for telemaintenance . . . 213
6.3 Schematic overview to the telemaintenance architecture . . . 214
6.4 Model of the VPN Tunnel . . . 215
6.5 Camera positions . . . 216
6.6 Photos of the cameras (left) and camera perspective of camera 1 . . . 216
6.7 Indicator plates and bulletins for camera surveillance . . . 216
6.8 1. Testbed: Control test environment . . . 217
6.9 2. Testbed: Emulator testbed for virtual manipulator control . . . 217
6.10 3. Testbed: Local tests with KUKA Industries control system . . . 218
6.11 Prototype 1: Testbed structure with network emulation (photo) . . . 219
6.12 Prototype 1: Testbed structure with network emulation . . . 219
6.13 Prototype 2: Teleoperation testbed with different data streams . . . 220
6.14 Test person with mobile device following the expert's instructions in front of the switch cabinet . . . 221
6.15 Abstract network structure with ping delay . . . 223
6.16 UDP and TCP delay (best case) . . . 224
6.17 Delay of TCP and UDP test streams (worst case) . . . 224
6.18 UDP delay (worst case) with losses . . . 225
6.19 Time domains for control . . . 226
6.20 Key timing goals for the prototype . . . 227

6.21 Requirements of the AMS Services in the Security Tunnel 227
6.22 Basic concept of the Adaptive Management and Security System (AMS) 228
6.23 Layers of the AMS architecture on top of OSI layer four 228
6.24 Basic AMS applications and services 229
6.25 Service Interactions 231
6.26 Composition of data streams in the encrypted tunnel 232
6.27 Schematic overview of the Telematics Connections Layer 233
6.28 Task interaction diagram out of the perspective of the service technician (left) 234
6.29 Task interaction diagram out of the perspective of the external expert (right) 235
6.30 Component model for remote expert 237
6.31 Prototype for remote expert 237
6.32 Component model for service technician 238
6.33 Prototype for service technician 239
6.34 Concept of video interaction 239
6.35 Video interaction realization in the Android application 240
6.36 Video interaction realization in the Processing application 243
6.37 Setup of Tracker evaluation a) LEGO environment and b) Switch cabinet 245
6.38 Results of the benchmark for the BoofCV trackers for both environments 246
6.39 Visualization of the concept 246
6.40 Latency of computer vision algorithms a) in the android application and b) at the Processing application on the PC 248
6.41 Realization on the tablet PC with LEGO examples 249
6.42 AR screen resolution problem 250
6.43 Status bar on the AR application 251
6.44 Video interaction on the Switch cabinet with HMD device 251
6.45 Service technician with tablet PC (left) and screenshot of tablet GUI with Painting App (right) 252
6.46 Test setting for motor exchange (July 28th, 2015) 253
6.47 Experimental setting and camera locations 254
6.48 Tablet PC screenshot of the video interaction 256
6.49 Expert with teach pendant at a remote working desk 256
6.50 Service technician view (head-mounted camera) on teach pendant . . 257
6.51 ISONORM values for expert (blue) and service technician (red) . . . 258
6.52 QUESI values for expert (blue) and service technician (red) 258
6.53 SART values for expert (blue) and service technician (red) 259
6.54 NASA TLX values for expert (blue) and service technician (red) . . . 259
6.55 Video-based subsequent evaluation (mean SART values) 260
6.56 The head-mounted action camera provided the best view on the difficult repair steps, but was very instable due to head movements . . . 262
6.57 ARTab application concept 263
6.58 Use of different coordinate systems 264

6.59 Hardware and network configuration 265
6.60 Software architecture 265
6.61 Frames used for ARTab 266
6.62 Visualization of camera calibration for ARTab 267
6.63 Measuring checkboard frame 268
6.64 Table in the real world (left) with virtual box (right) 269
6.65 Matrix adjustment tool on the right of the AR picture with virtual blue box on table 269
6.66 Matrix offset (green: z, blue: y, red: x) 270
6.67 Six experiment conditions for ARTab user study 272
6.68 Participant in exploration phase (freeplay) for ARTab user study . . . 273
6.69 Picture of the plant provided for the "Axis" task for ARTab user study 275
6.70 Participant at the "LEGO" task for ARTab user study 276
6.71 Sample solution for task "Process" for ARTab user study 277
6.72 Results of the SART questionnaire 279
6.73 Results of the TLX questionnaires for each task 280
6.74 Results of the QUESI questionnaire 280
6.75 Results of the ISONORM questionnaire 281
6.76 Results of the axis identification task 281
6.77 Results of the lego task 282
6.78 Results of the process identification task 282
6.79 Summary of the results of the AR Tab user study: red and green marks the experimental conditions compared, yellow marked entries show significant values ($p < 0.05$), light green marks ($p = 0.05$), blue marked entries show significant values, where the Box M-Test also has been significant 288
6.80 Summary of the innovations: Telemaintenance as Human Supervisory Control 290

7.1 Industry 4.0 and telemaintenace 295
7.2 Complete telemaintenance system 297
7.3 Innovations realized in MainTelRob 299
7.4 Potential for savings in MainTelRob 300

List of Tables

4.1 Results from t-testing 108

6.1 Motor exchange: Chronological working steps and identified critical decisions. 255
6.2 Mean questionnaire values 260
6.3 Critical decision method (excerpt) 261

References

[1] Bundesministerium für Bildung und Forschung. **Zukunftsbild 'industrie 4.0'**.

[2] Feng Xia, Laurence T Yang, Lizhe Wang, and Alexey Vinel. **Internet of things**. *International Journal of Communication Systems*, 25(9):1101, 2012.

[3] Jon Bruner. *Industrial Internet.* " O'Reilly Media, Inc.", 2013.

[4] Paulo Tabuada. **Cyber-physical systems: Position paper**. In *NSF Workshop on Cyber-Physical Systems*, 2006.

[5] August-Wilhelm Scheer. **Industrie 4.0: Wie sehen produktionsprozesse im jahr 2020 aus**. *IMC AG*, 2013.

[6] Detlef Zühlke. **Die cloud ist voraussetzung für industrie 4.0**, June 2013.

[7] VDI/VDE Gesellschaft Mess und Automatisierungstechik. **Automation 2020 bedeutung und entwicklung der automation bis zum jahr 2020**, Januar 2013.

[8] Dieter Spath, Oliver Ganschar, Stefan Gerlach, Moritz Hämmerle, Tobias Krause, and Sebastian Schlund. *Produktionsarbeit der Zukunft-Industrie 4.0.* Fraunhofer Verlag Stuttgart, 2013.

[9] Felix Sittner, Doris Aschenbrenner, Michael Fritscher, Ali Kheirkhah, Markus Krauss, and Klaus Schilling. **Maintenance and telematics for robots (maintelrob)**. In *3rd IFAC Symposium on Telematics Applications Conference (TA2013), Seoul.* IFAC, 2013.

[10] Doris Aschenbrenner, Michael Fritscher, Felix Sittner, and Klaus Schilling. **Design process for user interaction with robotic manipulators in industrial internet applications**. In *Proceedings of the 33rd Annual International Conference on the Design of Communication*, page 18. ACM, 2015.

[11] Doris Aschenbrenner, Florian Leutert, Felix Sittner, and Klaus Schilling. **Einsatzmöglichkeiten für mobile geräte bei der wartung von industrierobotern**. In *SPS Drives Kongress*, 2014.

[12] Robert Tscharn, Philipp Schaper Schaper, Doris Aschenbrenner, Diana Löffler, and Jörn Hurtienne. **Benutzerschnittstellen für industrieroboter - eine usability-evaluation**. In *Mensch und Computer*, 2016.

[13] D. Aschenbrenner, M. Latoschik, and K. Schilling. **Industrial maintenance with augmented reality, two case studies**. In *22nd ACM Symposium on Virtual Reality Software and Technology*, 2016.

[14] Doris Aschenbrenner, Felix Sittner, Michael Fritscher, Markus Krauss, and Klaus Schilling. **Teleoperation of an industrial robot in an active production line**. In *Submitted Paper for: 2nd IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control (CESCIT)*. IFAC, June 2015.

[15] Felix Sittner, Markus Krauss, Michael Fritscher, Doris Aschenbrenner, and Klaus Schilling. **Entwicklung eines adaptiven bandbreitenmanagement- und sicherheitssystems für die industrielle fernwartung**. In *Proceedings of SPS-IPC-Drives*, 2014.

[16] Michael Fritscher, Felix Sittner, Doris Aschenbrenner, Markus Krauss, and Klaus Schilling. **The adaptive management and security system for maintenance and teleoperation of industrial robots**. In *4th IFAC Symposium on Telematics Applications*, 2016.

[17] Piotr Rygielski, Viliam Simko, Felix Sittner, Doris Aschenbrenner, Samuel Kounev, and Klaus Schilling. **Automated extraction of network traffic models suitable for performance simulation**. In *Proceedings of the 7th ACM/SPEC on International Conference on Performance Engineering*, pages 27–35. ACM, 2016.

[18] Ali Kheirkhah, Doris Aschenbrenner, Michael Fritscher, Felix Sittner, and Klaus Schilling. **Networked control system with application in the industrial tele-robotics**. In *2nd IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control (CESCIT)*, 2015.

[19] Michael Fritscher, Felix Sittner, Doris Aschenbrenner, Markus Krauss, and Klaus Schilling. **Introducing the facility asynchronous data analysis tool (fadat) to optimize productive machine cycle in industrial plants**. In *4th IFAC Symposium on Telematics Applications*, 2016.

[20] Michael Albert, Doris Aschenbrenner, Paul Barth, Michael Fritscher, and Klaus Schilling. **Extension of a telemedicine framework for analysis of industrial machinery data**. In *4th IFAC Symposium on Telematics Applications*, 2016.

[21] Doris Aschenbrenner, Felix Sittner, Michael Fritscher, Markus Krauss, and Klaus Schilling. **Cooperative remote repair task in an active production line for industrial internet telemaintenance**. In *4th IFAC Symposium on Telematics Applications*, 2016.

[22] Doris Aschenbrenner, Nicolas Maltry, Johannes Kimmel, and Klaus Schilling. **Artab - using virtual and augmented reality methods for an improved situation awareness for telemaintenance**. In *4th IFAC Symposium on Telematics Applications*, 2016.

[23] Robert Shea and Robert Anton Wilson. *The Illuminatus! Trilogy*. Hachette UK, 2012.

[24] Simon Nora and Alain Minc. *L'informatisation de la société: annexes*, volume 1. La documentation française, 1978.

[25] David Boswarthick, Omar Elloumi, and Olivier Hersent. *M2m communications: a systems approach*. John Wiley & Sons, 2012.

[26] IInternational Electrotechnical Commission (IEC). **Elektropedia: The world's online electrotechnical vocabulary**, 2016. URL: `http://www.electropedia.org/`.

[27] International Electrotechnical Commission et al. **Iec 60050 (191) international electrotechnical vocabulary, chapter 191: Dependability and quality of service**, 1990.

[28] Marcus Bengtsson. *Condition based maintenance systems: an investigation of technical constituents and organizational aspects*. Citeseer, 2004.

[29] Deutsche Industrienorm. **Din 31051:2003-06. grundlagen der instandhaltung.**, 2003.

[30] Thorsten Förstemann and Christoph Hartung. *Analyse von Servicedaten medizin-technischer Geräte in Krankenhäusern*. PhD thesis, 2005. URL: `http://www.foerstemann.name/misc/doktorarbeit/doktorarbeit.html`.

[31] Soumitra Chowdhury and Asif Akram. **E-maintenance: Opportunities and challenges**. In *The 34th Information Systems Research Seminar in Scandinavia (IRIS), Turku, Finland, August 16-19, 2011*, pages 68–81. Turku Centre for Computer Science, 2011.

[32] DsiN.de. **It-sicherheitslage im mittelstand 2011. eine studie von deutschland sicher im netz, berlin**, 2011.

[33] Abdessamad Mouzoune and Saoudi Taibi. **Introducing e-maintenance 2.0**. *arXiv preprint arXiv:1401.8252*, 2014.

[34] Rolf Isermann. *Fault-diagnosis applications: model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems*. Springer Science & Business Media, 2011.

[35] International Electrotechnical Commission et al. **Iec 62264 enterprise-control system integration**, 2013.

[36] International Organization for Standardization. **Iso 8373:2012(en) robots and robotic devices - vocabulary**, 2012. URL: `https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en`.

[37] Saeed B Niku. *Introduction to robotics: analysis, systems, applications*, volume 7. Prentice Hall New Jersey, 2001.

[38] Soromo Haman. *Prozessnahes Qualitätsmanagement beim Spritzgießen.* PhD thesis, Technische UniversitÃ¤t Chemnitz, 2004.

[39] Claude E Shannon. **Communication in the presence of noise**. *Proceedings of the IRE*, 37(1):10–21, 1949.

[40] Andrew S Tanenbaum. *Computernetzwerke, 4. Auflage (2003).* Pearson Studium Verlag,[Tanenbaum].

[41] Lillian Goleniewski. *Telecommunications essentials: the complete global source for communications fundamentals, data networking and the Internet, and next-generation networks.* Addison-Wesley Professional, 2002.

[42] Douglas Adams. *The Complete Hitchhiker's Guide to the Galaxy: The Trilogy of Five.* Tor UK, 2012.

[43] International Standardization Organization. **Iso 13372:2012 condition monitoring and diagnostics of machines. vocabularys**, 2012.

[44] Ratna Babu Chinnam and Pundarikaksha Baruah. **A neuro-fuzzy approach for estimating mean residual life in condition-based maintenance systems**. *International Journal of Materials and Product Technology*, 20(1-3):166–179, 2004.

[45] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela H Byers. **Big data: The next frontier for innovation, competition, and productivity**. 2011.

[46] Landrock H. et al. **Big data vendor benchmark**. Ismaning Experton Group AG., 2013.

[47] Min Chen, Shiwen Mao, and Yunhao Liu. **Big data: A survey**. *Mobile Networks and Applications*, 19(2):171–209, 2014.

[48] Max Bramer, Max Bramer, and Max Bramer. *Principles of data mining*, volume 131. Springer, 2007.

[49] Wonhee Cho, Yoojin Lim, Hwangro Lee, Mohan Krishna Varma, Moonsoo Lee, and Eunmi Choi. **Big data analysis with interactive visualization using r packages**. In *Proceedings of the 2014 International Conference on Big Data Science and Computing*, pages 1–6, 2014.

[50] Nathan Yau. *Visualize This: The FlowingData Guide to Design, Visualization, and Statistics*. Wiley, Indianapolis, Ind., 2011.

[51] Ryan Hafen, Tara D. Gibson, Kerstin Kleese van Dam, and Terence Critchlow. **Large-scale exploratory analysis, cleaning, and modeling for event detection in real-world power systems data**. In *Proceedings of the 3rd International Workshop on High Performance Computing, Networking and Analytics for the Power Grid*, pages 1–9, 2013.

[52] Drew Conway and John Myles White. *Machine Learning for Hackers*. O'Reilly, Sebastopol, 2012.

[53] Thomas Herrmann. **Learning and teaching in socio-technical environments**. In *Informatics and the Digital Society*, pages 59–71. Springer, 2003.

[54] R. Hartson. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2012.

[55] John Ronald Reuel Tolkien. *The Lord of the Rings: One Volume*. Houghton Mifflin Harcourt, 2012.

[56] Alan Cooper et al. *The inmates are running the asylum:[Why high-tech products drive us crazy and how to restore the sanity]*. Sams Indianapolis, IN, USA:, 2004.

[57] Donald A Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[58] Detlef Zühlke. *Nutzergerechte Entwicklung von Mensch-Maschine-Systemen: Useware-Engineering für technische Systeme*. Springer-Verlag, 2011.

[59] Julie A Adams. **Critical considerations for human-robot interface development**. In *Proceedings of 2002 AAAI Fall Symposium*, pages 1–8, 2002.

[60] Sandra G Hart and Lowell E Staveland. **Development of nasa-tlx (task load index): Results of empirical and theoretical research**. *Advances in psychology*, 52:139–183, 1988.

[61] Karin Eilers, Friedhelm Nachreiner, and Kerstin Hänecke. **Entwicklung und überprüfung einer skala zur erfassung subjektiv erlebter anstrengung**. *Zeitschrift für Arbeitswissenschaft*, (4):214–224, 1986.

[62] Kenneth James Williams Craik. *The nature of explanation*, volume 445. CUP Archive, 1967.

[63] Donald A Norman. **Some observations on mental models**. *Mental models*, 7(112):7–14, 1983.

[64] Thomas B Sheridan. *Telerobotics, automation, and human supervisory control.* MIT press, 1992.

[65] Charles Kay Ogden, Ivor Armstrong Richards, Sv Ranulf, and E Cassirer. *The Meaning of Meaning. A Study of the Influence of Language upon Thought and of the Science of Symbolism.* JSTOR, 1923.

[66] Neville Stanton, Paul M Salmon, and Laura A Rafferty. *Human factors methods: a practical guide for engineering and design.* Ashgate Publishing, Ltd., 2013.

[67] Jakob Nielsen. *Usability engineering.* Elsevier, 1994.

[68] Stuart K Card, Allen Newell, and Thomas P Moran. **The psychology of human-computer interaction**. 1983.

[69] Paul Holleis, Friederike Otto, Heinrich Hussmann, and Albrecht Schmidt. **Keystroke-level model for advanced mobile phone interaction**. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1505–1514. ACM, 2007.

[70] International Organization for Standardization. **Din en iso 9241-110 (2006). ergonomics of human-system interaction - part 110: Dialogue principles.**, 2006.

[71] Jochen Prümper. **Der benutzungsfragebogen isonorm 9241/10: Ergebnisse zur reliabilität und validität**. In *Software-Ergonomie'97*, pages 253–262. Springer, 1997.

[72] Jörn Hurtienne and Anja Naumann. **Quesi – a questionnaire for measuring the subjective consequences of intuitive use**. *Interdisciplinary College*, 536, 2010.

[73] Edward Clarkson and Ronald C Arkin. **Applying heuristic evaluation to human-robot interaction systems.** In *Flairs Conference*, pages 44–49, 2007.

[74] Aaron Steinfeld, Terrence Fong, David Kaber, Michael Lewis, Jean Scholtz, Alan Schultz, and Michael Goodrich. **Common metrics for human-robot interaction**. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 33–40. ACM, 2006.

[75] Tom Gross, Michael Koch, and Michael Herczeg. *Computer-Supported Cooperative Work–Interaktive Medien zur Unterstützung von Teams und Communities.* Oldenbourg Verlag, 2007.

[76] Heiko Häckelmann, Hans J Petzold, and Susanne Strahringer. *Kommunikationssysteme: Technik und Anwendungen.* Springer-Verlag, 2011.

[77] P Watlawick, JH Beavin, and DD Jackson. *Menschliche Kommunikation. Formen, Störungen, Paradoxien.* Huber Verlag. Bern, 2000.

[78] Friedemann Schulz von Thun. *Miteinander reden 1: Störungen und Klärungen.* 1981.

[79] Herbert H Clark and Thomas B Carlson. **Context for comprehension**. *Attention and performance IX*, pages 313–330, 1981.

[80] Herbert H Clark and Thomas B Carlson. **Hearers and speech acts**. *Language*, pages 332–373, 1982.

[81] Herbert H Clark and Susan E Brennan. **Grounding in communication**. *Perspectives on socially shared cognition*, 13(1991):127–149, 1991.

[82] Robert B Ochsman and Alphonse Chapanis. **The effects of 10 communication modes on the behavior of teams during co-operative problem-solving**. *International Journal of Man-Machine Studies*, 6(5):579–619, 1974.

[83] Willibald Pschyrembel. *Klinisches Wörterbuch mit klinischen Syndromen und Nomina Anatomica. 256., neu bearbeitete Auflage.* 1990.

[84] Gunnar Johannsen. *Mensch-Maschine-Systeme.* Springer-Verlag, 2013.

[85] Jens Rasmussen. **Information processing and human-machine interaction. an approach to cognitive engineering**. 1986.

[86] Mica R Endsley and Daniel J Garland. *Situation awareness analysis and measurement.* CRC Press, 2000.

[87] Teun Adrianus Van Dijk, Walter Kintsch, and Teun Adrianus Van Dijk. *Strategies of discourse comprehension.* Citeseer, 1983.

[88] Mica R Endsley. **Design and evaluation for situation awareness enhancement**. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 32, pages 97–101. SAGE Publications, 1988.

[89] RM Taylor. **Situational awareness rating technique(sart): The development of a tool for aircrew systems design**. *AGARD, Situational Awareness in Aerospace Operations 17 p(SEE N 90-28972 23-53)*, 1990.

[90] Mica R Endsley and Michelle M Robertson. **Situation awareness in aircraft maintenance teams**. *International Journal of Industrial Ergonomics*, 26(2):301–325, 2000.

[91] Priti Shah and Akira Miyake. *The Cambridge handbook of visuospatial thinking.* Cambridge University Press, 2005.

[92] Christopher D Wickens, Michelle Vincow, and Michelle Yeh. *Design applications of visual spatial thinking: The importance of frame of reference.* na, 2005.

[93] Christopher D Wickens. **Aviation displays**. *Principles and practices of aviation psychology*, pages 147–199, 2003.

[94] Oscar Olmos, Christopher D Wickens, and Andrew Chudy. **Tactical displays for combat awareness: An examination of dimensionality and frame of reference concepts and the application of cognitive engineering**. *The International Journal of Aviation Psychology*, 10(3):247–271, 2000.

[95] Anthony J Aretz. **The design of electronic map displays**. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 33(1):85–101, 1991.

[96] Stanley N Roscoe. **Airborne displays for flight and navigation**. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 10(4):321–332, 1968.

[97] Chun-houh Chen, Wolfgang Karl Härdle, and Antony Unwin. *Handbook of data visualization.* Springer Science & Business Media, 2007.

[98] Bruce M Durding, Curtis A Becker, and John D Gould. **Data organization**. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 19(1):1–14, 1977.

[99] Hans-Jörg Schulz, Thomas Nocke, Magnus Heitzler, and Heidrun Schumann. **A design space of visualization tasks**. *IEEE transactions on visualization and computer graphics*, 19(12):2366–2375, 2013.

[100] William S Cleveland. **Robust locally weighted regression and smoothing scatterplots**. *Journal of the American statistical association*, 74(368):829–836, 1979.

[101] Danyel Fisher, Rob DeLine, Mary Czerwinski, and Steven Drucker. **Interactions with big data analytics**. *interactions*, 19(3):50–59, 2012.

[102] Steven P Callahan, Juliana Freire, Emanuele Santos, Carlos E Scheidegger, Cláudio T Silva, and Huy T Vo. **Vistrails: visualization meets data management**. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 745–747. ACM, 2006.

[103] Ben Shneiderman. **Extreme visualization: squeezing a billion records into a million pixels**. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 3–12. ACM, 2008.

[104] K. Sathyanarayana and G.V.V. Ravi Kumar. **Evolution of computer graphics and its impact on engineering product development**. In *5th International Conference on Computer Graphics, Imaging and Visualisation (CGIV)*, pages 32–37. IEEE, 2008.

[105] Gregorio Palmas, Myroslav Bachynskyi, Antti Oulasvirta, Hans-Peter Seidel, and Tina Weinkauf. **Movexp: A versatile visualization tool for human-computer interaction studies with 3d performance and biomechanical data**. *IEEE Transaction on Visualization and Computer Graphics*, 20(12):2359–2368, 2014.

[106] Mauro J.G. Figueiredo, Pedro J.S. Cardoso, Cesar D.F. Goncalves, and Joao M.F. Rodrigues. **Augmented reality and holograms for the visualization of mechanical engineering parts**. In *18th International Conference on Information Visualisation (IV)*, pages 368–373. IEEE, 2014.

[107] Edward Tse, Min Xin, Viktor Antonyuk, Henry Lai, and Carl Hudson. **Making 3d content accessible for teachers**. In Aaron Quigley, Giulio Jacucci, Michael Horn, and Miguel Nacenta, editors, *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*, pages 125–134. ACM, 2013.

[108] Jonathan Steuer, Frank Biocca, Mark R Levy, et al. **Defining virtual reality: Dimensions determining telepresence**. *Communication in the age of virtual reality*, pages 33–56, 1995.

[109] Minna Pakanen and Leena Arhippainen. **User experiences with web-based 3d virtual travel destination marketing portals**. In Tuck Leong, editor, *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design*, pages 430–439, 2014.

[110] Alan H. Watt. *3D Computer Graphics*. Addison-Wesley, Reading, 3rd edition edition, 2000.

[111] Karl Marx. **Das kapital, bd. 1**. *MEW, Bd*, 23:405, 1867.

[112] John Vince. *Mathematics for Computer Graphics*. Springer, London, 2nd edition edition, 2006.

[113] Lothar Steiger Professor Dr. Anett Mehler-Bicher, Michael Reiss. *Augmented Reality - Theorie und Praxis*. Oldenbourg Verlag MÃ¼nchen, 2011.

[114] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. **Augmented reality: A class of displays on the reality-virtuality continuum**. In *Photonics for Industrial Applications*, pages 282–292. International Society for Optics and Photonics, 1995.

[115] Ramesh Raskar Oliver Bimber. *Spatial Augmented Reality - Merging Real and Virtual Worlds*. A K Peters, Ltd., 2005.

[116] DWF Van Krevelen and R Poelman. **A survey of augmented reality technologies, applications and limitations**. *International Journal of Virtual Reality*, 9(2):1, 2010.

[117] Jean-Yves Bouguet. **Camera calibration toolbox for matlab**. online, October 2015. URL: `http://www.vision.caltech.edu/bouguetj/calib_doc/index.html`.

[118] Arthur Tang, Charles Owen, Frank Biocca, and Weimin Mou. **Experimental evaluation of augmented reality in object assembly task**. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 265–265. IEEE Computer Society, 2002.

[119] Ulrich Neumann and Anthony Majoros. **Cognitive, performance, and systems issues for augmented reality applications in manufacturing and maintenance**. In *Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998*, pages 4–11. IEEE, 1998.

[120] G. Reinhart and C. Patron. **Integrating augmented reality in the assembly domain - fundamentals, benefits and applications**. *{CIRP} Annals - Manufacturing Technology*, 52(1):5 – 8, 2003. URL: `http://www.sciencedirect.com/science/article/pii/S0007850607605174`.

[121] Sabine Webel, Uli Bockholt, Timo Engelke, Nirit Gavish, Manuel Olbrich, and Carsten Preusche. **An augmented reality training platform for assembly and maintenance skills**. *Robotics and Autonomous Systems*, 61(4):398 – 403, 2013. Models and Technologies for Multi-modal Skill Training. URL: `http://www.sciencedirect.com/science/article/pii/S0921889012001674`.

[122] Sabine Webel, Ulrich Bockholt, and Jens Keil. **Design criteria for ar-based training of maintenance and assembly tasks**. In *Virtual and Mixed Reality-New Trends*, pages 123–132. Springer, 2011.

[123] Claudio Melchiorri. **Robot teleoperation**. In John Baillieul and Tariq Samad, editors, *Encyclopedia of Systems and Control*, pages 1–14. Springer London, 2014. URL: `http://dx.doi.org/10.1007/978-1-4471-5102-9_172-1`.

[124] S Lichiardopol. **A survey on teleoperation**. *University of Eindhoven, Department Mechanical Engineering Dynamics and Control Group Eindhoven*, 2007.

[125] Robin R Murphy. **Human-robot interaction in rescue robotics**. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(2):138–153, 2004.

[126] Terrence Fong and Illah Nourbakhsh. **Interaction challenges in human-robot space exploration**. 2005.

[127] Markus Sauer. *Mixed-Reality for Enhanced Robot Teleoperation*. PhD thesis, Universität Würzburg - Lehrstuhl für Informatik VII, 2010.

[128] Frauke Driewer. *Teleoperation Interfaces in Human-Robot Teams*. PhD thesis, Universität Würzburg - Lehrstuhl für Informatik VII, 2009.

[129] S Riihiaho. **The pluralistic usability walk-through method**. *Ergonomics in Design*, 10(3):23–30, 2002.

[130] techconsult. **Consumerization study cio challenges 2012, herausforderungen im umgang mit "bring your own"**. Microsoft Whitepaper, 2012.

[131] J Barbier, J Bradley, J Macaulay, and C Reberger. **Byod and virtualization: Top 10 insights from cisco ibsg horizons study. cisco ibsg horizons**, 2012.

[132] Franz Som. **Smart robot programming - die einfache art zu programmieren**. URL: `http://www.reisrobotics.de/de/media/file/1496`.

[133] Mario Hermann, Tobias Pentek, and Boris Otto. **Design principles for industrie 4.0 scenarios**. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 3928–3927. IEEE, 2016.

[134] Jaime Campos, erkki Jantunen, David Baglee, Eduardo Gilabert, Luca Fumagalli, and Christos Emmanouilidis. **The use of mobile technologies and their economical benefits in maintenance**. In *Proceedings of the 10th World Congress on Engineering Asset Management (WCEAM 2015)*, pages 113–119. Springer, 2016.

[135] Jaime Campos. **Development in the application of ict in condition monitoring and maintenance**. *Computers in Industry*, 60(1):1–20, 2009.

[136] VDI/VDE. **Richtlinie 3850 nutzergerechte gestaltung von bediensystemen für maschinen blatt 3 dialoggestaltung für touchscreens**, 2014.

[137] Jesper Kjeldskov and Connor Graham. **A review of mobile hci research methods**. In *International Conference on Mobile Human-Computer Interaction*, pages 317–335. Springer, 2003.

[138] Everett N McKay. *UI is communication: How to design intuitive, user centered interfaces by focusing on effective communication*. Newnes, 2013.

[139] matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. **A survey on context-aware systems**. *International Journal of Ad Hov and Ubiquitous Computing*, 2:263–277, 2007.

[140] Dirk Reiners, Didier Stricker, Gudrun Klinker, and Stefan Müller. **Augmented reality for construction tasks: Doorlock assembly**. *Proc. IEEE and ACM IWAR*, 98(1):31–46, 1998.

[141] Alex Olwal, Jonny Gustafsson, and Christoffer Lindfors. **Spatial augmented reality on industrial cnc-machines**. In *Electronic Imaging 2008*, pages 680409–680409. International Society for Optics and Photonics, 2008.

[142] Mihai Duguleana, Florin Girbacia, and Adrian Stavar. **Off-line programming of industrial robots using co-located environments**. *Advanced Materials Research*, 463:1654–1657, 2012.

[143] AE Ameri, B Akan, and B Curuklu. **Augmented reality meets industry: Interactive robot programming**. In *Conference SIGRAD*, 2010.

[144] Florian Leutert, Christian Herrmann, and Klaus Schilling. **A spatial augmented reality system for intuitive display of robotic data**. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pages 179–180. IEEE Press, 2013.

[145] Florian Leutert and Klaus Schilling. **Augmented reality for telemaintenance and -inspection in force-sensitive industrial robot applications**. *IFAC-PapersOnLine*, 48:153–158, 2015.

[146] Bill Buxton. *Sketching user experiences: getting the design right and the right design*. Morgan Kaufmann, 2010.

[147] Saul Greenberg, Sheelagh Carpendale, Nicolai Marquardt, and Bill Buxton. *Sketching user experiences: The workbook*. Elsevier, 2011.

[148] Ben Fry and Casey Reas. **Processing**, 2001. URL: `https://processing.org/`.

[149] Open Source Project. **Zxing ("zebra crossing") library**. URL: `https://github.com/zxing/zxing`.

[150] Pokini. **Pokini tab a8 / a10**, 2016. URL: `http://www.pokini.de/de/produkte/tablet-pcs/pokini-tab-a810`.

[151] Hervé Abdi. **The bonferonni and šidák corrections for multiple comparisons**. *Encyclopedia of measurement and statistics*, 3:103–107, 2007.

[152] Jacob Cohen. **A power primer.** *Psychological bulletin*, 112(1):155, 1992.

[153] Metaio. **Metaio - the augmented reality company**, 2015. URL: `https://www.metaio.com/index.html`.

[154] Blender Foundation. **Blender 3d creation suite**, 2016. URL: `https://www.blender.org/`.

[155] Donald Ervin Knuth. *The art of computer programming: sorting and searching*, volume 3. Pearson Education, 1998.

[156] Dominic Mazzoni. **Audacity**, 2016. URL: `http://www.audacityteam.org/`.

[157] Kesheng Wang. **Applying data mining to manufacturing: the nature and implications**. *Journal of Intelligent Manufacturing*, 18(4):487–495, 2007.

[158] Changyu Shen, Lixia Wang, and Qian Li. **Optimization of injection molding process parameters using combination of artificial neural network and genetic algorithm method**. *Journal of Materials Processing Technology*, 183(2):412–418, 2007.

[159] Ke-Sheng Wang. **Towards zero-defect manufacturing (zdm)–a data mining approach**. *Advances in Manufacturing*, 1(1):62–74, 2013.

[160] J. A. Harding, M. Shahbaz, Srinivas, and A. Kusiak. **Data mining in manufacturing: A review**. *Journal of Manufacturing Science and Engineering*, 128(4):969, 2006.

[161] Dan Braha. *Data mining for design and manufacturing: methods and applications*, volume 3. Springer Science & Business Media, 2013.

[162] C. Angeli. **Diagnostic expert systems: From expert's knowledge to real-time systems**. In Rajendra Akerkar and Priti Sajja, editors, *Knowledge-Based Systems*, pages 50–73. Jones & Bartlett, Sudbury, Mass., 2010.

[163] Yuangao Ai and Zhaohui Li. **A virtual reality based diagnosis scheme for hydro power facility**. In *Power and Energy Engineering Conference (APPEEC), 2012 Asia-Pacific*, pages 1–5. IEEE, 2012.

[164] Li Zheng, Chunqiu Zeng, Lei Li, Yexi Jiang, Wei Xue, Jingxuan Li, Chao Shen, Wubai Zhou, Hongtai Li, Liang Tang, Tao Li, Bing Duan, Ming Lei, and Pengnian Wang. **Applying data mining techniques to address critical process optimization needs in advanced manufacturing**. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1739–1748, New York, New York, USA, 2014. ACM.

[165] Xin Yan, Mu Qiao, Jia Li, Timothy W. Simpson, Gary M. Stump, and Xiaolong Zhang. **A work-centered visual analytics model to support engineering design with interactive visualization and data-mining**. In *45th Hawaii International Conference on System Science*, pages 1845–1854, Piscataway, NJ, 2012. IEEE.

[166] Shi Yongsheng and Huang Xinming. **A vr-based visualization fault diagnosis method for the aircraft brake system**. In *International Conference on Computer Application and System Modeling (ICCASM 2010)*, pages 609–612, 2010.

[167] Michael J. Russell, George D. Lecakes, Shreekanth Mandayam, and Scott Jensen. **The "intelligent" valve: A diagnostic framework for integrated system-health management of a rocket-engine test stand**. *IEEE Transactions on Instrumentation and Measurement*, 60(4):1489–1497, 2011.

[168] Michael Albert, Matthias Gors, and Klaus Schilling. **Telemedical applications with rulebased descision-and information-systems (tardis)**. *IFAC-PapersOnLine*, 48(10):7–11, 2015.

[169] Robert J. Marks. *Introduction to Shannon sampling and interpolation theory*. Springer texts in electrical engineering. Springer, New York, 1991.

[170] Michael Frischter, Robin Heß, and Klaus Schilling. **Generic network and a browser-based hmi**. In *IARP RISE 2012 6th Workshop (RISE2012)*, 2012.

[171] Irmen de Jong. **Pyrolite: Dlient library for java and .net**, 2015. URL: https://pythonhosted.org/Pyro4/pyrolite.html.

[172] J. D. Hunter. **Matplotlib: A 2d graphics environment**. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[173] Wes McKinney. **Pandas: Python data analysis library**, 2015. URL: http://pandas.pydata.org/.

[174] Microsoft. **Die sprache c# im detail**, 2016. URL: https://msdn.microsoft.com/de-de/library/bb979023.aspx.

[175] Y. Shafranovich. **Common format and mime type for comma-separated values (csv) files**. 2005. URL: http://tools.ietf.org/html/rfc4252.

[176] Nick V Flor. **Side-by-side collaboration: A case study**. *International Journal of Human-Computer Studies*, 49(3):201–222, 1998.

[177] John C Tang. **Findings from observational studies of collaborative work**. *International Journal of Man-machine studies*, 34(2):143–160, 1991.

[178] Hideaki Kuzuoka. **Spatial workspace collaboration: a sharedview video support system for remote collaboration capability**. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 533–540. ACM, 1992.

[179] Robert E Kraut, Susan R Fussell, and Jane Siegel. **Visual information as a conversational resource in collaborative physical tasks**. *Human-computer interaction*, 18(1):13–49, 2003.

[180] Darren Gergle, Robert E Kraut, and Susan R Fussell. **Using visual information for grounding and awareness in collaborative tasks**. *Human–Computer Interaction*, 28(1):1–39, 2013.

[181] Per Cederberg, Magnus Olsson, and Gunnar Bolmsjö. **Remote control of a standard abb robot system in real time using the robot application protocol (rap)**. In *Proceedings of the 33rd ISR (International Symposium on Robotics) October*, volume 7, page 11, 2002.

[182] Mohsen Moradi Dalvand and Saeid Nahavandi. **Teleoperation of abb industrial robots**. *Industrial Robot: An International Journal*, 41(3):286–295, 2014. URL: http://www.emeraldinsight.com/doi/full/10.1108/IR-09-2013-400.

[183] Geoff Pegman, Philippe Desbats, Franck Geffard, Gérard Piolain, and Alain Coudray. **Force-feedback teleoperation of an industrial robot in a nuclear spent fuel reprocessing plant**. *Industrial Robot: An International Journal*, 33(3):178–186, 2006.

[184] Olivier David, François-Xavier Russotto, Max Da Silva Simoes, and Yvan Measson. **Collision avoidance, virtual guides and advanced supervisory control teleoperation techniques for high-tech construction: framework design**. *Automation in Construction*, 44:63–72, 2014.

[185] T Slama, D Aubry, R Oboe, and F Kratz. **Robust bilateral generalized predictive control for teleoperation systems**. In *Control & Automation, 2007. MED'07. Mediterranean Conference on*, pages 1–6. IEEE, 2007.

[186] Steve Vozar and Dawn M Tilbury. **Driver modeling for teleoperation with time delay**. In *IFAC World Congress*, volume 19, pages 3551–3556, 2014.

[187] JO Gray, Vincenzo Lippiello, Luigi Villani, and Bruno Siciliano. **An open architecture for sensory feedback control of a dual-arm industrial robotic cell**. *Industrial Robot: An International Journal*, 34(1):46–53, 2007.

[188] J Gomez Ortega, J Gamez García, LM Nieto Nieto, and A Sánchez García. **Open software architecture for advanced control of robotic manipulators**. 2010.

[189] Martin Hagele, Klas Nilsson, and J.Norberto Pires. **Industrial robotics**. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 963–986. Springer Berlin Heidelberg, 2008. URL: http://dx.doi.org/10.1007/978-3-540-30301-5_43.

[190] Angelika Salmen, Tobias Münch, Steffen Buzin, Jan Hladik, Werner Altmann, Conny Weber, Dimitris Karagiannis, Robert Buchmann, Jens Ziegler,

Johannes Pfeffer, et al. *Comvantage: Mobile enterprise collaboration reference framework and enablers for future internet information interoperability.* Springer, 2013.

[191] Yunyi Jia. *Teleoperation of mobile manipulators.* PhD thesis, MICHIGAN STATE UNIVERSITY, 2014.

[192] Imad H Elhajj, Nadine Bou Dargham, Ning Xi, and Yunyi Jia. **Real-time adaptive content-based synchronization of multimedia streams**. *Advances in Multimedia*, 2011:4, 2011.

[193] Robert J Kosinski. **A literature review on reaction time**. *Clemson University*, 10, 2008.

[194] Maria C Consiglio, Sara R Wilson, James Sturdy, Jennifer L Murdoch, and David J Wing. **Human in the loop simulation measures of pilot response delay in a self-separation concept of operations**. 2010.

[195] Reis Robotics / KUKA Industries. **Reis robotstar-vi combines safety and ease of use**, 2015. URL: `http://www.reisrobotics.de/en/news/reis-robotstar-vi-combines-safety-and-ease-of-use`.

[196] **Netfilter project**. URL: `http://www.netfilter.org/`.

[197] **Iptables**. URL: `http://www.netfilter.org/projects/iptables`.

[198] **traffic control documentation**. URL: `http://linux-ip.net/articles/Traffic-Control-HOWTO/`.

[199] **Openvpn homepage**. URL: `http://openvpn.net/`.

[200] Josh et. al Nimoy. **Myron - computer vision for artists**, 2016. URL: `http://webcamxtra.sourceforge.net/index.shtml`.

[201] Open Source Project. **Opencv for processing. a creative coding computer vision library based on th official opopen java api**, 2016. URL: `https://github.com/atduskgreg/opencv-processing`.

[202] Open Source Project. **Boofcv**, 2016. URL: `http://boofcv.org/`.

[203] Peter Abeles. **Speeding up surf**. In *International Symposium on Visual Computing*, pages 454–464. Springer, 2013.

[204] Keni Bernardin and Rainer Stiefelhagen. **Evaluating multiple object tracking performance: the clear mot metrics**. *EURASIP Journal on Image and Video Processing*, 2008(1):1–10, 2008.

[205] Jaclyn Ocumpaugh, Ryan S Baker, Ma Mercedes Rodrigo, Aatish Salvi, Martin van Velsen, Ani Aghababyan, and Taylor Martin. **Hart: the human affect recording tool**. In *Proceedings of the 33rd Annual International Conference on the Design of Communication*, page 24. ACM, 2015.

[206] DM Keirsey, R Milner, and V Wood. **Keirsey temperament sorter ii**, 2004.

[207] Robert R Hoffman, Beth Crandall, and Nigel Shadbolt. **Use of the critical decision method to elicit expert knowledge: A case study in the methodology of cognitive task analysis**. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 40(2):254–276, 1998.

[208] Robin Heß and Klaus Schilling. **Gps/galileo testbed using a high precision optical positioning system**. In *Simulation, Modeling, and Programming for Autonomous Robots*, pages 87–96. Springer, 2010.

[209] Unity Technologies. **Unity**, 2015. URL: `http://unity3d.com/`.

[210] Russell T Warne. **A primer on multivariate analysis of variance (manova) for behavioral scientists**. *Practical Assessment, Research & Evaluation*, 19(17), 2014.